A SageMath program for recovering points of superelliptic curves over a prime finite field

Jaime Gutierrez

University of Cantabria, Santander 39005, Spain
jaime.gutierrez@unican.es

1 Introduction

We present a SageMath implementation and empirical results of the main algorithm provided in paper [10] for reconstructing points of superelliptic curves.

Here we consider the following computational problem: given the polynomial $Y^n + f(X) \in \mathbb{F}_p[X,Y]$ and approximations to $(v_0, v_1) \in \mathbb{F}_p^2$ where $v_1^n + f(v_0) \equiv 0 \bmod p$, reconstruct (v_0, v_1).

Its has applications to, and has been motivated by, the predictability problem for the linear congruential generator on elliptic and hyperelliptic curves (see [3,7,9]).

This problem is a particular case of the problem of finding small solutions of multivariate polynomial congruences, see for instance [4,5,6,8]. All of them are based on the so called lattice reduction techniques [11,12].

On the other hand, this problem is also a special case of obtaining which is the number of roots where the roots have certain restrictions, sometimes these kind of question has been called additive energy, the subject has been studied quite recently in [2,13]

2 The reconstructing algorithm

Given a prime p and a positive integer Δ with $p > \Delta \geq 1$, we say that an integer pair $(w_0, w_1) \in \mathbb{Z}^2$ is a $\Delta-$approximation to $(v_0, v_1) \in \mathbb{F}_p^2$ if there exist integers $\varepsilon_0, \varepsilon_1$ satisfying:

$$|\varepsilon_0|, |\varepsilon_1| \leq \Delta, \ w_0 + \varepsilon_0 = v_0, \ w_1 + \varepsilon_1 = v_1.$$

We are considering irreducible bivariate polynomials $H_{(n,m,f)}(X,Y) \in \mathbb{F}_p[X,Y]$ of the form $Y^n + f(X)$ and the equation:

$$H_{(n,m,f)}(X,Y) = 0 \qquad (1)$$

where n, m are positive integers such that $nm > 1$, and $f = f(X) \in \mathbb{F}_p[X]$ is a monic univariate polynomial of degree m, i.e.,

$$f = X^m + a_{m-1}X^{m-1} + \cdots + a_1 X + a_0.$$

Theorem 1 **([10]).** *With the above notations and definitions, there exists a set $\mathcal{V}(\Delta; f) \subseteq \mathbb{F}_p$ of cardinality*

$$\#\mathcal{V}(\Delta; f) = O(A(n,m)\Delta^{\lambda_{n,m}})$$

where

- *If $m \geq n$*

$$A(n,m) = m^2(2m+2n)^{(m+n-1)/2}$$

  and

$$\lambda_{n,m} = \frac{m(m+1)+n(n-1)}{2}$$

- *If $n \geq m$*

$$A(n,m) = n^2(2m+2n)^{(m+n-1)/2}$$

  and

$$\lambda_{n,m} = \frac{n(n+1)+m(m-1)}{2}$$

with the following property: whenever $v_0 \notin \mathcal{V}(\Delta; f)$ then, given a $\Delta-$approximation (w_0, w_1) to a point (v_0, v_1) of the polynomial $H_{(n,m,f)}(X,Y)$ one can recover (v_0, v_1) in deterministic polynomial time in m, n and $\log p$.

An outline of the algorithm given in the proof of this Theorem goes as follows. The algorithm is divided into two stages.

- **Stage 1**: We construct a certain linear of system of congruences $\mathcal{LS}_{(n,m,f)}$ and the associated lattice $\mathcal{L}_{(n,m,f)}$ of dimension $m+n-1$; this lattice depends on the approximation (w_0, w_1). We also show that a certain vector $\mathbf{E}$ directly related to missing information about (v_0, v_1) is a very short vector. Now, we compute a solution $\mathbf{T}$ of the system of congruences $\mathcal{LS}_{(n,m,f)}$ in polynomial time using linear diophantine methods. Then apply (approximation) Closest Vector Problem to the vector $\mathbf{T}$ and lattice $\mathcal{L}_{(n,m,f)}$, using the algorithm in paper [1] and obtaining a vector $\mathbf{u}$ of the lattice $\mathcal{L}_{(n,m,f)}$.
- **Stage 2**: We show that $\mathbf{F} = \mathbf{T} - \mathbf{u}$ provides the required information about $\mathbf{E}$ for all (v_0, v_1) except when v_0 lies in a certain exceptional set $\mathcal{V}(\Delta; f) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{V}(\Delta; f) = O(A(n,m)\Delta^{\lambda_{n,m}})$ (which is defined as set of zeroes of a certain parametric family of 0-dimension bivariate polynomial ideals).

3 The SageMath implementation and empirical results

The input required by the above algorithm to recovering points of superelliptic curve include approximations to the point.

In the first case, a "bad" set of values for the component x_0 is described, proving that whenever that value lies outside the set, the algorithm works correctly. Furthermore, the size of the set is asymptotically bounded with $\Delta^{\lambda_{n,m}}$.

