A strategy for recovering roots of bivariate polynomials modulo a prime

Paula Bustillo, Domingo Gómez, Jaime Gutierrez, Álvar Ibeas
University of Cantabria,
Santander E–39071,
Spain
jaime.gutierrez@unican.es

Abstract

We show how, when given an irreducible bivariate polynomial with coefficients in a finite prime field and an approximation to one of its roots, one can recover that root efficiently, if the approximation is good enough. This result has been motivated by the predictability problem for non-linear pseudorandom number generators and other potential applications to cryptography.

1 Introduction

For a prime p, we denote by $\mathbb{F}_p$ the field with p elements and assume that it is represented by the set $\{0, 1, \ldots, p-1\}$. In particular, where obvious, we treat elements of $\mathbb{F}_p$ as integers in the above range.

Here we consider the following problem: given a bivariate polynomial $f(X,Y) \in \mathbb{F}_p[X,Y]$ and a point (w_0, w_1) whose components approximate those of $(v_0, v_1) \in \mathbb{F}_p^2$, where $f(v_0, v_1) = 0$, the goal is to recover (v_0, v_1).

The question has applications to, and has been motivated by, the predictability problem for non-linear pseudorandom number generators over $\mathbb{F}_p$ and the linear congruential generator on elliptic curves (see [2, 4, 5, 9, 11, 14, 16]).

The task we solve can be considered as a special case of the problem of finding small solutions of multivariate polynomial congruences. For polynomial congruences in one variable, an algorithm for solving this problem has been given by Coppersmith [6], see also [3, 8, 7, 12, 13]. However, in the general case only heuristic results are known. Here we are able to obtain rigorous results for a big class of irreducible bivariate polynomials modulo a prime number.

The remainder of the paper is structured as follows. We start with a very short outline of some basic facts about the closest vector problem in lattices in Subsection 2.1 and the number of $\mathbb{F}_q$-rational points on algebraic curves in Subsection 2.2. In Section 3 we formulate the algorithm and prove its correction on the average when the approximation is good enough. Finally, Section 4 analyzes the algorithm in a particular case: recovering roots for elliptic curve polynomials.

2 Preliminaries

2.1 Closest Vector Problem in Lattices

This brief introduction is given in order to keep this article auto-contained. For more details and references, we recommend consulting [10, 14, 18, 19, 20].

Let $\{\mathbf{b}_1, \dots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^r$. The set

$$\mathcal{L} = \{c_1\mathbf{b}_1 + \ldots + c_s\mathbf{b}_s \ : \ c_1, \dots, c_s \in \mathbb{Z}\}$$

is called an *s-dimensional lattice* with *basis* $\{\mathbf{b}_1, \dots, \mathbf{b}_s\}$.

One basic lattice problem is the *closest vector problem (CVP)*: given a basis of a lattice $\mathcal{L}$ in $\mathbb{R}^s$ and a vector $\mathbf{t}$ in $\mathbb{R}^s$, the goal consists in finding a vector in $\mathcal{L}$ whose distance to the target vector $\mathbf{t}$ is minimum. It is well-known that CVP is **NP**-hard when the dimension grows. However, CVP is solvable in polynomial time provided that the dimension of $\mathcal{L}$ is fixed (see [15], for example).

For the slightly weaker task of finding a vector whose distance to the target approximates the smallest possible, we use a result which follows from [1], and which is based on the celebrated *LLL algorithm* of Lenstra, Lenstra and Lovász [17].

Lemma 1 *There exists a polynomial time algorithm which, when given an s-dimensional lattice $\mathcal{L}$ and a vector $\mathbf{t} \in \mathbb{R}^r$, finds a lattice vector $\mathbf{u} \in \mathcal{L}$ satisfying the inequality*

$$\|\mathbf{t} - \mathbf{u}\| \leq \lambda_s \min\{\|\mathbf{t} - \mathbf{v}\| : \ \mathbf{v} \in \mathcal{L}\},$$

where

$$\lambda_s = 2^{s/2}.$$

2.2 Number of $\mathbb{F}_q$-rational Points on Plane Algebraic Curves

Our second basic result is an upper bound on the number of roots of a bivariate polynomial with coefficients in a finite field. We denote by $\mathbb{F}_q$ the finite field with $q = p^t$ elements.

