Breaking DLP in GF(p^5) using 3-dimensional sieving

Laurent Grémy[1], Aurore Guillevic[1] and François Morain[2]

[1]CNRS/Inria/Université de Lorraine, LORIA UMR 7503, Nancy, France
[2]École Polytechnique/CNRS-LIX UMR 7161/Université Paris-Saclay, Palaiseau, France

Abstract

We report on a discrete logarithm computation in $\mathbb{F}_{p^5}$ for a 20-decimal digit prime, using the number field sieve algorithm (NFS-DL), and a relation collection phase over degree-two polynomials, instead of the more classical degree-one case.

We select the 20-decimal-digit (dd) prime $p = 31415926535897932429 = \lfloor 10^{19}\pi \rfloor + 45$, and consider the finite field $\mathbb{F}_{p^5}$ where

$$p^5 - 1 = (p-1) \cdot 11 \cdot 101 \cdot 191 \cdot 7363691 \cdot 330316562322043642598658456150 41 \cdot \ell$$

where $\ell = 188723576570256606887670701559269 11$ is a 35dd prime. Our aim is to compute discrete logarithms in the prime order subgroup of $\mathbb{F}_{p^5}^*$ of cardinality ℓ. Since the extension degree is prime, the ExTNFS [4] algorithm is restrained to its TNFS original form [1], with R a degree-5 number field above $\mathbb{Q}$ and sieving in dimension 10, or to the high degree variant of NFS-DL (NFS-HD), with $R = \mathbb{Q}$ (no tower). We implemented the latter option: NFS-HD. Our computations were done using `Xeon CPU E5520 @ 2.27GHz` cores.

1 Polynomial selection

The Joux–Lercier–Smart–Vercauteren–1 (JLSV$_1$) and generalized Joux–Lercier (GJL) are expected to be the best polynomial selections, according to Figure 1. We get the best pair of polynomials (f_0, f_1) using the JLSV$_1$ method (see Section B).

To improve the running-time of the sieving, we searched for pairs of polynomials providing good smoothness propoerties. We precomputed many irreducible polynomials f_0 of degree 5 in the Brumer family (see Section B.3), with $|a| \leq 8$ and b of 30 to 34 bits. We selected the ones such that $|a| \leq 4$ and b is 33-bit long, and satisfying $\alpha(f_0) \leq -4.0$ (α is defined in [2]). We obtained 8798 polynomials f_0 in 9 core-days on an `Xeon E5-2609 @ 2.40GHz`. Then we computed f_1 for these good f_0. It took again 6 core-days. We get:

$$f_0 = x^5 - 5x^4 - 5368736472x^3 + 10737472959x^2 - 5368736477x - 2,$$
$$f_1 = 5851642500x^5 - 29258212500x^4 + 25042672429x^3 + 37689292642x^2 - 4215540071x - 11703285000.$$

2 Relation collection

2.1 Three-dimensional relation collection

The relation collection was performed using the special-$\mathfrak{q}$ sieve [3] and the three-dimensional sieving algorithms described in [2]. The smoothness bounds are set to 2^{25}, and the cofactorization is performed if on both sides, the remaining norms are smaller than 2^{80}, that is slightly more than three times the size of the large prime involved in the factorization of the norms. The special-$\mathfrak{q}$s are set on side 1 and have norm in $]2^{21}, 2^{23.75}[$: inside a special-$\mathfrak{q}$-lattice, we sieve on both sides the ideals of inertia degree 1 that have a norm bellow 2^{21}. There are 156,186 such ideals on side 0 and 155,192 on side 1. There are fewer ideals of inertia degree 2 of norm below than the smoothness bounds (759 on side 0 and 778 on side 1), and rare projective ideals (6 on side 1, which is coherent with the factorization of the leading coefficient of f_1, that is $5851642500 = 2^2 \cdot 3^4 \cdot 5^4 \cdot 11 \cdot 37 \cdot 71$). The latter two set of ideals were ignored during the sieving step.

In each special-$\mathfrak{q}$-lattice, we consider a sieving region containing 2^{25} elements $\boldsymbol{c}$ of the lattice, where the coordinates $(\boldsymbol{c}[0], \boldsymbol{c}[1], \boldsymbol{c}[2])$ are in the sieving region $[-2^8, 2^8[\times[-2^8, 2^8[\times[0, 2^7[$. The time per special-$\mathfrak{q}$ during the computation was between 15.37 seconds and 93.87 seconds, and the largest number of relations per special-$\mathfrak{q}$ is 34. The cost to find the 6,171,924 relations was about 359 CPU days.

