Lattice-Based Solving Strategy Using Coppersmith's Techniques and Its Applications

Mengce Zheng

Zhejiang Wanli University

Crypto Seminar, Caen, France

April 30, 2025

Outline

1. Introduction

2. Theoretical Foundations
   - 2.1 Coppersmith's Idea
   - 2.2 Coppersmith-Type Theorems
   - 2.3 Lattice-Based Solving Strategy
   - 2.4 More Optimizing Techniques

3. Cryptanalysis Applications
   - 3.1 Standard RSA
   - 3.2 New RSA Variant

4. Recent Advances

1.

1.1 Initial Research

At EUROCRYPT 1996, Coppersmith proposed two lattice-based methods for finding small roots of polynomial equations, one method for polynomial equations over the integers[a] and one for modular polynomial equations[b].

Example: Solve $f(x) \equiv 0 \bmod N$

- Find all roots smaller than a certain bound X in polynomial time
- Bound X is generally of exponential size in the bit-size of modulus
- The use of LLL reduction algorithm is sufficient (no need for SVP)
- Bridges modular polynomial equations and integer solutions

[a] Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known
[b] Finding a Small Root of a Univariate Modular Equation

1.2 Cryptanalysis Applications

The prominent cryptanalysis applications of Coppersmith's techniques include small private key attacks on RSA and its variants.

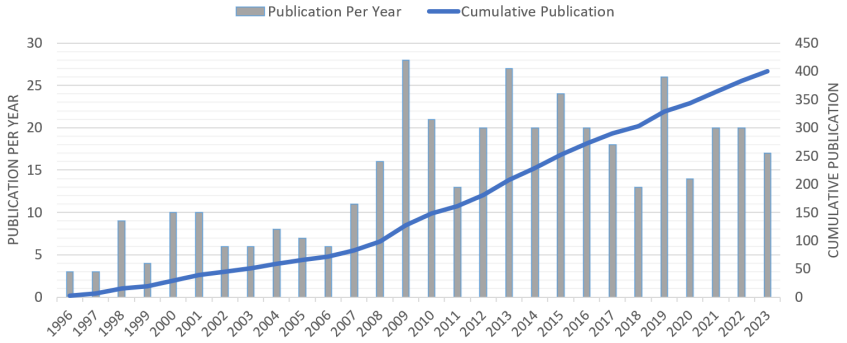
Some Known Results

- $d < N^{0.292}$ for standard RSA[a] with $d \equiv e^{-1} \bmod \varphi(N)$
- $d_p, d_q < N^{0.122}$ for CRT-RSA[b] with $d_p = d \bmod p - 1$ and $d_q = d \bmod q - 1$
- $d < N^{1-\sqrt{1/r}}$ for Multi-Prime RSA[c] with $N = p_1 \cdots p_r, r \geq 3$
- $d < N^{\frac{2-\sqrt{2}}{r+1}}$ for Takagi's (Prime Power) RSA variant[d] with $N = p^r q, r \geq 2$

[a] Cryptanalysis of RSA with Private Key d Less than $N^{0.292}$
[b] Small CRT-exponent RSA revisited
[c] General Bounds for Small Inverse Problems and Its Applications to Multi-Prime RSA
[d] Small Secret Key Attack on a Variant of RSA Due to Takagi

1.3 Lattice-Based RSA Cryptanalysis

The initial stage (1996–2006) has a gradual growth with few publications and the development stage (2007–present) reveals an increasing interest[a].



[a] Lattice-Based Cryptanalysis of RSA-Type Cryptosystems: A Bibliometric Analysis

2.

2.1.1 Coppersmith's Idea

Given a polynomial $f(x)$ of degree δ over the ring $\mathbb{Z}_N$ for some integer N of unknown factorization, one aims to find all roots of $f(x)$ in a certain interval.

Idea Description

One tries to construct a polynomial $g(x)$ of usually larger degree from $f(x)$ such that every small modular root x_0 of f, i.e. $f(x_0) \equiv 0 \bmod N$ with $|x_0| < X$, is also a root of g over $\mathbb{Z}$.

$$f(x_0) \equiv 0 \bmod N \quad \longrightarrow \quad g(x_0) = 0$$

It reduces modular univariate root finding to integer univariate root finding, for which there exist standard methods.

2.1.2 Further Analysis (1)

Fix $m \in \mathbb{Z}$ and construct g as an integer linear combination of multiples of

$$h_{i,j} = x^j N^i f^{m-i}(x).$$

Notice that every root x_0 of f satisfies

$$h_{i,j}(x_0) \equiv 0 \bmod N^m$$

Hence if g is an integer linear combination of the $h_{i,j}$'s then one shall have

$$g(x_0) \equiv 0 \bmod N^m.$$

The core issue is to calculate the corresponding coefficients of g.

2.1.3 Further Analysis (2)

Identify the polynomials $h_{i,j}(x)$ with their coefficient vectors. The integer linear combinations of these vectors form an integer lattice Λ. The small vectors in Λ correspond to possible linear combinations $g(x)$ with small coefficients.

Key Observation

If $g(x)$ has small coefficients, and is evaluated at small points x_0 with $|x_0| \leq X$, then the result must also be (somewhat) small. Assume that $g(x_0)$ is in absolute value smaller than N^m for all $|x_0| \leq X$:

$$g(x_0) \equiv 0 \bmod N^m \quad \text{and} \quad |g(x_0)| \leq |g(X)| < N^m.$$

This implies that $g(x)$ has the desired roots over the integers!

Further Analysis

(3)

If $g(x)$ has sufficiently small coefficients, $|g(x_0)| < N^m$ should automatically be fulfilled. An important lemma makes this intuition precise, which is usually contributed to Howgrave-Graham[a].

HG Lemma

Let $g(x)$ be a univariate polynomial with n monomials. Let m, X be positive integers. Suppose that

Property 1. $g(x_0) \equiv 0 \bmod N^m$, $|x_0| \leq X$, and

Property 2. $\|g(xX)\| < N^m/\sqrt{n}$.

Then $g(x_0) = 0$ holds over the integers.

[a] Finding Small Roots of Univariate Modular Equations Revisited

2.1.5 Further Analysis (4)

Suppose $g(x) = \sum_{i=0}^{n} c_i x^i$ is a given univariate polynomial and its coefficient vector is $(c_0, c_1, \ldots, c_n)$. Then *scaled polynomial* $g(xX)$ has coefficient vector $(c_0, c_1 X, \ldots, c_n X^n)$, and its Euclidean norm is denoted by $\|g(xX)\|$.