Given a polynomial $f(X,Y) \in \mathbb{F}_q[X,Y]$ and a positive integer r, we denote by $N_{q^r}(f)$ the number of solutions of the equation $f(x,y) = 0$ in the finite field $\mathbb{F}_{q^r}$. We use the following well-known result (see for instance [21, 22]).

Suppose that f is an absolutely irreducible polynomial of total degree m. Then, the inequality

$$|N_{q^r}(f) - q^r| \leq c(f)q^{r/2}$$

holds for a certain function $c(f) < m^2$. As a consequence, we have the following:

Lemma 2 *Suppose that f is an absolutely irreducible polynomial of total degree $m > 1$. Then, the inequality*

$$M_{q^r}(f) \geq (q^r - c(f)q^{r/2})/m$$

is valid for the number $M_{q^r}(f) = \#\{x \in \mathbb{F}_{q^r} \mid \exists y \in \mathbb{F}_{q^r}, f(x,y) = 0\}$.

Proof. By the above result, a lower bound for the number of roots is

$$N_{q^r}(f) \geq q^r - c(f)q^{r/2}.$$

For any $a \in \mathbb{F}_{q^r}$, we have that $f(a,Y) \in \mathbb{F}_{q^r}[Y]$ has at most m roots. So, the following inequality holds:

$$mM_{q^r}(f) \geq N_{q^r}(f) \geq q^r - c(f)q^{r/2}$$

and finishes the proof. ■

3 Root Recovering Algorithm

In this section we formulate and prove our main result.

3.1 Algorithm Description

Given a positive integer Δ, we say that a pair $(w_0, w_1) \in \mathbb{Z}^2$ is a Δ-approximation to another pair $(v_0, v_1) \in \mathbb{F}_p^2$ if there exist integers $\varepsilon_0, \varepsilon_1$ satisfying $|\varepsilon_i| \leq \Delta$ and such that v_i is the residue class of $w_i + \varepsilon_i$ modulo p.

We consider a bivariate polynomial over the finite field with p elements:

$$f(X, Y) = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} a_{i,j} X^i Y^j \in \mathbb{F}_p[X, Y],$$

where $m_1 < p$ and $m_2 < p$. Assume that f has an unknown root $(v_0, v_1) \in \mathbb{F}_p$ for which we have a Δ-approximation $(w_0, w_1) \in \mathbb{Z}^2$. We derive a probabilistic algorithm (Algorithm 1) for recovering that root. The parameter Δ measures how well the value (w_0, w_1) approximates the root (v_0, v_1) and it is assumed to vary independently of p subject to satisfying the inequality $\Delta < p$ (and is not involved in the complexity estimates of our algorithms).

Using the notation ε_i for the approximation errors, as defined above, the Taylor expansion of f at (w_0, w_1) provides:

$$\sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \frac{f^{(i,j)}(w_0, w_1)}{i!j!} \varepsilon_0^i \varepsilon_1^j \equiv 0 \bmod p.$$

Our algorithm will try to find vector

$$\mathbf{e} := \left(\Delta^{m_1+m_2-i-j} \varepsilon_0^i \varepsilon_1^j \mid 0 \leq i \leq m_1,\ 0 \leq j \leq m_2,\ i+j > 0 \right),$$

which is a solution of the following linear system of congruences in $(m_1 + 1)(m_2 + 1) - 1$ variables:

$$\left\{ \begin{array}{rcl} \displaystyle\sum_{\substack{0 \leq i \leq m_1, 0 \leq j \leq m_2 \\ 0 < i+j}} \Delta^{i+j} \frac{f^{(i,j)}(w_0, w_1)}{i!j!} X_{i,j} & \equiv & -\Delta^{m_1+m_2} f(w_0, w_1) \bmod p \\ X_{i,j} & \equiv & 0 \bmod \Delta^{m_1+m_2-i-j}. \end{array} \right. \qquad (1)$$