2.2 Two-dimensional relation collection

For comparison purposes, a second relation collection was performed using a special-$\mathfrak{q}$ strategy but with a two-dimensional sieving, using the CADO-NFS implementation. For this computation, we use a polynomial pair coming from the JLSV_0 polynomial selection (see Section B.2).

The special-$\mathfrak{q}$s are set on side 0 and have norm in $]2^{24.25}, 2^{26}[$: inside a special-$\mathfrak{q}$-lattice, we sieve on both sides the ideals of inertia degree 1 that have a norm bellow $2^{24.25}$. The smoothness bounds are set to 2^{26} on side 0 and 2^{27} on side 1, and the cofactorization is performed if on both sides, the remaining norms is less than 2^{52} on side 0 and 2^{54} on side 1, that is 2 large primes on both sides.

In each special-$\mathfrak{q}$-lattice, we consider a sieving region that contains 2^{29} elements $\boldsymbol{c}$ of the lattice, where the coordinates $(\boldsymbol{c}[0], \boldsymbol{c}[1])$ are in $[-2^{14}, 2^{14}[\times[0, 2^{14}[$. The time per special-$\mathfrak{q}$ during the computation was between 0.54 second and 18.14 seconds, and the largest number of relations per special-$\mathfrak{q}$ is 21. The cost to find the $10,458,616$ relations was about 375 CPU days. It is smaller than the $11,561,362 = \pi(2^{26}) + \pi(2^{27})$ that are almost needed, where π is the prime counting function. We also provide in Section C different parameters for

which we estimate that a complete set of relations would be obtained, using a two-dimensional relation collection.

The best running time for the relation collection and the smallest matrix are reached by using the three-dimensional relation collection, as shown in Table 2.

3 Filtering

On the 6,171,924 relations produced with the three-dimensional relation collection, 4,999,773 were unique, and this led to a 1,489,631×1,489,625 matrix after singleton removal, reduced to a final 490,307×490,301 matrix after more intensive filtering.

4 Linear algebra

The linear algebra step is performed using the block-Wiedemann algorithm. The parameters used were $m = 12$ and $n = 6$. Then 6 parallel jobs were used, one for each of the 6 sequences. Each parallel job used a 2×2 node topology, each node having 8 cores.

The time to compute the Krylov subspaces was 237 hours, then 4 hours for the linear generator and 35 hours to create the solutions from the generator. 3,787,509 logs were reconstructed (out of at most 4,128,343 possible logs).

5 Individual logarithms

To completely validate our work, we report the computation of an individual logarithm. We first note that $h = X + 1$ generates the whole multiplicative group $\mathbb{F}_{p^5}^*$. We find that h lifts to K_0 as $z + 1$ of norm $2^2 \cdot 3^2 \cdot 5^2 \cdot 23 \cdot 1037437$, corresponding to the factorization in O_0

$$(z+1) = \langle 3, x+4\rangle^2 \langle 2, x^3+x^2+x+3\rangle \langle 5, x+6\rangle^2 \langle 23, x+1\rangle \langle 1037437, x+1\rangle$$

All logs were known from the first phase (including that of the degree-two ideal above 2), but that of norm 1037437 that we needed to descend. Finally,

$$\mathrm{vlog}(h) = 69480237664316728325370489421116 17 \bmod \ell.$$

Now, consider the target made of the decimals of π

$$t = 3141592653589793238X^4 + 4626433832795028841X^3 + 971693993751058209 7X^2 + 4944592307816406286X + 2089986280348253421.$$

After 20,000 seconds we find that the lift of $h^{9002259}\, t$ has a smooth norm and corresponding ideal factorization

$$\langle 2, x^3+2\rangle^2 \langle 41, x+11\rangle \langle 43, x+21\rangle \langle 3471899, x+3245828\rangle$$
$$\langle 37276061, x+17122378\rangle \langle 3115088134901, x+1265257252254\rangle$$
$$\langle 366996697855783, x+268803256185002\rangle$$
$$\langle 377568478750783, x+9644708969240\rangle$$
$$\langle 4811620104558151, x+2380670555180752\rangle$$
$$\langle 120866356812660071, x+98064663938425303\rangle$$
$$\langle 4133950459282418267, x+1195413435698177697\rangle.$$

All ideals of norm > 37276061 had to be re-expressed in terms of prime ideals of smaller norm. Contrary to the relation collection step, we can re-express the elements by looking for a relation given by a degree 1 polynomial, and therefore use the program `las_descent` of the CADO-NFS package [7], which took 11,958 seconds, finally leading to

$$\mathrm{vlog}(t) = 284270745040684398905938148353673 8 \bmod \ell.$$

Note that we could use ideals of inertia degree larger than 1 whose logarithms would be known from the first step, though they rarely pop up at this stage, except for the smallest ones. Re-express these ideals will require to find a relation given by a polynomial of degree at most 2.