Proof Sketch

Property 2 implies

$$|g(x_0)| = \left|\sum_i c_i x_0^i\right| \leq \sum_i \left|c_i x_0^i\right| \leq \sum_i |c_i| X^i \leq \sqrt{n}\|g(xX)\| < N^m.$$

Property 1 indicates that $g(x_0)$ is a multiple of N^m, and therefore $g(x_0) = 0$.

The goal is to determine $g(x)$ with its coefficients using *lattice reduction*.

2.1.6 Lattice

A Lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$ as well as the set of all integer linear combinations of linearly independent vectors.

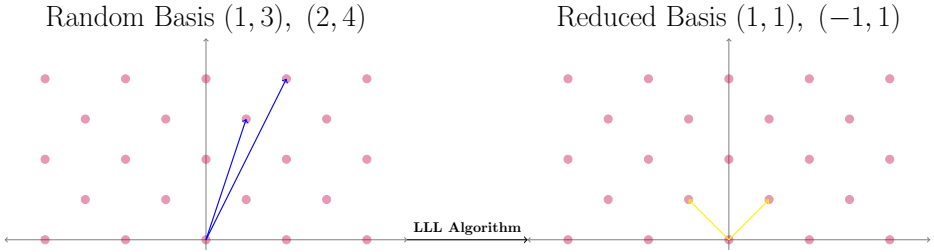
$$\Lambda = \mathbb{Z}\vec{b}_1 + \cdots + \mathbb{Z}\vec{b}_n = \left\{ \sum_{i=1}^{n} z_i \vec{b}_i : z_i \in \mathbb{Z},\ \vec{b}_i \in \mathbb{R}^n \right\}$$

Basic Concepts

- Full-rank: $\dim(\Lambda) = n$
- Basis vectors: $\vec{b}_1, \ldots, \vec{b}_n$
- Basis matrix: $B = ((\vec{b}_i)_j)_{n \times n}$
- Lattice determinant: $\det(\Lambda) = |\det(B)|$

2.1.7 Lattice Reduction (1)

Lenstra-Lenstra-Lovász reduction algorithm[a] outputs approximately shortest vectors. Lattice-based cryptanalysis using Coppersmith's techniques is widely applied to public key cryptosystems.



The LLL-algorithm runs in polynomial time regarding its input size.

[a] Factoring Polynomials With Rational Coefficients

2.1.8 Lattice Reduction (2)

Let Λ be the lattice spanned by the coefficient vectors. The LLL theorem relates the length of a shortest vector in reduced basis of Λ to $\det(\Lambda)$.

LLL Theorem

Let Λ be a lattice spanned by $\vec{b}_1, \ldots, \vec{b}_n$. The LLL-algorithm outputs a lattice vector $\vec{v} \in \Lambda$ satisfying

$$\|\vec{v}\| \leq 2^{\frac{n-1}{4}} \det(\Lambda)^{\frac{1}{n}}$$

in time $\mathcal{O}(n^6 \log^3 B_{\max})$ and $B_{\max} := \max_{i,j} |(\vec{b}_i)_j|$ is the largest basis entry.

A faster LLL-variant[a] runs in time $\mathcal{O}(n^{4+\epsilon} \log^{1+\epsilon} B_{\max})$ for any constant $\epsilon > 0$.

[a] Faster LLL-Type Reduction of Lattice Bases

2.1.9 Lattice Reduction (3)

A vector $\vec{v}$ in a Coppersmith-type lattice relates to a certain polynomial $g(x)$, for which it requires to satisfy HG Lemma. When LLL-algorithm outputs a vector $\vec{v}$ is short enough and **Property 2** have a link.

Key Link

The terms concerning n, $2^{\frac{n-1}{4}}$, $\sqrt{n}$ can be omitted for sufficiently large N:

$$\|\vec{v}\| \leq 2^{\frac{n-1}{4}} \det(\Lambda)^{\frac{1}{n}} < N^m/\sqrt{n} \quad \longrightarrow \quad \det(\Lambda) < N^{mn}.$$

This simplified inequality is the so-called *enabling condition*.

Try to construct $h_{i,j}$s' coefficient vectors with $\det(\Lambda)$ as small as possible.

2.2.1 Univariate Polynomial Case

Ready to formulate Coppersmith-type theorem for univariate polynomials and a full proof can be found in May's work[a].

Theorem (Univariate)

Let N be an integer of unknown factorization. Let $f(x)$ be a univariate monic polynomial of constant degree δ. Then one can find all solutions x_0 of the equation

$$f(x) \equiv 0 \bmod N \quad \text{with} \quad |x_0| < N^{\frac{1}{\delta}}$$

in time $\mathcal{O}(\log^{6+\epsilon} N)$ for any $\epsilon > 0$.

[a] Using LLL-Reduction for Solving RSA and Factorization Problems

2.2.2 Univariate Polynomial Case

Choose $m \approx \log N/\delta$ and define the collection of polynomials as

$$h_{i,j}(x) = x^j N^i f(x)^{m-i} \quad \text{for} \quad 0 \le i < m, 0 \le j < \delta.$$

Proof Sketch

The coefficient vectors of $h_{i,j}(xX)$ form an $n = \delta m \approx \log N$-dimensional lattice basis B with $\det(\Lambda) = \det(B) \approx N^{\delta m^2/2} X^{n^2/2}$.

$$N^{\frac{\delta m^2}{2}} X^{\frac{n^2}{2}} = N^{\frac{\delta m^2}{2}} X^{\frac{\delta^2 m^2}{2}} < N^{mn} = N^{\delta m^2} \quad \longrightarrow \quad X < N^{\frac{1}{\delta}}.$$

It works in an $n \approx \log N$-dimensional lattice with largest entries of bit-size $\log B_{\max} = \mathcal{O}(m \log N) = \mathcal{O}(\log^2 N)$. The runtime is $\mathcal{O}(\log^{6+\epsilon} N)$.

2.2.3 Extension: Larger Bound

Any small root bound X can be be extended to cX for some real number c at the expense of an additional run time factor of c.