Algorithm 1: Recovering algorithm

```
Input: $(f, \Delta, w_0, w_1)$ such that $(w_0, w_1)$ is a $\Delta$-approximation to a
       root $(v_0, v_1)$ of $f$.
Output: $(v_0, v_1)$
Compute a solution $\mathbf{h}$ of (1) with small Euclidean norm.
$v_0' \leftarrow w_0 + h_{1,0}/\Delta^{m_1+m_2-1}$
$v_1' \leftarrow w_1 + h_{0,1}/\Delta^{m_1+m_2-1}$
$a \leftarrow f(v_0', v_1')$
if $a = 0$ then
    return $(v_0', v_1')$
else
    return failure
end
```

The computation of a small solution of an unhomogeneous system of congruences is done by a polynomial time algorithm for the approximated version of CVP [1].

3.2 Algorithm Correctness

Recall that we define λ_s to be the approximation factor given in Lemma 1. Now we introduce a class of polynomials for which we will prove the correctness of the algorithm. We say that a bivariate polynomial of total degree m is in the class $\mathcal{C}$ if there exist indexes $i, j \in \{0, \ldots, m-1\}$ such that X^iY^{m-i} and $X^{j+1}Y^{m-j-1}$ occur in f and $X^{i+1}Y^{m-i-1}$ and X^jY^{m-j} do not. We are now ready to state the main theorem of the paper.

Theorem 3 *With the above notations and definitions, if $f(X,Y) \in \mathcal{C}$ is an irreducible polynomial with $m_1m_2 > 1$, then Algorithm 1 recovers (v_0, v_1) in deterministic polynomial time in m_1, m_2 and $\log p$ provided that v_0 does not lie in a certain set $\mathcal{V}(\Delta, f) \subseteq M_p(f) \subseteq \mathbb{F}_p$ of cardinality*

$$\#\mathcal{V}(\Delta, f) \ll (\sqrt{s}\lambda_s)^s \Delta^{\omega_{m_1,m_2}},$$

where $s = m_1m_2 + m_1 + m_2$ and

$$\omega_{m_1,m_2} = 2 + (m_1 + m_2)\frac{s-1}{2}.$$

Proof. Let $\mathcal{L}$ be the lattice associated to linear system of congruences (1), that is, $\mathcal{L}$ is the set of integer solutions

$$\mathbf{x} = (X_{i,j} \mid 0 \le i \le m_1,\ 0 \le j \le m_2,\ i+j > 0) \in \mathbb{Z}^s$$

satisfying:

$$\left\{ \begin{array}{rcl} \displaystyle\sum_{\substack{0 \le i \le m_1, 0 \le j \le m_2 \\ 0 < i+j}} \Delta^{i+j} \frac{f^{(i,j)}(w_0, w_1)}{i!j!} X_{i,j} & \equiv & 0 \bmod p \\ X_{i,j} & \equiv & 0 \bmod \Delta^{m_1+m_2-i-j}. \end{array} \right. \tag{2}$$

We compute a solution $\mathbf{t}$ of the linear system of congruences (1). Then, algorithm of Lemma 1 applied to vector $\mathbf{t}$ and lattice $\mathcal{L}$ returns a vector $\mathbf{u}$. We aim to show that $\mathbf{h} := \mathbf{t} - \mathbf{u}$ contains sufficient information about $\mathbf{e}$, provided that v_0 does not lie in the "bad" set $\mathcal{V}(\Delta, f)$ which we define below.

