

Faster Individual Discrete logarithm in non-prime finite fields $GF(p^n)$ with the Number and Function Field Sieve Algorithms

Aurore Guillevic

University of Calgary, PIMS-CNRS

CMS, Edmonton, June 25, 2016



Motivation: Pairing-based cryptography

The Number Field Sieve algorithm

Individual Discrete Log

Asymmetric 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

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

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

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

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

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

Supersingular elliptic curves

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

$$E : y^2 = x^3 + x = \mathbb{F}_p; \quad p = 3 \pmod{4}$$

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 , $p \geq 5$

$$E : y^2 = x^3 + x = \mathbb{F}_p; \quad p = 3 \pmod{4}$$

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 , $p \geq 5$

$$E : y^2 = x^3 + x = \mathbb{F}_p; \quad p = 3 \pmod{4}$$

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{Ruck: 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.

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

Pairing-based cryptography

1999: Frey{Muller{Ruck: 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.

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

Attacks

Pairing-based cryptography

1999: Frey{Muller{Ruck: 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.

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

Attacks

- | inversion of e : hard problem (exponential)

Pairing-based cryptography

1999: Frey{Muller{Ruck: 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.

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

Attacks

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

Pairing-based cryptography

1999: Frey{Muller{Ruck: 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.

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

Attacks

- | inversion of e : hard problem (exponential)
- | discrete logarithm computation in $E(\mathbb{F}_q)$: hard problem (exponential, in $O(\sqrt{\cdot})$)
- | discrete logarithm computation in $\mathbb{F}_{q^n}^*$: **easier, subexponential** \rightarrow take a large enough field

Common target groups \mathbb{F}_{q^n}

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

DLP hardness for a 3072-bit finite field:

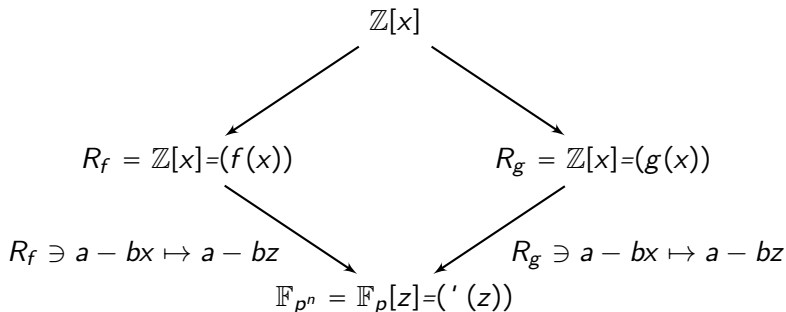
- | **hard** in \mathbb{F}_p where p is a 3072-bit prime
- | **easy** in $\mathbb{F}_{2^{4n}}, \mathbb{F}_{3^{6n}}$
[Barbulescu, Gaudry, Joux, Thomé 14, Granger et al. 14]
- | what about \mathbb{F}_{p^n} where $2 \leq n \leq 12$ and p^n is a 3072-bit?

Number Field Sieve algorithm for DL in $GF(p^n)$

How to generate relations ?

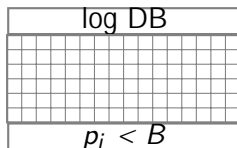
Use *two* distinct rings $R_f = \mathbb{Z}[x] = (f(x))$, $R_g = \mathbb{Z}[x] = (g(x))$ and two maps f, g that map $x \in R_f$, resp. $x \in R_g$ to the same element $z \in \mathbb{F}_{p^n}$:

$$\begin{cases} f : x \in R_f \mapsto z; \\ g : x \in R_g \mapsto z \end{cases}$$



Number Field Sieve algorithm for DL in $GF(p^n)$

1. Polynomial selection
2. Relation collection
3. Linear algebra



- ▶ 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 $a_j - b_j x = \mathfrak{p}_j \in \mathbb{Z}[x]=(f(x))$,
s.t. $|\text{Norm}(\mathfrak{p}_j)| = p_j < B$