Theorem (Univariate with Larger Bound)

Let N be an integer of unknown factorization and $c \geq 1$. Let $f(x)$ be a univariate monic polynomial of constant degree δ. Then one can find all solutions x_0 of the equation

$$f(x) \equiv 0 \bmod N \quad \text{with} \quad |x_0| < cN^{\frac{1}{\delta}}$$

in time $\mathcal{O}(c \log^{6+\epsilon} N)$ for any $\epsilon > 0$.

One can split the interval $[-cN^{\frac{1}{\delta}}, cN^{\frac{1}{\delta}}]$ in c sub-intervals of size each $2N^{\frac{1}{\delta}}$.

2.2.4 Extension: Unknown Divisor

One can extend Coppersmith's method to find roots of $f(x)$ modulo b, where $b \geq N^\beta$ is an unknown divisor of N.

Theorem (Univariate with Unknown Divisor)

Let N be an integer of unknown factorization, which has an unknown divisor $b \geq N^\beta, 0 < \beta \leq 1$. Let $c \geq 1$, and let $f(x)$ be a univariate monic polynomial of constant degree δ. Then one can find all solutions x_0 of the equation

$$f(x) \equiv 0 \bmod b \quad \text{with} \quad |x_0| < cN^{\frac{\beta^2}{\delta}}$$

in time $\mathcal{O}(c \log^{6+\epsilon} N)$ for any $\epsilon > 0$.

Use a similar strategy and work modulo b instead of N.

2.2.5 Extension: Tight Bound

The bound can be made tighter due to Coppersmith[a] and May[b].

Theorem (Univariate with Tight Bound)

Let N be an integer of unknown factorization, which has an unknown divisor $b \geq N^{\beta}, 0 < \beta \leq 1$. Let $0 < \epsilon \leq \beta/7$, and let $f(x)$ be a univariate monic polynomial of degree δ. Then one can find all solutions x_0 of the equation

$$f(x) \equiv 0 \bmod b \quad \text{with} \quad |x_0| \leq \frac{1}{2} N^{\frac{\beta^2}{\delta} - \epsilon}$$

in time $\mathcal{O}(\epsilon^{-7} \delta^5 \log^2 N)$.

[a] Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities
[b] Using LLL-Reduction for Solving RSA and Factorization Problems

2.2.6 Bivariate Polynomial Case

There exists Coppersmith-type theorem for bivariate integer polynomials and some improvements can be found in Coron's work[a].

Theorem (Bivariate)

Let $f(x, y)$ be an irreducible bivariate polynomial of maximum degree δ separately. Let X and Y be the upper bounds, and let $W = \max_{i,j} |c_{ij}|X^i Y^j$. If $XY < W^{2/(3\delta)}$, then one can find all integer pairs (x_0, y_0) such that

$$f(x_0, y_0) = 0 \quad \text{with} \quad |x_0| \leq X,\ |y_0| \leq Y$$

in time polynomial in $(\log W, 2^\delta)$.

[a] Finding Small Roots of Bivariate Integer Polynomial Equations: A Direct Approach

2.3.1 Multivariate Generalization

Coppersmith's method can be generalized to scenarios with more variables and more equations. It will be referred to as *lattice-based solving strategy*.

Involved Stages

1. Identify polynomials to be solved along with estimated bounds
2. Construct polynomials sharing a root for well-chosen parameters
3. Transform the scaled coefficient vectors into a lattice basis matrix
4. Calculate smallest reduced basis vectors from the above lattice
5. Transform output reduced vectors into several integer equations
6. Extract desired root over the integers using any simple methods

2.3.2 Detailed Description (1)

Target Problem

Given an irreducible n-variate integer polynomial $f(x_1, \ldots, x_n)$ such that

$$f(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod u,$$

and given bounds $|x_i^\star| \leq X_i$, the target is to efficiently recover the root $(x_1^\star, \ldots, x_n^\star)$. Establishing appropriate bounds X_i is crucial for deriving a solvable condition.

There exist some extended cases:

- The number of given polynomials can be extended to more than one
- The modulus can be extended to an unknown one with its multiple u

Shift Polynomials

Construct *shift polynomials* $g_k(x_1, \ldots, x_n)$ for m and $1 \leq k \leq \omega$ such that each $g_k(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod u^m$ using a positive integer m. A standard way is to define

$$g_k(x_1, \ldots, x_n) := x_1^{i_{k1}} \cdots x_n^{i_{kn}} f^{j_k} u^{m-j_k},$$

where $(i_{k1}, \ldots, i_{kn}, j_k)$ belong to an index set (relating to a monomial set)

$$\mathcal{I} = \{(i_{k1}, \ldots, i_{kn}, j_k) : i_{k1}, \ldots, i_{kn}, j_k \in \mathbb{Z},\ i_{k1}, \ldots, i_{kn} \geq 0,\ 0 \leq j_k \leq m\}.$$

They shall share a common root $(x_1^\star, \ldots, x_n^\star)$ modulo $R = u^m$.

It's suggested to use extra shifts on variables x_i's with positive integers t_i's.

Detailed Description

Coefficient Vectors

Using a proper ordering of monomials and polynomials, form a lattice Λ by representing the coefficient vectors of the scaled shift polynomials

$$g_k(x_1X_1, \ldots, x_nX_n)$$

as the rows $\vec{b}_1, \ldots, \vec{b}_\omega$ of a basis matrix B. Moreover, B can be *full-rank and triangular* under suitable arranging orders.

Generally, each shift polynomial will introduce its leading monomial as a new contribution to the diagonal of basis matrix B.

Detailed Description

Lattice Reduction

Regarding derived coefficient vectors as $\vec{b}_i$ for $i = 1, \ldots, \omega$ and construct

$$\Lambda = \left\{ \sum_{i=1}^{\omega} z_i \vec{b}_i : z_i \in \mathbb{Z} \right\}.$$

Perform LLL reduction to B to obtain several reduced vectors $\vec{v}_1, \ldots, \vec{v}_k$ with $k \geq n$. These vectors correspond to integer polynomials $h_i(x_1, \ldots, x_n)$, each satisfying $h_i(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod R$ by construction.

How to make $\det(\Lambda)$ as small as possible and how to use helpful polynomials as many as possible are two most challenging issues.