The vector

$$\mathbf{d} := \mathbf{e} - \mathbf{h} = \left(\Delta^{m_1+m_2-i-j} d_{i,j} \mid 0 \le i \le m_1,\ 0 \le j \le m_2,\ i+j > 0\right)$$

lies in $\mathcal{L}$ and using (2), we obtain:

$$\sum_{\substack{0 \le i \le m_1, 0 \le j \le m_2 \\ 0 < i+j}} \frac{f^{(i,j)}(w_0, w_1)}{i!j!} d_{i,j} \equiv 0 \bmod p. \tag{3}$$

On the other hand, the norm of vector $\mathbf{d}$ satisfies:

$$\|\mathbf{d}\| \le \|\mathbf{h}\| + \|\mathbf{e}\| \le (\lambda_s + 1)\|\mathbf{e}\| \le 2\sqrt{s}\lambda_s \Delta^{m_1+m_2}.$$

Hence,

$$\begin{array}{c} |d_{i,j}| \le 2\sqrt{s}\lambda_s \Delta^{i+j}, \\ 0 \le i \le m_1,\ 0 \le j \le m_2,\ i+j > 0. \end{array} \tag{4}$$

We remark that if $d_{1,0} \equiv d_{0,1} \equiv 0 \bmod p$, then the first two components of $\mathbf{h}$, i.e., $h_{1,0}$ and $h_{0,1}$ contain the approximation errors. It implies that we can recover (v_0, v_1). Hence, we may assume that either $d_{1,0}$ or $d_{0,1}$ is non-zero modulo p.

Substituting $w_0 = X - \varepsilon_0, w_1 = Y - \varepsilon_1$ in (3), we obtain a bivariate polynomial

$$g(X,Y) = \sum_{i=0}^{m_1-1} \sum_{j=0}^{m_2-1} b_{i,j} X^i Y^j \in \mathbb{F}_p[X,Y],$$

where $b_{i,j} \in \mathbb{Z}[\varepsilon_0, \varepsilon_1, d_{1,0}, \ldots, d_{m_1,m_2}]$, verifying: $g(v_0, v_1) = 0$. Moreover, the fact that f lies in class $\mathcal{C}$ and not both of $d_{1,0}$, $d_{0,1}$ are zero implies that $g(X,Y)$ is not the zero polynomial modulo p. Now, we consider the polynomial system in $\mathbb{F}_p$:

$$\begin{cases} g(X,Y) & \equiv 0 \mod p \\ f(X,Y) & \equiv 0 \mod p \end{cases} \tag{5}$$

Then, for every choice of $\varepsilon_0, \varepsilon_1$ and vector $\mathbf{d}$ with not both $d_{0,1}$, $d_{1,0}$ zero, only a constant number of values v_0 are posible. This is because the classical Bézout Theorem for algebaic curves applies. We note that $f(X,Y)$ is an irreducible polynomial and $g(X,Y)$ is not a multiple of f. Then, the number of the solutions of system (5) is at most $(m_1 + m_2)^2$. We place any solution v_0 to (5) for any possible values of $d_{i,j}$ and $\varepsilon_0, \varepsilon_1$ into the set $\mathcal{V}(\Delta, f)$. We need to show that the cardinality of $\mathcal{V}(\Delta, f)$ is as claimed in the statement of the theorem.

By the bounds obtained in (4), the total number of possible choices for the integers $\varepsilon_0, \varepsilon_1$ and $d_{i,j}, i = 0, \ldots, m_1, j = 0, \ldots, m_2$ is at most:

$$4\Delta^2 \quad \times \prod_{\substack{0 \le i \le m_1, 0 \le j \le m_2 \\ 0 < i+j}} (4\sqrt{s}\lambda_s \Delta^{m_1+m_2-i-j}) \le (16\sqrt{s}\lambda_s)^s \Delta^{\omega_{m_1,m_2}},$$

where

$$\omega_{m_1,m_2} = 2 + (m_1 + m_2)\frac{s-1}{2}.$$

To finish the proof, we note that $\mathcal{L}$ is defined using given information, and recall that the approximated version of the closest vector problem can be solved in deterministic polynomial time in the bit size of the given lattice basis and in the lattice dimension.