4. Individual discrete logarithm

Initial Splitting in \mathbb{F}_p

An integer T is B -smooth if $N = \prod_i p_i^{e_i}$ and $p_i \leq B$

Algorithm 1: Generic Initial Splitting

Input: Target $T_0 \in \mathbb{F}_p$, generator g , subgroup order ℓ , bound B

Output: $t \in \mathbb{Z} = \ell\mathbb{Z}$, $\mathbf{T} \in \mathbb{Z}$ a preimage of $T = g^t T_0$, such that \mathbf{T} is B -smooth

```
1 repeat
2   | take  $t$  at random in  $\{1; \dots; \ell - 1\}$ 
3   |  $T \leftarrow g^t T_0$ 
4   |  $\mathbf{T} \leftarrow u=v \equiv T \pmod{p}$  a rational reconstruction of  $T \pmod{p}$ 
5 until  $\mathbf{T}$  is  $B$ -smooth, i.e.  $u$  and  $v$  are  $B$ -smooth
6 return  $\mathbf{T} = u=v; t$  //  $\log_g T_0 = \log_g (\mathbf{T}) - t$ 
```

Initial Splitting in $\mathbb{F}_p, \mathbb{F}_{p^n}, \mathbb{F}_{2^n}, \mathbb{F}_{3^n}$

- | \mathbb{F}_p : Rational Reconstruction. $T \in \mathbb{Z} = p\mathbb{Z}$, \mathbf{T} is an integer $< p$.
Rational Reconstruction gives $\mathbf{T} = u=v \bmod p$ with $u; v < \sqrt{p}$
- | [Blake Fuji-Hara Mullin Vanstone 84] Waterloo algorithm in \mathbb{F}_{2^n} : $\mathbb{F}_2[x] \ni \mathbf{T} \equiv U=V = \frac{u_0 + \dots + u_{\lfloor n/2 \rfloor} x^{\lfloor n/2 \rfloor}}{v_0 + \dots + v_{\lfloor n/2 \rfloor} x^{\lfloor n/2 \rfloor}}$ **reduce degree**
- | [Joux Lercier Smart Vercauteren 06] in \mathbb{F}_{p^n} :
 $\mathbf{T} \equiv U=V = \frac{u_0 + \dots + u_d x^d}{v_0 + \dots + v_d x^d}$, $d = \deg f \geq n$, $|u_i|; |v_i| \sim p^{n/(2 \deg f)}$
reduce coe **cient size**

Individual Discrete Log of target $T_0 \in \mathbb{F}_{p^n}^*$

Given g and a DL database s.t. for all $p_i < B_0 \sim 2^{27}$, $\log p_i$ is known,

Individual Discrete Log of target $T_0 \in \mathbb{F}_{p^n}^*$

Given g and a DL database s.t. for all $p_i < B_0 \sim 2^{27}$, $\log p_i$ is known,

1. initial splitting step (a.k.a. smoothing step):

DO

1.1 take t at random in $\{1; \dots; \ell - 1\}$ and set $T = g^t T_0$
(hence $\log_g(T_0) = \log_g(T) - t$)

1.2 factorize

$$\text{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } 2^{27} < q_i \leq 2^{90}} \times (\text{elements in DL database});$$

UNTIL $q_i \leq B_1 \sim 2^{90}$

Individual Discrete Log of target $T_0 \in \mathbb{F}_{p^n}^*$

Given g and a DL database s.t. for all $p_i < B_0 \sim 2^{27}$, $\log p_i$ is known,

1. initial splitting step (a.k.a. smoothing step):

DO

1.1 take t at random in $\{1; \dots; \ell - 1\}$ and set $T = g^t T_0$
(hence $\log_g(T_0) = \log_g(T) - t$)

1.2 factorize

$$\text{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } 2^{27} < q_i \leq 2^{90}} \times (\text{elements in DL database});$$

UNTIL $q_i \leq B_1 \sim 2^{90}$

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 $q_i = \prod_{B_0 < q_j < B_j} q_j \times (\text{elements in DL database})$
- ▶ $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

end while

Individual Discrete Log of target $T_0 \in \mathbb{F}_{p^n}^*$

Given g and a DL database s.t. for all $p_i < B_0 \sim 2^{27}$, $\log p_i$ is known,

1. initial splitting step (a.k.a. smoothing step):

DO

1.1 take t at random in $\{1; \dots; \ell - 1\}$ and set $T = g^t T_0$
(hence $\log_g(T_0) = \log_g(T) - t$)

1.2 factorize

$$\text{Norm}(\mathbf{T}) = \underbrace{q_1 \cdots q_i}_{\text{too large: } 2^{27} < q_i \leq 2^{90}} \times (\text{elements in DL database});$$

