Discrete logarithm record in a 508-bit finite field GF(p^3) with the Number Field Sieve algorithm

Aurore Guillevic and François Morain and Emmanuel Thomé

University of Calgary, PIMS–CNRS, LIX–École Polytechnique, Inria, Loria

CNTA, June 24, 2016

Outline

Motivation: Pairing-based cryptography

The Number Field Sieve algorithm

GF(p^3): breaking a 508-bit MNT curve

Asymetric cryptography

Factorization (RSA cryptosystem)

Discrete logarithm problem (Diffie–Hellman, etc)

Given a finite cyclic group $(\mathbf{G}, \cdot)$, a generator g and $y \in \mathbf{G}$, compute x s.t. $y = g^x$.

Common choice of $\mathbf{G}$:
prime finite field $\mathbb{F}_p$ (since 1976), characteristic 2 finite field $\mathbb{F}_{2^n}$, elliptic curve $E(\mathbb{F}_p)$ (since 1985)

Elliptic curves in cryptography

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

- proposed in 1985 by Koblitz, Miller
- $E(\mathbb{F}_p)$ has an efficient group law (chord an tangent rule) $\rightarrow$ **G**
- $\#E(\mathbb{F}_p) = p + 1 - t$, trace t: $|t| \leqslant 2\sqrt{p}$

Need a prime-order (or with tiny cofactor) elliptic curve:

$$h \cdot \ell = \#E(\mathbb{F}_p),\ \ \ell \text{ is prime},\ \ h \text{ tiny, e.g. } h = 1, 2$$

- compute t
- slow to compute in 1985: can use *supersingular curves* whose trace is 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.

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
There exists a pairing e that embeds the group $E(\mathbb{F}_p)$ into $\mathbb{F}_{p^2}$ where **DLP is much easier**.
Do not use supersingular curves.

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
There exists a pairing e that embeds the group $E(\mathbb{F}_p)$ into $\mathbb{F}_{p^2}$
where **DLP is much easier**.
Do not use supersingular curves.
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.

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
There exists a pairing e that embeds the group $E(\mathbb{F}_p)$ into $\mathbb{F}_{p^2}$ where **DLP is much easier**.
Do not use supersingular curves.
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** $\rightarrow$ take a large enough field

Common target groups $\mathbb{F}_{p^n}$

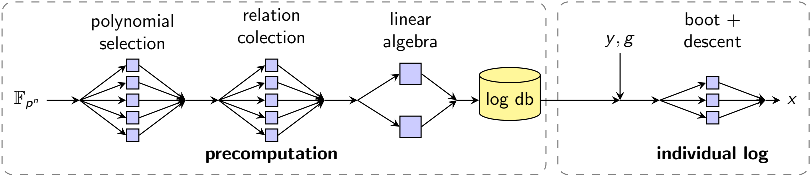
- $\mathbb{F}_{p^2}$ where $E/\mathbb{F}_p$ is a supersingular curve
- $\mathbb{F}_{p^3}, \mathbb{F}_{p^4}, \mathbb{F}_{p^6}$ where E is an ordinary MNT curve *[Miyaji Nakabayashi Takano 01]*
- $\mathbb{F}_{p^{12}}$ where E is a BN curve *[Barreto-Naehrig 05]*

DLP hardness for a 3072-bit finite field:

- **hard** in $\mathbb{F}_p$ where p is a 3072-bit prime
- **easy** in $\mathbb{F}_{2^n}$ where $n = 3072$ *[Barbulescu, Gaudry, Joux, Thomé 14, Granger et al. 14]*
- what about $\mathbb{F}_{p^3}$ where p is a 1024-bit prime?

NFS algorithm to compute discrete logarithms

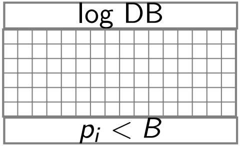
Input : finite field $\mathbb{F}_{p^n}$, generator g, target y
Output : discrete logarithm x of y in basis g, $g^x = y$



[graph: N. Heninger]

Relation collection and Linear algebra

1. Polynomial selection
2. Relation collection (cado-nfs: Gaudry, Grémy)
3. Linear algebra (cado-nfs: Thomé, 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 $a_i - b_i x = \mathfrak{p}_i \in \mathbb{Z}[x]/(f(x))$, s.t. $|\,\mathrm{Norm}(\mathfrak{p}_i)| = p_i < B$