Detailed Description

Root Extraction

If $h_i(x_1, \ldots, x_n)$ for $1 \leq i \leq k$ are algebraically independent, these simultaneous integer equations can be solved through Gröbner basis approach or resultant computation. The common root $(x_1^\star, \ldots, x_n^\star)$ is finally recovered. While the assumption of algebraic independence for $n \geq 2$ is heuristic, numerical experiments generally support its validity.

There are mainly two methods for extracting the root:

- The Gröbner basis computation can be used for more variables
- The resultant computation may be used for two or three variables

Detailed Description

For the generalized lattice-based solving strategy, one needs similar LLL Lemma in a generalized form.

LLL Lemma (Generalization)

Let Λ be a given ω-dimensional lattice with input basis matrix B. The LLL-algorithm outputs a reduced basis $(\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_\omega)$ such that

$$\|\vec{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \det(\Lambda)^{\frac{1}{\omega+1-i}} \quad \text{for} \quad i = 1, 2, \ldots, \omega.$$

in time polynomial in $(\omega, B_{\max})$.

For any integer $k \leq \omega$, the LLL bound is $\|\vec{v}_1\|, \ldots, \|\vec{v}_k\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-k)}} \det(\Lambda)^{\frac{1}{\omega+1-k}}$.

Detailed Description

For the generalized lattice-based solving strategy, one needs similar HG Lemma in a generalized form.

HG Lemma (Generalization)

Let $h(x_1, x_2, \ldots, x_n)$ be an integer polynomial containing at most ω monomials. Suppose $R, X_1, X_2, \ldots, X_n$ are certain positive integers. If

$$h(x_1^\star, x_2^\star, \ldots, x_n^\star) \equiv 0 \bmod R \quad \text{and} \quad \|h(x_1X_1, x_2X_2, \ldots, x_nX_n)\| < \frac{R}{\sqrt{\omega}},$$

with $|x_i^\star| \leq X_i$ for $i = 1, \ldots, n$, then it follows that $h(x_1^\star, x_2^\star, \ldots, x_n^\star) = 0$ over the integers.

2.3.9 Enabling Condition

Solution can be achieved when the following condition holds:

$$2^{\frac{\omega(\omega-1)}{4(\omega+1-k)}} \det(\Lambda)^{\frac{1}{\omega+1-k}} < \frac{R}{\sqrt{\omega}},$$

which is rearranged as

$$\det(\Lambda) < 2^{-\frac{\omega(\omega-1)}{4}} \omega^{-\frac{\omega+1-k}{2}} R^{\omega+1-k}.$$

In practice, since $k < \omega \ll R$, it implies $\det(\Lambda) < R^{\omega-\epsilon}$ for some tiny ϵ.

Asymptotic Enabling Condition

$$\det(\Lambda) < R^{\omega} \quad \longrightarrow \quad |\det(B)| < R^{\omega}$$

2.4.1 Bivariate Integer Polynomials

Blömer and May[a] offers a general framework for extracting potential roots of bivariate integer polynomials with various *Newton polygons*.

BM Theorem

Consider an irreducible bivariate integer polynomial $f(x, y)$, where the degrees are d_x and d_y respectively. Let X, Y be the upper bounds on potential root $(x^\star, y^\star)$, let W denote $\|f(xX, yY)\|_\infty$, and let $\mathcal{S}, \mathcal{M}$ be two admissible monomial sets with $\mathcal{S} \subseteq \mathcal{M}$. Set $s = |\mathcal{S}|$, $m = |\mathcal{M}|$ and $s_x = \sum_{x^i y^j \in \mathcal{M} \setminus \mathcal{S}} i$, $s_y = \sum_{x^i y^j \in \mathcal{M} \setminus \mathcal{S}} j$ as the exponent sums for monomials $x^i y^j$. Then all potential roots satisfying $f(x^\star, y^\star) = 0$ can be extracted in time polynomial in $(m, d_x, d_y, \log W)$ if $X^{s_x} Y^{s_y} < W^s$.

[a] A Tool Kit for Finding Small Roots of Bivariate Polynomials Over the Integers

2.4.2 Multivariate Modular Polynomials

Jochemsz and May[a] describes a framework for finding small modular roots of multivariate polynomials. Let l be a leading monomial of f and fix a positive integer m. For $k \in \{0, \ldots, m+1\}$, define the set $\mathcal{M}_k$ of monomials:

$$\mathcal{M}_k := \left\{ x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \mid x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \in f^m \wedge \frac{x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}}{l^k} \in f^{m-k} \right\}.$$

Define shift polynomials $g_{i_1,\ldots,i_n} := \frac{x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}}{l^k} f^k u^{m-k}$, and condition is

$$\prod_{j=1}^{n} X_j^{s_j} < u^{s_u}, \quad \text{for} \quad s_j = \sum_{x_1^{i_1} \cdots x_n^{i_n} \in \mathcal{M}_0} i_j, \; s_u = \sum_{k=1}^{m} |\mathcal{M}_k|.$$

[a] A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants

2.4.3 Multivariate Integer Polynomials

Let degrees be d_j for x_j in f respectively. Let W denote $\|f(x_1X_1, \ldots, x_nX_n)\|_\infty$ and fix a positive integer m. Set $R = W \prod_{j=1}^{n} X_j^{d_j(m-1)}$ and $f' := c_0^{-1} f \bmod R$ to restrict constant term of 1. Define $\mathcal{S}, \mathcal{M}$ including monomials of f^{m-1} and f^m:

$$\mathcal{S} := \left\{ x_1^{i_1} \cdots x_n^{i_n} \mid x_1^{i_1} \cdots x_n^{i_n} \in f^{m-1} \right\}, \ \mathcal{M} := \left\{ x_1^{i_1} \cdots x_n^{i_n} \mid x_1^{i_1} \cdots x_n^{i_n} \in f^{m} \right\}.$$

Define shift polynomials $g := \alpha f' \prod_{j=1}^{n} X_j^{d_j(m-1)-i_j}$ for $\alpha \in \mathcal{S}$ and $g' := \alpha R$ for $\alpha \in \mathcal{M} \setminus \mathcal{S}$. The condition is

$$\prod_{j=1}^{n} X_j^{s_j} < W^{s_W}, \quad \text{for} \quad s_j = \sum_{x_1^{i_1} \cdots x_n^{i_n} \in \mathcal{M} \setminus \mathcal{S}} i_j, \ s_W = |\mathcal{S}|.$$

2.4.4 Unravelled Linearization

Herrmann and May[a] present a simpler method to construct optimized lattices that are used for finding small roots of polynomial equations. It optimizes small private key attack on RSA using $f(x, y) := 1 + x(A + y) \bmod e$.