UNTIL $q_i \leq B_1 \sim 2^{90}$

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 $q_i = \prod_{B_0 < q_j < B_j} q_j \times (\text{elements in DL database})$
- ▶ $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

end while

3. log combination to find the individual target DL

Individual Discrete Log of target $T_0 \in \mathbb{F}_{p^n}^*$

Given g and a DL database s.t. for all $p_i < B_0 \sim 2^{27}$, $\log p_i$ is known,

1. initial splitting step (a.k.a. smoothing step):

DO

1.1 take t at random in $\{1; \dots; \ell - 1\}$ and set $T = g^t T_0$
(hence $\log_g(T_0) = \log_g(T) - t$)

1.2 factorize

$$\underbrace{\text{Norm}(T)}_{\text{reduce this}} = \underbrace{q_1 \cdots q_i}_{\text{too large: } 2^{27} < q_i \leq 2^{90}} \times (\text{elements in DL database});$$

UNTIL $q_i \leq B_1 \sim 2^{90}$

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 $q_i = \prod_{B_0 < q_j < B_j} q_j \times (\text{elements in DL database})$
- ▶ $\mathcal{S} \leftarrow \mathcal{S} \setminus \{q_i\} \cup \{q_j\}_{j \in J}$

end while

3. log combination to find the individual target DL

508-bit \mathbb{F}_{p^3} Polynomial Selection

$p = 908761003790427908077548955758380356675829026531247$
of 170 bits

$$A = 28y^2 + 16y - 109$$

$$' = x^3 - yx^2 - (y + 3)x - 1; \quad (x) \mapsto -x - 1 = x$$

$$f = \text{Res}_y(A; ')$$

$$= 28x^6 + 16x^5 - 261x^4 - 322x^3 + 79x^2 + 152x + 28$$

$$\|f\|_{\infty} = 8.33 \text{ bits}$$

$$(f) = -2.9$$

$$g = 24757815186639197370442122x^3 + 40806897040253680471775183x^2 \\ - 33466548519663911639551183x - 24757815186639197370442122$$

$$\|g\|_{\infty} = 85.01 \text{ bits}$$

$$(g) = -4.1$$

Murphy's E value:

$$\mathbb{E}(f; g) = 1.31 \cdot 10^{-12}$$

508-bit \mathbb{F}_{p^3} individual discrete logarithm

Target:

$$T_0 = 0x11a2f1f13fa9b08703a033ee3c4321539156f865ee9+0x1098c3b7280ef2cf8b091d08197de0a9ba935ff79c6 z \\ +0x221205020e7729cb46166a9edfd5acb3bf59dd0a7d4 z^2 \in \mathbb{F}_p[z] = (f(z))$$

Preimage: $T_0 = t_0 + t_1x + t_2x^2 \in \mathbb{Z}[x]$

$$\text{Norm}_f(T) = \text{Res}(f; T) \leq A \|T\|_\infty^{\deg f} \|f\|_\infty^{\deg T}$$

$$\text{Norm}_f(T_0) = \text{Res}(f; T_0) \text{ of } 1032 \text{ bits } \approx p^6 = Q^2$$

$$\text{Norm}_g(T_0) = \text{Res}(g; T_0) \text{ of } 670 \text{ bits } \approx p^4 = Q^{4/3}$$

Joux-Lercier:

$$\text{Norm}_f(JL_f(T_0)) \approx p^3 = Q$$

$$\text{Norm}_g(JL_g(T_0)) \approx p^4 = Q^{4/3}$$

Preimage improvement [G. 15]

Lemma

Let $T \in \mathbb{F}_{p^n}$.

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

Preimage improvement [G. 15]

Lemma

Let $T \in \mathbb{F}_{p^n}$.

$\log(T) = \log(u \cdot T) \pmod{\mathfrak{m}}$ for any u in a proper subfield of \mathbb{F}_{p^n} .

- | \mathbb{F}_p is a proper sub field of \mathbb{F}_{p^n}
- | target $T = t_0 + t_1x + \dots + t_dx^d$
- | we divide the target by its leading term:

$$\log(T) = \log(T=t_d) \pmod{\mathfrak{m}}$$

We can assume that the target is monic.

Preimage improvement [G. 15]

Lemma

Let $T \in \mathbb{F}_{p^n}$.