4. Individual discrete logarithm

NFS algorithm for DL in $\mathrm{GF}(p^n)$

How to generate relations ?
Use *two* distinct rings $R_f = \mathbb{Z}[x]/(f(x))$, $R_g = \mathbb{Z}[x]/(g(x))$ and two maps ρ_f, ρ_g that map $x \in R_f$, resp. $x \in R_g$ to *the same* element $z \in \mathrm{GF}(p^n)$:

$$\begin{cases} \rho_f : x \in R_f \mapsto z, \\ \rho_g : x \in R_g \mapsto z \end{cases}$$

$$\begin{tikzcd}
 & \mathbb{Z}[x] \arrow[dl] \arrow[dr] & \\
R_f = \mathbb{Z}[x]/(f(x)) \arrow[dr, "R_f \ni x \mapsto z"'] & & R_g = \mathbb{Z}[x]/(g(x)) \arrow[dl, "R_g \ni x \mapsto z"] \\
 & \mathrm{GF}(p^n) = \mathrm{GF}(p)[z]/(\varphi(z)) &
\end{tikzcd}$$

Weak MNT curve, 170-bit prime p, 508-bit $\mathbb{F}_{p^3}$

[Miyaji Nakabayashi Takano 01]

$E/\mathbb{F}_p : \ y^2 = x^3 + ax + b$, where

$a =$ 0x22ffbb20cc052993fa27dc507800b624c650e4ff3d2

$b =$ 0x1c7be6fa8da953b5624efc72406af7fa77499803d08

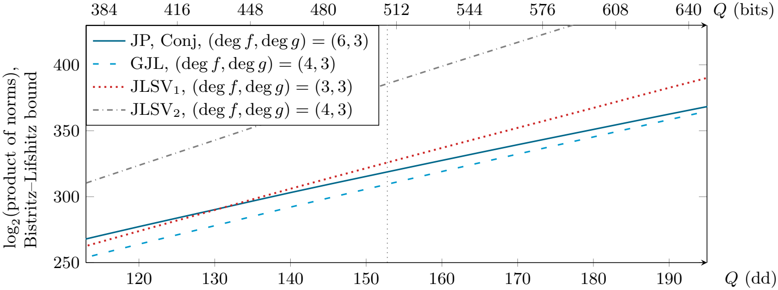
$p =$ 0x26dccacc5041939206cf2b7dec50950e3c9fa4827af

$\ell =$ 0xa60fd646ad409b3312c3b23ba64e082ad7b354d

such that

$$\begin{aligned} x_0 &= -\text{0x732c8cf5f983038060466} \\ t &= 6x_0 - 1 \\ p &= 12x_0^2 - 1 \\ \#E(\mathbb{F}_p) &= p + 1 - t = 7^2 \cdot 313 \cdot \ell \end{aligned}$$

Polynomial selection: norm estimates



Polynomial selection: norm estimates

Joux–Pierrot and Conjugation	319 bits	Galois aut. order 3
Generalized Joux–Lercier	310 bits	–
JouxLercierSmartVercauteren JLSV1	326 bits	Galois aut. order 3

Galois automorphism of order 3 $\rightarrow$ will obtain 3 times more relations for free

- JLSV1: $\sqrt{p} \approx 2^{85}$ possible polynomials f
- Conjugation: allow non-monic polynomials $\rightarrow \approx 2^{20}$ possible f

Polynomial Selection

Parameterized family:
$\varphi(x, y) = x^3 - yx^2 - (y + 3)x - 1$ s.t. $\mathbb{Q}[x]/(\varphi(x))$ has a degree 3 Galois automorphism $x \mapsto -1 - 1/x$
$f(x) = \mathrm{Resultant}_y(\varphi(x, y), A(y))$ where $A(y) = ay^2 - by + c$
Precomputation (independant of p):
Enumerate many A s.t. $\Delta(A) > 0$, $\|f\|_\infty \leq 2^9$ and f has good smoothness properties (α, Murphy's E value)
$\rightarrow$ enumerated $320749 \approx 2^{18}$ polys $A(y)$, kept 4143 ones s.t. $\alpha(f) < -1.5$.

For each good f:

1. compute a root y_0 mod p of $P(y)$
2. compute two rational reconstructions $y_0 \equiv u_1/v_1 \equiv u_2/v_2 \bmod p$ s.t. $|u_i|, |v_i| \approx \sqrt{p}$
3. $g_i \leftarrow v_i x^3 - u_i x^2 - (u_i + 3v_i)x - v_i$ so that $g_i = v_i \varphi \bmod p$.
4. take the best linear combination $g \leftarrow \lambda_1 g_1 + \lambda_2 g_2$, where $|\lambda_i| < 2^5$.