References

[1] Barbulescu, R., Gaudry, P., Kleinjung, T.: The Tower Number Field Sieve. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9453, pp. 31–55. Springer (2015)

[2] Gaudry, P., Grémy, L., Videau, M.: Collecting relations for the Number Field Sieve in $GF(p^6)$. LMS Journal of Computation and Mathematics 19, 332–350 (2016)

[3] Hayasaka, K., Aoki, K., Kobayashi, T., Takagi, T.: An experiment of number field sieve for discrete logarithm problem over GF(p^{12}). In: Fischlin, M., Katzenbeisser, S. (eds.) Number Theory and Cryptography. LNCS, vol. 8260, pp. 108–120. Springer (2013)

[4] Kim, T., Barbulescu, R.: Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 543–571. Springer (2016)

[5] Lehmer, E.: Connection between Gaussian periods and cyclic units. Math. Comp. 50(182), 535–541 (April 1988), `http://www.ams.org/journals/mcom/1988-50-182/S0025-5718-1988-0929551-0/`

[6] Schoof, R., Washington, L.C.: Quintic polynomials and real cyclotomic fields with large class number. Math. Comp. 50(182), 543–556 (April 1988), `http://www.ams.org/journals/mcom/1988-50-182/S0025-5718-1988-0929552-2`

[7] The CADO-NFS Development Team: CADO-NFS, An Implementation of the Number Field Sieve Algorithm (2017), `http://cado-nfs.gforge.inria.fr/`, development version

A Summary of the computation

We summarize in Table 1 the running time of the four main steps of our computation using a three-dimensional relation collection.

Part	Time (CPU days)
Polynomial selection	15
Relation collection	359
Linear algebra	11.5
Individual logarithm	0.37
Total	386

Table 1: Timing for each part of the computation.

B Polynomial selections

In this appendix, the computation of the α values is done using the three-dimensional version defined in [2].

B.1 Comparison of polynomial selections

We have considered the different polynomial selections:

- Joux–Lercier–Smart–Vercauteren–0 (JLSV_0);
- Joux–Lercier–Smart–Vercauteren–1 (JLSV_1);
- generalized Joux–Lercier (GJL);
- conjugation (Conj);
- Sarkar–Singh (SarSin).

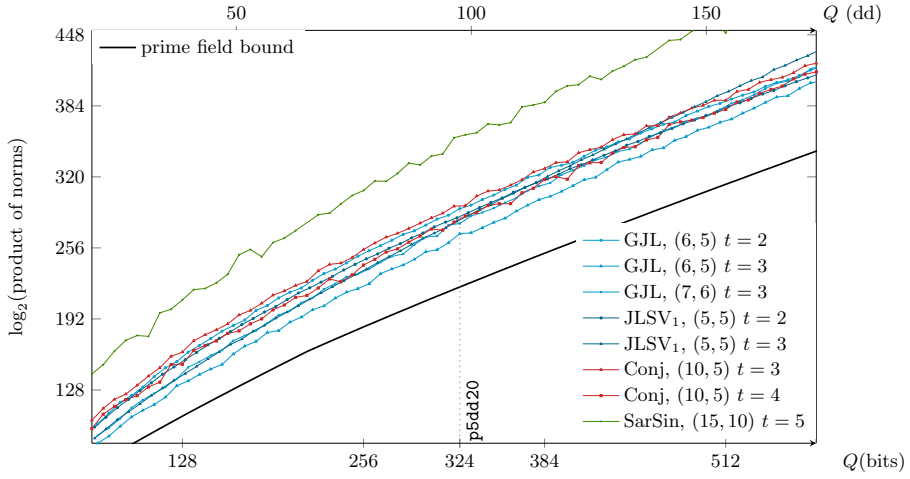

Figure 1: Average norm sizes for fixed searching space.

B.2 JLSV$_0$

The polynomials are chosen to be, using the JLSV$_0$ polynomial selection,

$$\begin{aligned} f_0 &= x^5 + 14x^4 - 7x^3 - 4x^2 - 4x + 15, \\ f_1 &= 2^8 f_0 + p \\ &= 256x^5 + 3584x^4 - 1792x^3 - 1024x^2 - 1024x + 31415926535897936269. \end{aligned}$$

We get $\alpha(f_0) = -2.0$ and $\alpha(f_1) = -3.5$.

B.3 JLSV$_1$

With the polynomials defined in Section 1, we get $\alpha(f_0) = -4.0$ and $\alpha(f_1) = -8.3$. To build these polynomials, we have chosen the Brumer degree-5 parameterized polynomial family in $\mathbb{Q}_{a,b}[X]$:

$$\mathrm{Bru}(a, b; X) = X^5 + (a-3)X^4 - (a-b-3)X^3 + (a^2 - a - 2b - 1)X^2 + bX + a$$

whose Galois group is the dihedral group D_5 of order 10. We took the linear parameter b as the JLSV$_1$ parameter, and set a to a small value, in $[-4, 4]$ in our experiments.