$\log(T) = \log(u \cdot T) \bmod \mathbb{F}_p$ for any u in a proper subfield of \mathbb{F}_{p^n} .

- | \mathbb{F}_p is a proper subfield of \mathbb{F}_{p^n}
- | target $T = t_0 + t_1x + \dots + t_dx^d$
- | we divide the target by its leading term:

$$\log(T) = \log(T/t_d) \bmod \mathbb{F}_p$$

We can assume that the target is monic.

Similar technique in pairing computation: Miller loop denominator elimination [Boneh Kim Lynn Scott 02]

Sub-eld Simplification + LLL

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

$$| \quad ' = x^3 - yx^2 - (y + 3)x - 1, y \in \mathbb{Z}$$

$$| \quad \mathbf{T} = t_0 + t_1x + x^2$$

$$| \quad \text{define } L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ ' _0 & ' _1 & ' _2 & 1 & 0 & 0 \\ 0 & ' _0 & ' _1 & ' _2 & 1 & 0 \\ 0 & 0 & ' _0 & ' _1 & ' _2 & 1 \end{bmatrix}$$

- | LLL(L) outputs a short vector r , linear combination of L 's rows. $r = {}_0p + {}_1px + {}_2T + {}_3' + {}_4x' + {}_5x^2'$.
 $r = r_0 + \dots + r_5x^5; \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$

Sub-eld Simplification + LLL

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

$$| \quad ' = x^3 - yx^2 - (y + 3)x - 1, y \in \mathbb{Z}$$

$$| \quad \mathbf{T} = t_0 + t_1x + x^2$$

$$| \quad \text{define } L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ ' _0 & ' _1 & ' _2 & 1 & 0 & 0 \\ 0 & ' _0 & ' _1 & ' _2 & 1 & 0 \\ 0 & 0 & ' _0 & ' _1 & ' _2 & 1 \end{bmatrix}$$

| LLL(L) outputs a short vector r , linear combination of L 's rows. $r = {}_0p + {}_1px + {}_2T + {}_3' + {}_4x' + {}_5x^2'$.

$$r = r_0 + \dots + r_5x^5; \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$$

| $\text{Norm}_f(r) = O(p^2)$ of ≈ 340 bits instead of $O(p^3)$ of 508 bits

Sub-eld Simplification + LLL

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

$$\lfloor \quad ' = x^3 - yx^2 - (y + 3)x - 1, y \in \mathbb{Z}$$

$$\lfloor \quad \mathbf{T} = t_0 + t_1x + x^2$$

$$\lfloor \quad \text{define } L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ t_0 & t_1 & 1 & 0 & 0 & 0 \\ ' _0 & ' _1 & ' _2 & 1 & 0 & 0 \\ 0 & ' _0 & ' _1 & ' _2 & 1 & 0 \\ 0 & 0 & ' _0 & ' _1 & ' _2 & 1 \end{bmatrix} \quad \begin{array}{l} (p) = 0 \in \mathbb{F}_{p^n} \\ T \\ (') = 0 \in \mathbb{F}_{p^n} \end{array}$$

$\lfloor \quad \text{LLL}(L)$ outputs a short vector r , linear combination of L 's rows. $r = {}_0p + {}_1px + {}_2T + {}_3' + {}_4x' + {}_5x^2'$.

$$r = r_0 + \dots + r_5x^5; \quad \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/3})$$

$\lfloor \quad \text{Norm}_f(r) = O(p^2)$ of ≈ 340 bits instead of $O(p^3)$ of 508 bits

$\lfloor \quad \log(r) = \log(T) \bmod \cdot$ because $(r) = {}_2T$ with ${}_2 \in \mathbb{F}_p$

Initial Splitting step complexity

Given a target $T_0 \in \mathbb{F}_{p^n}^*$, and g a generator of $\mathbb{F}_{p^n}^*$

repeat

1. take t at random in $\{1; \dots; \ell - 1\}$ and set $T = g^t T_0$
2. factorize $\text{Norm}(\mathbf{T})$

until it is B_1 -smooth: $\text{Norm}(\mathbf{T}) = \prod_{q_i \leq B_1} q_i \prod_{p_i \leq B_0} p_i$

L -notation: $\mathbf{c} > 0$,

$$Q = p^n; \quad L_Q[1=3; \mathbf{c}] = e^{(\mathbf{c} + o(1))(\log Q)^{1/3}} (\log \log Q)^{2/3};$$

Norm factorization done with ECM method, in time $L_{B_1}[1=2; \sqrt{2}]$

Lemma (Initial Splitting step running-time)

If $\text{Norm}(\mathbf{T}) \leq Q^e$, 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).