Polynomial Selection

$$\begin{aligned}
p &= 908761003790427908077548955758380356675829026531247 \\
&\text{of 170 bits} \\
A &= 28y^2 + 16y - 109 \\
f &= 28x^6 + 16x^5 - 261x^4 - 322x^3 + 79x^2 + 152x + 28 \\
&\|f\|_\infty = 8.33 \text{ bits} \\
\alpha(f) &= -2.9
\end{aligned}$$

$$\begin{aligned}
g &= 2475781518663919737044212 2x^3 + 408068970402536804717751 83x^2 \\
&\quad -334665485196639116395511 83x - 2475781518663919737044212 2 \\
&\|g\|_\infty = 85.01 \text{ bits} \\
\alpha(g) &= -4.1 \\
&\text{Murphy's E value:} \\
\mathbb{E}(f, g) &= 1.31 \cdot 10^{-12}
\end{aligned}$$

Relation Collection: sieving

Smoothness bound $B = 50000000 (= 2^{25.6})$ on both sides
Special-q in $[B, 2^{27}]$

660 core-days (4-core Intel Xeon E5520 @ 2.27GHz).

$57 \cdot 10^6$ relations $\rightarrow$ filtered $\rightarrow$
1982791×1982784 matrix with weight $w(M) = 396558692$.
The whole matrix would have 7 more columns for taking the 7 Schirokaurer Maps into account.

Linear Algebra (`cado-nfs`)

8 sequences in Block-Wiedemann algorithm.
8 Krylov sequences 250 core-days, four 16-code nodes / sequence
finding linear matrix generator 3.1 core-days / 64 cores
building solution 170 core-days
we were able to reconstruct virtual logarithms for 15196345 out of the 15206761 elements of the bases (99.9%).

423 core-days on a cluster Intel Xeon E5-2650, 2.4GHz

Individual discrete logarithm

Take $P_0 = [x_P, y_P] \in E(\mathbb{F}_p)$,
$x_P = \lfloor \pi 10^{50} \rfloor = 314159265358979323846264338327950288419716939937510$
$y_P = \sqrt{x_P^3 + a x_P + b} = 460095575547938627692618282835762310592027720907930$
and set $\mathrm{Target}_E = P = [7^2 \cdot 313] P_0$.

e is the reduced Tate pairing $e_\ell(P, Q)^{(p^3-1)/\ell}$

$E[\ell] \cong \mathbb{Z}/\ell\mathbb{Z} \oplus \mathbb{Z}/\ell\mathbb{Z} \simeq \langle G_1 \rangle \oplus \langle G_2 \rangle$ where
G_1 a generator of $E(\mathbb{F}_p)[\ell]$
G_2 a generator of $E(\mathbb{F}_{p^3})[\ell] \cap \ker(\pi_p - [p])$

Target in $\mathbb{F}_{p^3}$: $T = e(P, G_2)$, Basis: $g = e(G_1, G_2)$
Change $\mathbb{F}_{p^3} = \mathbb{F}_p[X]/(X^3 + X + 1)$ to $\mathbb{F}_p[Z]/(\varphi(Z))$

$$T = 0x11a2f1f13fa9b08703a033ee3c4321539156f865ee9 + 0x1098c3b7280ef2cf8b091d08197de0a9ba935ff79c6\ Z + 0x221205020e7729cb46166a9edfd5acb3bf59dd0a7d4\ Z^2$$

$$G_T = 0xd772111b150ec08f0ad89d987f1b037c630155608c + 0xf956cab6840c7e909abc29584f1aee48ccbd39d698\ Z + 0x205eb5b1e09f76bf0ef85efeaa3fdcb3827d43441b3\ Z^2$$

Individual discrete logarithm

Initial splitting: 32-core hours
preimage of g^{52154} in K_f has 59-bit-smooth norm
preimage of $g^{35313}T$ in K_f has 54-bit-smooth norm

Descent procedure: 13.4 hours.

Virtual log of g:
$\mathrm{vlog}(g) = 0x8c58b66f0d8b2e99a1c0530b2649ec0c76501c3$
virtual log of the target:
$\mathrm{vlog}(T) = 0x48a6bcf57cacca997658c98a0c196c25116a0aa$
Then $\log_g(T) = \mathrm{vlog}(T)/\mathrm{vlog}(g) \bmod \ell$.

$$\log(T) = \log(P) = 0x711d13ed75e05cc2ab2c9ec2c910a98288ec038 \bmod \ell\ .$$

Future work

- 600-bit DL record in $\mathbb{F}_{p^3}, \mathbb{F}_{p^4}, \mathbb{F}_{p^6}, \mathbb{F}_{p^{12}}$ (with Gaudry, Grémy, Morain, Thomé)
- need new techniques for $\mathbb{F}_{p^4}, \mathbb{F}_{p^6}, \mathbb{F}_{p^{12}}$ (*[Kim]* and *[Barbulescu–Gaudry–Kleinjung]*)
- implementation in `cado-nfs`

Consequences:
Increase the size of the target groups $\mathbb{F}_{p^n}$ in pairing-based cryptography

https://hal.inria.fr/hal-01320496