We have performed some numerical tests with a SageMath program. Firstly, we fixed the integers n and m and generate a superelliptic curve $H_{(n,m,f)}(X,Y) = 0$ over a prime finite field of a desired size by choosing randomly in $\mathbb{F}_p$ the parameters/coefficients of the univariate polynomial f to fix Eq. (1).

```
n=2; m=5; tolerance= (m*(m+1) +n*(n-1))/2; bits =1024
p=next_prime(ZZ.random_element(2**(bits)))
R.<x> = GF(p)[]
f= x**m
while 1:
    for i in range(m):
        f +=ZZ.random_element(p)*x**i
    if gcd(f, diff(f,x)) ==1:
        pass
    break
```

Then, we generate randomly a point in the curve by choosing their first coordinate and try to solve Eq. (1). We choose the quality of the approximation by selecting the number of digits of $p^{1/tolerance}$. For several approximations to the point are given as input to our algorithm, we construct the lattice $\mathcal{L}_{(n,m,f)}$:=`L_n_m(f)`, and the algorithm returns the number of the hits.

```
Digits= 4;  Bound= int((1./tolerance)*(10**Digits))/10**Digits
Delta = int(p**Bound)
Number_test = 100; Number_hits =0
for i in range(Number_test):
    w0 = ZZ.random_element(-Delta+int(v0), Delta+int(v0))
    w1= ZZ.random_element(-Delta+int(v1),Delta+int(v1))
    A=matrix(ZZ, L_n_m(f))
    Base=basis(A.transpose().kernel())
    L=matrix(ZZ, [[Base[i][j] for j in range(m+n-1)] for i in range(m+n-1)])
    C=vector(ZZ,[Delta**(m-1)*((-f(x=w0)-w1**n)%p)]+[0]*(n+m-2))
    T=vector(ZZ,[solving_linear(A,C)[i] for i in range(n+m-1)])
    F=Approx_CVP(L,T)+T
    if (F[0]//Delta**(m-1)== v0-w0 or F[0]//Delta**(m-1)== w0-v0):
        Number_hits += 1
Number_hits
```

The above code includes the functions `solving_linear` to solve linear systems over the integers which we have implemented using the Hermite Normal Form of a matrix, and the function `Approx_CVP` that we have implemented the Babai's Nearest plane algorithm in [1].

We summarize its results in the following tables. We have selected primes of several sizes, and note the obtained success threshold. As we can see, $1/\lambda_{n,m}$ appears as the correct threshold:

- $n = 1, m = 5$, $1/\lambda_{1,5} = 1/15 = 0.066666$

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.65	0.066	0.0664	0.0666

- $n = 2, m = 3$, $1/\lambda_{2,3} = 1/7 = 0.142857$

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.13	0.140	0.14	0.142

- $n = 2, m = 5$, $1/\lambda_{2,5} = 1/16 = 0.06250$

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.05	0.06	0.061	0.062

Another argument to show that the threshold is correct is the so-called Gaussian heuristic. The so-called "Gaussian heuristic" suggests that and s-dimensional lattice $\mathcal{L}$ with volume $vol(\mathcal{L})$ is unlikely to have a nonzero vector which is substantially shorter than $vol(\mathcal{L})^{1/s}$.

The lattice $\mathcal{L}_{(n,m,f)}$ has volumen the product of the modulo integers, that is,

$$vol(\mathcal{L}_{(n,m,f)}) = p\Delta^{\frac{m(m-1)+(n-1)(2m-n)}{2}}$$

Since the dimension of the lattice $\mathcal{L}_{(n,m,f)}$ is $m + n - 1$. Then, vector $\mathbf{E}$ is likely to be the one founded whenever

$$\Delta^m < p^{1/(m+n-1)} \Delta^{\frac{m(m-1)+(n-1)(2m-n)}{2(m+n-1)}} \implies \Delta < p^{1/\lambda_{n,m}}.$$

Which it is exactly the same bound provided in the Theorem 1.

References

1. László Babai, On Lovász' lattice reduction and the nearest lattice point problem, *C*ombinatorica, **6(1)** (1986), 1–13.
2. Roger C Baker, Marc Munsch, and Igor E Shparlinski, Additive energy and a large sieve inequality for sparse sequences, preprint, arXiv*2103.12659*.
3. L. Beshaj, J. Gutierrez and T. Shaska, Weighted greatest common divisors and weighted heights. *J.* Number Theory, **213** (2020), 319–346.
4. Johannes Blömer and Alexander May, A tool kit for finding small roots of bivariate polynomials over the integers, In *Advances in Cryptology (Eurocrypt 2005)*, pages 251–267. Springer-Verlag, 2005.
5. D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptology*, **10** (1997), 233–260.
6. Jean-Sébastien Coron, Finding small roots of bivariate integer polynomial equations: A direct approach, In Alfred Menezes, editor, *C*RYPTO, volume 4622 of *Lecture Notes in Computer Science*, pages 379–394. Springer, 2007.
7. G. Frey and T. Shaska, Curves, Jacobians and cryptography. Algebraic curves and their applications, 279–344, Contemp. Math., 724, Amer. Math. Soc., 2020.
8. Domingo Gómez and Jaime Gutierrez, Recovering zeros of polynomials modulo a prime, *M*ath. Comput., **83(290)** (2014), 2953–2965.
9. Jaime Gutierrez, Attacking the linear congruential generator on elliptic curves via lattice techniques, *C*ryptography and Communications, **12** (2022), 505–525.
10. Jaime Gutierrez, Reconstructing points of superelliptic curves over a prime finite field, *A*dvances in Mathematics of Communications, *doi 10.3934/amc.2022022*, 2022.
11. Antoine Joux and Jacques Stern, Lattice reduction: A toolbox for the cryptanalyst, *J.* Cryptology, **11(3)** (1998), 161–185.
12. A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász, Factoring polynomials with rational coefficients, *Math. Ann.*, **261** (1982), 515–534.
13. László Mérai and Igor E Shparlinski, Sparsity of curves and additive and multiplicative expansion of rational maps over finite fields. preprint, arXiv*1803.02165*.