Sub-eld Simplification + LLL

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

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

MNT example: $\log Q = 508$ bits

	$\text{Norm}_f(\mathbf{T})$		$\text{Norm}_g(\mathbf{T})$		$L_Q[1=3; c]$		$q_i \leq B_1 =$
	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}
Sub-eld	$Q^{2/3}$	340	Q	508	1.26	2^{42}	2^{69}

Combined with Pomerance Early Abort Strategy, we obtained a 54-bit smooth initial splitting for $g^{35313} T_0$ and a 59-bit smooth initial splitting for g^{52154} in 32 core-hours.

The descent took 13.4 and 10.7 core hours.

With more sub elds: e.g. \mathbb{F}_{p^6}

JLSV1 polynomial selection: $\|f\|_\infty = \|g\|_\infty = \sqrt{p}$,

$\deg f = \deg g = 6$

$\text{Norm}_f(\mathbf{T}_0) = \|f\|_\infty^{\deg \mathbf{T}} \|\mathbf{T}\|_\infty^6$

Let $\{1; U; U^2\}$ be a polynomial basis of $\mathbb{F}_{p^3} \subset \mathbb{F}_{p^6}$, e.g. $U = g^{1+p^3}$

$$E = \begin{bmatrix} * & * & * & 1 & 0 & 0 \\ * & * & * & * & 1 & 0 \\ * & * & * & * & * & 1 \end{bmatrix} = \text{RowEchelonedForm} \left(\begin{bmatrix} T \\ UT \\ U^2T \end{bmatrix} \right)$$

| E obtained with \mathbb{F}_p -linear combinations of $\{T; UT; U^2T\}$

| for each row $\leftrightarrow r_i \in \mathbb{Z}[x]$,

$$r_i = \underbrace{0T + 1UT + 2U^2T}_{=u_i \in \mathbb{F}_{p^3}} T$$

$$\log_g(r_i) = \log_g(T) \text{ mod } \cdot$$

Sub-eld Simplification + LLL in \mathbb{F}_{p^6}

We want to reduce $\|\mathbf{T}\|_\infty$.

| $\mathbf{T} = t_0 + t_1x + x^2$

| define $L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 & 0 \\ * & * & * & 1 & 0 & 0 \\ * & * & * & * & 1 & 0 \\ * & * & * & * & * & 1 \end{bmatrix}$

| $\text{LLL}(L) \rightarrow r = r_0 + r_1x + r_2x^2 + r_3u_0T + r_4u_1T + r_5u_2T$.
 $r = r_0 + \dots + r_5x^5; \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/2})$

Sub-eld Simplification + LLL in \mathbb{F}_{p^6}

We want to reduce $\|\mathbf{T}\|_\infty$.

| $\mathbf{T} = t_0 + t_1x + x^2$

| define $L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 & 0 \\ * & * & * & 1 & 0 & 0 \\ * & * & * & * & 1 & 0 \\ * & * & * & * & * & 1 \end{bmatrix}$

| $\text{LLL}(L) \rightarrow r = r_0 + r_1x + r_2x^2 + r_3u_0T + r_4u_1T + r_5u_2T$.
 $r = r_0 + \dots + r_5x^5; \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/2})$

| $\text{Norm}_f(r) = O(p^{3+\frac{5}{2}}) = O(Q^{11/12})$ of ≈ 470 bits instead of $O(p^3)$ of 508 bits

Sub-eld Simplification + LLL in \mathbb{F}_{p^6}

We want to reduce $\|\mathbf{T}\|_\infty$.