Example

Perform a suitable linearization of the original polynomial:

$$\underbrace{1 + xy}_{u} + Ax \bmod e \quad \longrightarrow \quad \bar{f}(u, x) = u + Ax$$

with a relation $u = xy+1$. The induced relation is incorporated to substitute each occurrence of xy by the term $u - 1$.

[a] Maximizing Small Root Bounds by Linearization and Applications to Small Secret Exponent RSA

2.4.5 Root Balancing

Takayasu and Kunihiro[a] introduce the strategy for algorithm constructions that take into account the sizes of the root bounds and gain improvements.

Example

Given some positive integers $N, a_0, a_1, \ldots, a_n$, and given $(0, 1)$ real numbers $\gamma_1, \ldots, \gamma_n, \beta$, one wants to find all integers $r_1, \ldots, r_n$ such that $|r_1| \leq N^{\gamma_1}, \ldots, |r_n| \leq N^{\gamma_n}, b \geq N^{\beta}$, and $a_0 + a_1 r_1 + \cdots + a_n r_n = 0 \bmod b$. The optimized selection (using m, t) of shift polynomials is to satisfy

$$0 \leq \sum_{j=1}^{n} i_j \leq m \quad \text{and} \quad 0 \leq \sum_{j=1}^{n} \gamma_j i_j \leq \beta t.$$

[a] Better Lattice Constructions for Solving Multivariate Linear Equations Modulo Unknown Divisors

2.4.6 Minkowski Sum

Aono[a] investigates lattice constructions for more simultaneous equations and proposes a method to construct a lattice by combining lattices for solving single equations.

Example

Suppose one has lattices spanned by shift polynomials $\{g_1, \ldots, g_{\omega_1}\}$ and $\{g'_1, \ldots, g'_{\omega_2}\}$ for modular $f_1(x_1, y)$ and $f_2(x_2, y)$. The Minkowski sum based lattice construction can generate a lattice basis as a set of polynomials:

$$\sum a_k g_{k_1} g'_{k_2}$$

with certain range of (k_1, k_2) and coefficients a_k of the combination.

[a] Minkowski Sum Based Lattice Construction for Multivariate Simultaneous Coppersmith's Technique and Applications to RSA

2.4.7 Exponents Adjustment

Lu et al.[a] revisit the problem of finding small solutions to a collection of linear equations modulo an unknown divisor. They propose several generalizations by introducing multiple parameters for adjusting exponents.

Theorem

Let N be a composite integer of unknown factorization, which has a divisor b^u ($b \geq N^\beta, u \geq 1$). Let $f(x)$ be a univariate linear polynomial. Then one can find all solutions x_0 of the equation

$$f(x) \equiv 0 \bmod b^v \quad \text{with} \quad v \geq 1,\ |x_0| \leq N^{uv\beta^2-\epsilon}$$

in time $\mathcal{O}(\epsilon^{-7}v^2 \log^2 N)$ for every $\epsilon > 0$.

[a] Solving Linear Equations Modulo Unknown Divisors: Revisited

3.

3.1.1 Stereotyped Messages (1)

Consider working with RSA using a small public encryption exponent $e = 3$. Let $c = m^3 \bmod N$, where one knows an approximation m_1 of the message up to an additive error of size at most $m - m_1 < N^{\frac{5}{21}}$, i.e., there exists m_0 such that

$$m_0 = m - m_1 \quad \text{for} \quad m_0 < N^{\frac{5}{21}}.$$

Problem Description

Given: $c = m^3 \bmod N$ and some m_1 satisfying $m - m_1 < N^{\frac{5}{21}}$

Polynomial: $f(x) = (x + m_1)^3 - c \bmod N$ with root $x_0 = m - m_1 < N^{\frac{5}{21}}$

Parameters: degree $\delta = 3$

3.1.2 Stereotyped Messages (2)

Set $m = 2$, i.e. all polynomials have root x_0 modulo N^2. Define the collection of seven polynomials:

$$N^2,\ N^2x,\ N^2x^2,\ Nf(x),\ xNf(x),\ x^2Nf(x),\ f^2(x).$$

Let X denote the bound and $g_1(x), \ldots, g_7(x)$ denote the above collection. The coefficient vectors of $g_i(xX)$ for $1 \leq i \leq 7$, define the following lattice basis:

$$B = \begin{pmatrix}
N^2 & & & & & & & \\
 & N^2X & & & & & & \\
 & & N^2X^2 & & & & & \\
N(m_1^3-c) & 3Nm_1^2X & 3Nm_1X^2 & NX^3 & & & & \\
 & N(m_1^3-c)X & 3Nm_1^2X^2 & 3Nm_1X^3 & NX^4 & & & \\
 & & N(m_1^3-c)X^2 & 3Nm_1^2X^3 & 3Nm_1X^4 & NX^5 & & \\
(m_1^3-c)^2 & 6(m_1^3-c)m_1^2X & (6(m_1^3-c)m_1+9m_1^4)X^2 & (20m_1^3-2c)X^3 & 15m_1^2X^4 & 6m_1X^5 & X^6 &
\end{pmatrix}.$$

3.1.3 Stereotyped Messages (3)

Lattice basis B spans a 7-dimensional lattice Λ with $\det(\Lambda) = |\det(B)| = N^9 X^{21}$. Using the enabling condition, one obtains

$$N^9 X^{21} < N^{2 \cdot 7} \quad \longrightarrow \quad N^{\frac{5}{21}}.$$

One recovers the lower $5/21 \approx 0.238$-fraction of m in polynomial time. Theorem application yields a superior bound for the bits that one can recover at the cost of an increased running time.

Stereotyped Messages

Let $c' = m^e \bmod N$ with constant e. Assume one knows some m_1 satisfying $m - m_1 < cN^{\frac{1}{e}}$ for some $c \geq 1$. Then m can be found in time $\mathcal{O}(c \log^{6+\epsilon} N)$.