■

The quality of the approximation (w_0, w_1) becomes a measure of the success probability of the algorithm. A "bad" set of values for the component v_0 has been described, proving that whenever that value lies outside the set, the algorithm works correctly. The size of the set is asymptotically $(\sqrt{s}\lambda_s)^s \Delta^{\omega_{m_1,m_2}}$. Therefore, when the polynomial f is absolutely irreducible, using Lemma 2, the error probability of Algorithm 1 is upper bounded by:

$$\frac{\#\mathcal{V}(\Delta, f)}{\#M_p(f)} \ll \frac{m(\sqrt{s}\lambda_s)^s \Delta^{\omega_{m_1,m_2}}}{p - c(f)p^{1/2}} \ll \frac{m}{p}(\sqrt{s}\lambda_s)^s \Delta^{\omega_{m_1,m_2}}.$$

This bound provides the threshold

$$\left(\frac{p}{m(\sqrt{s}\lambda_s)^s}\right)^{1/\omega_{m_1,m_2}}$$

for the tolerance Δ, in such a way that when the tolerance remains below that threshold and p is large enough the method is unlikely to fail.

We believe that the requierement for the polynomial to be absolutely irreducible is not neccesary for the algorithm to work. However, several aspects must be taken into account before considering the threshold expressed above for the error tolerance upon which the algorithm fails. Firstly, the constants hidden in the asymptotic reasoning (namely, the size of the prime p). Second, the threshold could be higher, as the "bad" set does not guarantee that the method need fail. And the most important fact, the behavior of the proposed algorithm has been studied for dense bivariate polynomials, but in many applications we need to work with a special bivariate polynomial and, maybe, for that polynomial we can obtain a much better tolerance. Finally, we have introduced somehow artificially the class $\mathcal{C}$ in order to prove the correctness of the algorithm but we also believe that this approach works for other polynomials. The following section will illustrate the last two remarks for elliptic curve equations.

4 Elliptic Curve Case

Let $E(\mathbb{F}_p)$ be an elliptic curve defined over $\mathbb{F}_p$ given by an *affine Weierstrass equation*, which for $\gcd(p, 6) = 1$ takes form

$$Y^2 = X^3 + aX + b, \tag{6}$$

for some $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$. We note that this polynomial does not belong to the class $\mathcal{C}$.

Lemma 4 *With the above conditions and definitions. Algorithm 1, with input polynomial (6), recovers (v_0, v_1) in polynomial time in $\log p$ provided that v_0 does not lie in a certain set $\mathcal{V}(\Delta, a) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{V}(\Delta, a) \ll \Delta^{27}$.*

Proof. Apply the Theorem 3 with $m_1 = 3$ and $m_2 = 2$. Even though polynomial (6) does not belong to class C, the condition $d_{1,0}$ or $d_{0,1}$ non-zero in the proof of Theorem 3 for this particular polynomial can be seen to still imply that the corresponding polynomial g is non-zero.

■

However, we can obtain a better result for this sparse polynomial (6).

Theorem 5 *With the above notations and definitions. A slight modification of Algorithm 1 and with input polynomial (6), allows to recover* (v_0, v_1) *in polynomial time in* $\log p$ *provided that* v_0 *does not lie in a certain set* $\mathcal{V}(\Delta, a) \subseteq \mathbb{F}_p$ *of cardinality,* $\#\mathcal{V}(\Delta; a) = O(\Delta^8)$.