There exists a parameterized cyclic family: Lehmer's family proposed in [5]

$$\begin{gathered} X^5 + t^2X^4 - (2t^3 + 6t^2 + 10t + 10)X^3 + (t^4 + 5t^3 + 11t^2 + 15t + 5)X^2 \\ +(t^3 + 4t^2 + 10t + 10)X + 1. \end{gathered}$$

The Galois automorphism

$$x \mapsto \frac{(t+2) + tx - x^2}{1 + (t+2)x}$$

was computed by Schoof and Washington [6, eq. (3.2)]. Despite the explicit automorphism, this family is not suited to the JLSV$_1$ method because the parameter t in the coefficients of the polynomial, is not linear: t, t^2, t^3, t^4 appear in the coefficients of $f_t(X)$.

B.4 GJL

The best pair of polynomials (f_0, f_1) we get using the GJL method is:

$$\begin{aligned} f_0 &= 2x^6 + 3x^5 - x^4 + 2x^3 - 3x^2 - 2x - 3, \\ f_1 &= 4682288594364150x^5 + 10520016140415817x^4 - 17832477142237943x^3 \\ &\quad - 15171722661935206x^2 + 1592160578567340x + 1708993376270808. \end{aligned}$$

We get $\alpha(f_0) = -0.4$ and $\alpha(f_1) = -4.5$.

C Summary of parameters for the relation collection

In this section, we summarized the parameters we use for the real computation and the results of some experiments with other parameters without performing the whole computation, but by inferring the results using a sample of some special-$\mathfrak{q}$s. We denote by JLSV_1 the polynomials described in Section 1 and by JLSV_0 the polynomials of Section B.2.

Dimension	2	2	2	3
Polynomial	JLSV_0	JLSV_0	JLSV_1	JLSV_1
Sieving bounds	$2^{24.25}$, $2^{24.25}$	$2^{24.25}$, $2^{24.25}$	$2^{24.25}$, $2^{24.25}$	2^{21}, 2^{21}
Smoothness bounds	2^{26}, 2^{27}	2^{26}, 2^{27}	2^{26}, 2^{26}	2^{25}, 2^{25}
Thresholds	2^{52}, 2^{54}	2^{52}, 2^{54}	2^{52}, 2^{52}	2^{80}, 2^{80}
Special-$\mathfrak{q}$ side	0	1	1	1
Special-$\mathfrak{q}$-range	$]2^{24.25}, 2^{26}[$	$]2^{24.25}, 2^{27}[$	$]2^{24.25}, 2^{26}[$	$]2^{21}, 2^{23.75}[$
Sieving region	2^{29}	2^{29}	2^{29}	2^{25}
Mean of norms	$2^{118}, 2^{203}$	—	—	$2^{144}, 2^{128}$
Raw relations	$10,458,616$	—	—	$6,171,924$
Unique relations	$8,256,215$	$\approx 16,000,000$	$\approx 9,900,000$	$4,999,773$
Needed relations	$\approx 11,561,362$	$\approx 11,561,362$	$\approx 7,915,618$	$\approx 4,127,378$
Set of relations	Incomplete	Complete (probably)	Complete (probably)	Complete
Time (CPU days)	375	≈ 900	≈ 400	359

Table 2: Informations about the relation collections.

D Sage verification script

To verify our computation of the individual logarithm in Section 5, we provide Sage script in Listing 1 (tested using version `7.5.1`).

```
Q.<x> = ZZ[]

p = floor(10^19*pi) + 45; n = 5
ell = 188723576570256606887670701559269l1; cof = (p^n-1) // ell

f0 = x^5 - 5*x^4 - 5368736472*x^3 + 10737472959*x^2 - 5368736477*x - 2
f1 = 5851642500*x^5 - 29258212500*x^4 + 25042672429*x^3 + 37689292642*x^2 \
        - 4215540071*x - 11703285000
varphi = f0

Fpn.<T> = GF(p^n, modulus=varphi)

gen = T + 1
target1 = 3141592653589793238*T^4 + 4626433832795028841*T^3 \
        + 9716939937510582097*T^2 + 4944592307816406286*T \
        + 2089986280348253421

vlog_gen = 69480237664316728325370489421l1617
vlog_target1 = 28427074504068439890593814835367l38

assert(gen^(cof * vlog_target1) == target1^(cof * vlog_gen))
```

Listing 1: Sage verification script.