3.1.4 Factoring with Known Bits (1)

Let $N = pq$ be an RSA modulus, w.l.o.g. $p > q$ and therefore $p > N^{\frac{1}{2}}$. Assume that one knows a good approximation p_1 of p up to an additive term of size at most $N^{\frac{1}{5}}$, i.e., there exists some p_1 such that

$$p_0 = p - p_1 \quad \text{for} \quad p_0 < N^{\frac{1}{5}}.$$

Problem Description

Given: $N = pq,\ p > N^{\frac{1}{2}}$ and some p_1 satisfying $p - p_1 < N^{\frac{1}{5}}$

Polynomial: $f(x) = x + p_1 \bmod p$ with root $x_0 = p - p_1 < N^{\frac{1}{5}}$

Parameters: degree $\delta = 1$, divisor size $\beta = \frac{1}{2}$

3.1.5 Factoring with Known Bits (2)

Set $m = 2$, i.e. all polynomials have root x_0 modulo p^2. Define the collection of five polynomials:

$$N^2,\ Nf(x),\ f^2(x),\ xf^2(x),\ x^2f^2(x).$$

Let X denote the bound and $g_1(x), \ldots, g_5(x)$ denote the above collection. The coefficient vectors of $g_i(xX)$ for $1 \leq i \leq 5$, define the following lattice basis:

$$B = \begin{pmatrix} N^2 & & & & \\ Np_1 & NX & & & \\ p_1^2 & 2p_1X & X^2 & & \\ & p_1^2X & 2p_1X^2 & X^3 & \\ & & p_1^2X^2 & 2p_1X^3 & X^4 \end{pmatrix}.$$

Lattice basis B spans a 5-dimensional lattice with $\det(\Lambda) = |\det(B)| = N^3X^{10}$.

3.1.6 Factoring with Known Bits (3)

Using the enabling condition, one obtains

$$N^3 X^{10} < N^{\frac{1}{2} \cdot 2 \cdot 5} = N^5 \quad \longrightarrow \quad X < N^{\frac{1}{5}}.$$

LLL reduction runs on lattice basis B in time $\mathcal{O}(\log^3 N)$. Using above Theorem, one may increase the bound $N^{\frac{1}{5}}$ at the cost of an increased run time.

Factoring with Known Bits

Let N be composite with divisor $p > N^\beta$. Assume one is given p_1 satisfying $p - p_1 < cN^{\beta^2}$ for some $c \geq 1$. Then p can be found in time $\mathcal{O}(c \log^{6+\epsilon} N)$.

Let $N = pq$ be an RSA modulus with $p > q$, i.e. $\beta = \frac{1}{2}$. It implies that N can be factored in polynomial time given half of the bits of p, i.e. $p - p_1 < N^{\frac{1}{4}} < p^{\frac{1}{2}}$.

3.1.7 Small Private Key Attack (1)

Let $N = pq$ be an RSA modulus with $\phi(N) = (p-1)(q-1)$ and $ed \equiv 1 \bmod \phi(N)$. Assume that e is approximately of size N. Wiener[a] discovered that when $d < \frac{1}{3}N^{\frac{1}{4}}$, N can be factored via continued fractions. One has

$$ed = 1 + k(N + 1 - p - q) \quad \longrightarrow \quad ed = 1 - k(p + q - 1) + kN.$$

Problem Description

Given: $N = pq$ and e satisfying $ed \equiv 1 \bmod \phi(N)$ with $d < N^{\frac{1}{4}}$

Polynomial: $f(u, x) = u + xN \bmod e$ with root $(u_0, x_0) = (1 - k(p+q-1), k)$

Parameters: $|x_0| = |k| < d$ and $|u_0| < k(p + q - 1) < 3dN^{\frac{1}{2}}$

[a] Cryptanalysis of Short RSA Secret Exponents

3.1.8 Small Private Key Attack (2)

Set $m = 1$, i.e. all polynomials have root (u_0, x_0) modulo e. Define the collection of two polynomials:

$$g_1(u, x) = ex, \quad g_2(u, x) = f(u, x).$$

Let U, X denote the bounds and the integer linear combinations of the coefficient vectors of $g_1(uU, xX)$ and $g_2(uU, xX)$ form the following lattice basis:

$$B = \begin{pmatrix} eX & \\ NX & U \end{pmatrix}.$$

Lattice basis B spans a 2-dimensional lattice with $\det(\Lambda) = |\det(B)| = eUX$.

3.1.9 Small Private Key Attack (3)

Using the enabling condition, one obtains

$$eUX \approx X^2 N^{\frac{3}{2}} < e^2 \approx N^2 \quad \longrightarrow \quad X < N^{\frac{1}{4}}.$$

The LLL-algorithm runs on lattice basis B in time $\mathcal{O}(\log^{1+\epsilon} N)$. Coppersmith's method guarantees only that one can find a polynomial

$$h(u, x) = h_0 u + h_1 x$$

from an LLL-reduced shortest lattice vector such that $h(u_0, x_0) = 0$. It remains to recover the root (u_0, x_0). One can conclude from $h(u_0, x_0) = 0$ that

$$h_0 u_0 = -h_1 x_0$$

Since $\gcd(u_0, x_0) = 1$, it follows that $x_0 = |h_0|$ and $u_0 = -|h_1|$.

3.1.10 Small Private Key Attack (4)

Boneh and Durfee[a] further improve on Wiener's $N^{\frac{1}{4}}$-bound. They focus on the same modular equation using more shift polynomials.

Problem Description

Given: $N = pq$ and e satisfying $ed \equiv 1 \bmod \phi(N)$ with $d < N^{0.263}$

Polynomial: $f(u, x) = u + xN \bmod e$ with root $(u_0, x_0) = (1 - k(p + q - 1), k)$

Parameters: $|x_0| = |k| < d$ and $|u_0| < k(p + q - 1) < 3dN^{\frac{1}{2}}$

Set $m = 4$ and denote $d < X = N^{\delta}$, $U \approx XY = N^{\frac{1}{2}+\delta}$. All the polynomials have root (u_0, x_0) modulo e^4.