Proof. In this case, we are looking for the vector $\mathbf{e} \in \mathbb{Z}^4$ which is of the form

$$\mathbf{e} := \left(\Delta^2\varepsilon_0, \Delta^2\varepsilon_1, \Delta\varepsilon_0^2, -\varepsilon_1^2 + \varepsilon_0^3\right),$$

and also a solution of the following linear system of congruences:

$$\begin{cases} C_1\Delta X_1 + C_2\Delta X_2 + C_3\Delta^2 X_3 + C_4\Delta^3 X_4 & \equiv & -\Delta^3 C \bmod p \\ X_1 & \equiv & 0 \bmod \Delta^2 \\ X_2 & \equiv & 0 \bmod \Delta^2 \\ X_3 & \equiv & 0 \bmod \Delta; \end{cases} \tag{7}$$

where

$$C_1 = 3w_0^2 + a,\ C_2 = -2w_1, C_3 = 3w_0, C_4 = 1, C = w_0^3 + aw_0 + b - w_1^2.$$

Let $\mathbf{f}$ be a vector with smallest Euclidean norm satisfying the above linear system of congruences (7). We may hope that $\mathbf{e}$ and $\mathbf{f}$ are the same, or at least, that we can recover the approximation errors from $\mathbf{f}$. If not, we will show that v_0 belongs to the subset $\mathcal{V}(\Delta, a) \subseteq \mathbb{F}_p$. Let us bound the "bad" possibilities for which this process does not succeed. Vector $\mathbf{d} = \mathbf{e} - \mathbf{f} = (\Delta^2 d_1, \Delta^2 d_2, \Delta d_3, d_4)$ lies in the lattice associated to (7):

$$\begin{cases} C_1\Delta X_1 + C_2\Delta X_2 + C_3\Delta^2 X_3 + C_4\Delta^3 X_4 & \equiv & 0 \bmod p \\ X_1 & \equiv & 0 \bmod \Delta^2 \\ X_2 & \equiv & 0 \bmod \Delta^2 \\ X_3 & \equiv & 0 \bmod \Delta. \end{cases} \tag{8}$$

Since $\|\mathbf{e}\| < \sqrt{7}\Delta^3$, we have that

$$|d_1| \le 2\sqrt{7}\Delta, \quad |d_2| \le 2\sqrt{7}\Delta, \quad |d_3| \le 2\sqrt{7}\Delta^2, \quad |d_4| \le 2\sqrt{7}\Delta^3. \tag{9}$$

If $d_1 \equiv d_2 \equiv 0 \bmod p$, then we can recover the root (v_0, v_1). Hence, we may assume that either d_1 or d_2 is non-zero.

Substituting $w_0 = X - \varepsilon_0, w_1 = Y - \varepsilon_1$ in the first equation of lattice (8), we obtain a bivariate polynomial:

$$g(X,Y) = (3(X-\varepsilon_0)^2 + a)d_1 - 2(Y-\varepsilon_1)d_2 + 3(X+\varepsilon_0)d_3 + d_4,$$

whose coefficients are in $\mathbb{Z}[d_1, d_2, d_3, d_4, \varepsilon_0, \varepsilon_1]$ and verifies:

$$\begin{cases} g(v_0, v_1) & \equiv & 0 \bmod p \\ f(v_0, v_1) & \equiv & 0 \bmod p. \end{cases} \tag{10}$$

Now, for every choice of $\varepsilon_0, \varepsilon_1$ and d_1, d_2, d_3, d_4 with not both d_1, d_2 zero, the number of values v_0 satisfying system (10) is at most 6 because $g(X,Y)$ is a non-zero polynomial of degree at most two.

We place any such solution v_0 into the set $\mathcal{V}(\Delta, a)$. We need to show that the cardinality of $\mathcal{V}(\Delta, a)$ is as claimed in the statement of the theorem.

We write

$$g(X,Y) = (3X^2 - 6X\varepsilon_0 + a)d_1 - 2Yd_2 + 3Xd_3 + A,$$

where $A \equiv -3\varepsilon_0 d_1 + 2\varepsilon_1 d_2 - 3\varepsilon_0 d_3 + d_4 \bmod p$.

By (9), the total number of possible choices for $d_1, d_2, d_3, \varepsilon_0$ is $O(\Delta^5)$. On the other hand, A can take $O(\Delta^3)$ distinct values. Hence there are only $O(\Delta^8)$ values of v_0 that satisfy the system of congruences (10).