$$\mathbf{T} = t_0 + t_1x + x^2$$

$$\text{define } L = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 & 0 \\ * & * & * & 1 & 0 & 0 \\ * & * & * & * & 1 & 0 \\ * & * & * & * & * & 1 \end{bmatrix} \quad (p) = 0 \in \mathbb{F}_{p^n}$$

$$\begin{matrix} u_0 T \\ u_1 T \\ u_2 T \end{matrix}$$

$$\text{LLL}(L) \rightarrow r = {}_0p + {}_1px + {}_2px^2 + {}_3u_0T + {}_4u_1T + {}_5u_2T.$$

$$r = r_0 + \dots + r_5x^5; \quad \|r_i\|_\infty \leq C \det(L)^{1/6} = O(p^{1/2})$$

$$\text{Norm}_f(r) = O(p^{3+\frac{5}{2}}) = O(Q^{11/12}) \text{ of } \approx 470 \text{ bits instead of } O(p^3) \text{ of } 508 \text{ bits}$$

$$\log(r) = \log(T) \pmod{\cdot} \text{ because}$$

$$(r) = ({}_3u_0 + {}_4u_1 + {}_5u_2)T \text{ with } {}_{i+3}u_i \in \mathbb{F}_{p^3}$$

Theorem

Let $T \in \mathbb{F}_{p^n}^*$ an element which is not in a proper subfield of \mathbb{F}_{p^n} . We want to compute its discrete logarithm modulo a (large) prime ℓ , where $\ell \mid n(p)$. Let f, R_f given by a polynomial selection method. Let d be the largest divisor of n , $d < n$ and $d = 1$ if n is prime.

Then there exists a preimage \mathbf{T} in $\mathbb{Z}[x] = (f(x))$ of $T \in \mathbb{F}_{p^n}^*$, such that $\log(\mathbf{T}) \equiv \log T \pmod{\ell}$ and whose norm in R_f is bounded by $O(q^e)$, where $q = p^n$ and q^e equals

1. $q^{1-d/n}$ for the GJL, Conjugation, Joux-Pierrot, Sarkar-Singh and TNFS-like methods (and for all the possible methods where $\|f\|_\infty = o(p)$);
2. $q^{\frac{3}{2} - \frac{d}{n} - \frac{1}{2n}}$ for the JLSV1 method;
3. $q^{2 - \frac{d}{n} - \frac{2}{D+1}}$ for the JLSV2 method, where D is the degree of g .

Small characteristic $\mathbb{F}_{2^{4m}}$ and $\mathbb{F}_{3^{6m}}$

Same idea as for \mathbb{F}_{p^6} but without LLL:

The largest sub field of $\mathbb{F}_{2^{4m}}$ is $\mathbb{F}_{2^{2m}}$, let $d = 2m$: Compute two \mathbb{F}_2 -linear Gaussian eliminations

$$A = \begin{bmatrix} T \\ UT \\ \vdots \\ U^{d-1}T \end{bmatrix} \rightarrow \begin{bmatrix} * & * & * & 0 & \cdots & 0 \\ 0 & * & * & * & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & * & * & * \end{bmatrix}$$

Each row r_i corresponds to $u_i T_i = u_i X^i \cdot T'_i$, where T'_i is of degree $n=2 = 2m$ and $\log_g(X^i T'_i) = \log_g T \pmod{\cdot}$

- | Need one poly of degree $n=2$ to be B -smooth instead of two polys
- | cost of Gaussian elimination shared over $n=2$ tests
- | Magma implementation for $\mathbb{F}_{3^{6 \cdot 509}}$ and $\mathbb{F}_{2^{12 \cdot 367}}$ available

Asymptotic complexity

L -notation: $c > 0$,

$$Q = p^n; \quad L_Q[1=3; c] = e^{(c+o(1))(\log Q)^{1/3}} (\log \log Q)^{2/3}$$

Set $B = \log L_{2^n}[2=3;]$

- | Blake{Fuji-Hara{Mullin{Vanstone Waterloo alg.: $L_{2^n} \left[\frac{1}{3}; \frac{1}{3^\gamma} \right]$
- | Sub eld alg.: $L_{2^n} \left[\frac{1}{3}; \frac{d-1}{d} \frac{1}{3^\gamma} \right]$ where d is the largest proper divisor of n (best case: $d = n-2$, $L_{2^n} \left[\frac{1}{3}; \frac{1}{2} \frac{1}{3^\gamma} \right]$)

For $\mathbb{F}_{3^6 \cdot 509} = \mathbb{F}_{3^6}[x] = (I(x))$: T_0 is a degree 508 polynomial over \mathbb{F}_{3^6} .

- | We found a 30-smooth polynomial over \mathbb{F}_{3^6}
- | much less elements to "descent"
- | improve also the width and depth of "descent" tree

Thank you!

Pre-print available soon.