[a] Cryptanalysis of RSA with Private Key d Less than $N^{0.292}$

3.1.11 Small Private Key Attack (5)

One takes the powers of $f^i(u,x)$ for $i = 3,4$ and in addition the polynomial $yf^4(u,x)$. That is, there are now 10 polynomials:

$$e^4x^3,\ e^3x^2f,\ e^2xf^2,\ ef^3,\ e^4x^4,\ e^3x^3f,\ e^2x^2f^2,\ exf^3,\ f^4,\ yf^4.$$

Let U, X denote the bounds and the integer linear combinations of the coefficient vectors of $g_i(uU, xX)$ for $1 \le i \le 10$ form the following lattice basis B:

$$\begin{pmatrix}
e^4X^3 & & & & & & & & & \\
e^3NX^3 & e^3UX^2 & & & & & & & & \\
e^2N^2X^3 & 2e^2NUX^2 & e^2U^2X & & & & & & & \\
eN^3X^3 & 3eN^2UX^2 & 3eNU^2X & eU^3 & & & & & & \\
 & & & & e^4X^4 & & & & & \\
 & & & & e^3NX^4 & e^3UX^3 & & & & \\
 & & & & e^2N^2X^4 & 2e^2NUX^3 & e^2U^2X^2 & & & \\
 & & & & eN^3X^4 & 3eN^2UX^3 & 3eNU^2X^2 & eU^3X & & \\
 & & & & N^4X^4 & 4N^3UX^3 & 10N^2U^2X^2 & 4NU^3X & U^4 & \\
N^4X^3 & 4N^3UX^2 & 10N^2U^2X & 4NU^3 & & -N^4UX^3 & -4N^3U^2X^2 & -10N^2U^3X & -4NU^3 & U^4Y
\end{pmatrix}.$$

3.1.12 Small Private Key Attack (6)

Lattice basis B spans a 10-dimensional lattice Λ and its determinant is $\det(\Lambda) = |\det(B)| = e^{20}U^{20}X^{16}Y$. Using the enabling condition, one obtains

$$e^{20}U^{20}X^{16}Y < e^{4 \cdot 10} \quad \longrightarrow \quad N^{\delta} < N^{\frac{19}{72}} \approx N^{0.263}.$$

One finds polynomials with root (x_0, y_0) over the integers in polynomial time via lattice reduction. The bound is finally improved to $N^{0.292}$

Small Private Key Attack

Let N be an RSA modulus with $ed \equiv 1 \bmod \phi(N)$, $e \approx N$ and $d < N^{0.292}$. Then N can be factored in time $\mathcal{O}(\log^{6+\epsilon} N)$ for any $\epsilon > 0$.

One computes $y_0 = p + q - 1$ from two polynomials via resultant computation.

3.2.1 Cubic Pell RSA

Basic Information

- A new RSA variant introduced by Murru and Saettone[a]
- Based on cubic Pell equation $x^3 + ry^3 + r^2z^3 - 3rxyz = 1$
- Use a novel group with a non-standard product $\odot$ on tuple (m_1, m_2)

[a] A Novel RSA-Like Cryptosystem Based on a Generalization of the Rédei Rational Functions

Key Information

- Public/private keys are $(N,\ e,\ r)/(d,\ p,\ q)$ with $N = pq$
- Use $ed \equiv 1 \bmod \phi(N)$ for $\phi(N) = (p^2 + p + 1)(q^2 + q + 1)$
- Key equation is $ed - k(p^2 + p + 1)(q^2 + q + 1) = 1$ for an unknown k

3.2.2 Small Private Key Attack (1)

Zheng et al.[a] investigate the potential small private key attack and show it is vulnerable for $d < N^{2-\sqrt{2}}$.

Bivariate Modular Equation

Its key equation is $ed = k((p+q)^2+(N+1)(p+q)+N^2-N+1)+1$. Consider the following bivariate polynomial $f(x, y)$:

$$x(y^2 + ay + b) + 1 \equiv 0 \bmod e$$

where $a = N + 1$ and $b = N^2 - N + 1$. Thus, $(x^\star, y^\star) = (k, p + q)$ is the modular root. We set the upper bounds to be $X = 2N^{\alpha+\delta-2}$, $Y = 3N^{\frac{1}{2}}$.

[a] Cryptanalysis of the RSA Variant Based on Cubic Pell Equation

3.2.3 Small Private Key Attack (2)

Let $h(y) := y^2 + ay + b = y^2 + \bar{h}(y)$ for $\bar{h}(y) := ay + b$. The original polynomial $f(x, y)$ is rewritten to

$$f(x, y) = xh(y) + 1 = x(y^2 + \bar{h}(y)) + 1 = (xy^2 + 1) + x\bar{h}(y).$$

Variable Relation

Letting $z := xy^2 + 1$, we have $\bar{f}(x, y, z) := z + x\bar{h}(y)$. The shift polynomials $g_{[i,j,k]}(x, y, z)$ are defined as

$$g_{[i,j,k]}(x, y, z) := x^i y^j \bar{f}^k(x, y, z) e^{m-k} = x^i y^j (z + x\bar{h}(y))^k e^{m-k}$$

for a fixed positive integer m and non-negative integers i, j, k.

Small Private Key Attack

Shift Polynomials

We denote the set of shift polynomials by $\mathcal{F} := \mathcal{G} \cup \mathcal{H}$ for

$$\mathcal{G} := \{g_{[i,j,k]}(x, y, z) : (i, j, k) \in \mathcal{I}_{\mathcal{G}}\},$$
$$\mathcal{H} := \{g_{[i,j,k]}(x, y, z) : (i, j, k) \in \mathcal{I}_{\mathcal{H}}\},$$

where the corresponding index set $\mathcal{I} := \mathcal{I}_{\mathcal{G}} \cup \mathcal{I}_{\mathcal{H}}$ is defined by

$$\mathcal{I}_{\mathcal{G}} := \{(i, j, k) : i = 0, \ldots, m;\ j = 0, 1;\ k = 0, \ldots, m - i\},$$
$$\mathcal{I}_{\mathcal{H}} := \{(i, j, k) : i = 0;\ j = 2, \ldots, \lfloor \tau k \rfloor + 1;\ k = 0, \ldots, m\},$$

for a parameter $0 \leq \tau \leq 1$ to be optimized later.

3.2.5 Small Private Key Attack (4)

Coefficient vectors of scaled shift polynomials $g_{[i,j,k]}(xX, yY, zZ)$ generate the basis matrix.

Coefficient Vectors

The polynomial order $\prec_{\mathrm{p}}$ is defined as $g_{[i,j,k]} \prec_{\mathrm{p}} g_{[i',j',k']}$ if

- $i + k < i' + k'$; or
- $i + k = i' + k'$ and $i < i'$; or
- $i = i'$, $k = k'$ and $j < j'$.