Again, to finish the proof we note that the lattice is defined using given information, and that the CVP can be solved in deterministic polynomial time in $\log p$ in any fixed dimension.

■

It is well known that the elliptic curve polynomial is absolutely irreducible, then Lemma 2 applies. Obviously this result is non-trivial only for $\Delta < p^{1/8}$. Thus increasing the size of the admissible values of Δ is very interesting.

References

[1] L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[2] Simon R. Blackburn, Domingo Gómez, Jaime Gutierrez, and Igor E. Shparlinski. Predicting nonlinear pseudorandom number generators. *Math. Comp.*, 74(251):1471–1494 (electronic), 2005.

[3] Johannes Blömer and Alexander May. A tool kit for finding small roots of bivariate polynomials over the integers. In *Advances in cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Comput. Sci.*, pages 251–267. Springer, Berlin, 2005.

[4] Dan Boneh, Shai Halevi, and Nick Howgrave-Graham. The modular inversion hidden number problem. In *Advances in cryptology—ASIACRYPT 2001 (Gold Coast)*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 36–51. Springer, Berlin, 2001.

[5] Joan Boyar. Inferring sequences produced by pseudo-random number generators. *J. Assoc. Comput. Mach.*, 36(1):129–141, 1989.

[6] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[7] Don Coppersmith. Finding small solutions to small degree polynomials. In *Cryptography and lattices (Providence, RI, 2001)*, volume 2146 of *Lecture Notes in Comput. Sci.*, pages 20–31. Springer, Berlin, 2001.

[8] Jean-Sébastien Coron. Finding small roots of bivariate integer polynomial equations revisited. In *Advances in cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Comput. Sci.*, pages 492–505. Springer, Berlin, 2004.

[9] Alan M. Frieze, Johan Håstad, Ravi Kannan, Jeffrey C. Lagarias, and Adi Shamir. Reconstructing truncated integer variables satisfying linear congruences. *SIAM J. Comput.*, 17(2):262–280, 1988. Special issue on cryptography.

[10] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.

[11] Jaime Gutierrez and Álvar Ibeas. Inferring sequences produced by a linear congruential generator on elliptic curves missing high-order bits. *Des. Codes Cryptogr.*, 45(2):199–212, 2007.

[12] Nicholas Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and coding (Cirencester, 1997)*, volume 1355 of *Lecture Notes in Comput. Sci.*, pages 131–142. Springer, Berlin, 1997.

[13] Ellen Jochemsz and Alexander May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In *Advances in cryptology—ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Comput. Sci.*, pages 267–282. Springer, Berlin, 2006.

[14] Antoine Joux and Jacques Stern. Lattice reduction: a toolbox for the cryptanalyst. *J. Cryptology*, 11(3):161–185, 1998.

[15] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

[16] Hugo Krawczyk. How to predict congruential generators. *J. Algorithms*, 13(4):527–545, 1992.

[17] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[18] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems*. The Kluwer International Series in Engineering and Computer Science, 671. Kluwer Academic Publishers, Boston, MA, 2002. A cryptographic perspective.

[19] Phong Q. Nguyen and Jacques Stern. Lattice reduction in cryptology: an update. In *Algorithmic number theory (Leiden, 2000)*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 85–112. Springer, Berlin, 2000.

[20] Phong Q. Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *Cryptography and lattices (Providence, RI, 2001)*, volume 2146 of *Lecture Notes in Comput. Sci.*, pages 146–180. Springer, Berlin, 2001.

[21] Igor E. Shparlinski. *Computational and algorithmic problems in finite fields*, volume 88 of *Mathematics and its Applications (Soviet Series)*. Kluwer Academic Publishers Group, Dordrecht, 1992.

[22] Serguei A. Stepanov. *Arithmetic of algebraic curves.* Monographs in Contemporary Mathematics. Consultants Bureau, New York, 1994. Translated from the Russian by Irene Aleksanova.