The monomial order $\prec_{\mathrm{m}}$ is defined as $x^i y^j z^k \prec_{\mathrm{m}} x^{i'} y^{j'} z^{k'}$ in a similar way.

Notice that we shall substitute each occurrence of xy^2 by the term $z - 1$.

Small Private Key Attack

Integer Lattice

Regarding derived coefficient vectors as $\vec{b}_i$ for $i = 1, \ldots, \omega$ and construct

$$\Lambda = \left\{ \sum_{i=1}^{\omega} z_i \vec{b}_i : z_i \in \mathbb{Z} \right\}.$$

The lattice dimension ω is calculated as

$$\omega = \sum_{(i,j,k) \in \mathcal{I}} 1 = \frac{2+\tau}{2} m^2 + o(m^2).$$

Small Private Key Attack

A toy example of the lattice basis matrix for $m = 2$ and $\tau = 1$ is shown:

	1	y	x	xy	z	yz	y^2z	x^2	x^2y	xz	xyz	z^2	yz^2	y^2z^2	y^3z^2
$g_{[0,0,0]}$	e^2														
$g_{[0,1,0]}$		e^2Y													
$g_{[1,0,0]}$			e^2X												
$g_{[1,1,0]}$				e^2XY											
$g_{[0,0,1]}$			–	–	eZ										
$g_{[0,1,1]}$	–			–	–	eYZ									
$g_{[0,2,1]}$	–	–			–	–	eY^2Z								
$g_{[2,0,0]}$								e^2X^2							
$g_{[2,1,0]}$									e^2X^2Y						
$g_{[1,0,1]}$								–	–	eXZ					
$g_{[1,1,1]}$			–						–	–	$eXYZ$				
$g_{[0,0,2]}$			–					–	–	–	–	Z^2			
$g_{[0,1,2]}$			–	–	–				–	–	–	–	YZ^2		
$g_{[0,2,2]}$	–		–	–	–	–				–	–	–	–	Y^2Z^2	
$g_{[0,3,2]}$	–	–		–	–	–	–				–	–	–	–	Y^3Z^2

Small Private Key Attack

Lattice Determinant

A lower triangular basis matrix only requires multiplication of the diagonal terms for computing the determinant:

$$\det(\Lambda) = e^{n_e} X^{n_X} Y^{n_Y} Z^{n_Z}.$$

We obtain the respective exponents

$$n_e = \frac{1}{6}(\tau + 4)m^3 + o(m^3),\ n_X = \frac{1}{3}m^3 + o(m^3),$$
$$n_Y = \frac{\tau^2}{6}m^3 + o(m^3),\ n_Z = \frac{1}{3}(\tau + 1)m^3 + o(m^3).$$

Small Private Key Attack

Attack Bound

The solving condition $\det(\Lambda) < R^{\omega}$ with $R = e^m$ yields

$$N^{\alpha n_e + (\alpha+\delta-2)n_X + \frac{1}{2}n_Y + (\alpha+\delta-1)n_Z} < N^{\alpha m \omega}.$$

Simplify the exponents over N and obtain $\tau^2 + (4\delta - 4)\tau + 4\alpha + 8\delta - 12 < 0$. By setting $\tau = 2 - 2\delta$, it further leads to $\delta < 2 - \sqrt{\alpha}$. Note that we must ensure $0 \leq \tau = 2 - 2\delta \leq 1$ and finally have

$$\delta < 2 - \sqrt{\alpha} \text{ for } 1 \leq \alpha < \frac{9}{4}, \quad \text{or} \quad \delta < \frac{5}{4} - \frac{\alpha}{3} \text{ for } \frac{9}{4} \leq \alpha < \frac{15}{4}.$$

3.2.10 Generalized Cryptanalysis (1)

Kang and Zheng[a] present generalized cryptanalysis of cubic Pell RSA with its generalized key equation $eu - (p^2+p+1)(q^2+q+1)v = w$.

Trivariate Modular Equation

We have $v(p+q)^2 + (N+1)(p+q)v + (N^2-N+1)v + w \equiv 0 \bmod e$. Consider the following trivariate polynomial:

$$f(x,y,z) = xy^2 + axy + bx + z,$$

where $a = N+1$ and $b = N^2 - N + 1$. Thus, $(x^\star, y^\star, z^\star) = (v, p+q, w)$ is the modular root. Set upper bounds to be $X = 2N^{\beta+\delta-2}$, $Y = 3N^{\frac{1}{2}}$, $Z = N^\gamma$.

[a] Generalized Cryptanalysis of Cubic Pell RSA

Generalized Cryptanalysis

Monomial Sets

Let m be a positive integer and t be a non-negative integer to be optimized later. For $0 \leq k \leq m$, we define the following monomial set

$$\mathcal{M}_k = \bigcup_{0 \leq j \leq 2+t} \left\{ x^{i_1} y^{i_2+j} z^{i_3} : \ x^{i_1} y^{i_2} z^{i_3} \text{ is a monomial of } f(x,y,z)^m \right.$$
$$\left. \text{and } \frac{x^{i_1} y^{i_2} z^{i_3}}{(xy^2)^k} \text{ is a monomial of } f(x,y,z)^{m-k} \right\}.$$

We can obtain an accurate description of i_1, i_2, i_3 for each $x^{i_1} y^{i_2} z^{i_3} \in \mathcal{M}_k$:

$$i_1 = k, \ldots, m, \ i_2 = 2k, \ldots, 2i_1 + 2 + t, \ i_3 = m - i_1.$$

Generalized Cryptanalysis

(3)

Shift Polynomials

We define the following shift polynomials for $x^{i_1}y^{i_2}z^{i_3} \in \mathcal{M}_k \setminus \mathcal{M}_{k+1}$:

$$g_{k,i_1,i_2,i_3}(x,y,z) = \frac{x^{i_1}y^{i_2}z^{i_3}}{(xy^2)^k} f(x,y,z)^k e^{m-k}.$$

Furthermore, shift polynomials can be divided into two polynomial sets:

$$\mathcal{G}_{k,i_1,i_2,i_3}(x,y,z) = x^{i_1-k}y^{i_2-2k}z^{i_3} f(x,y,z)^k e^{m-k},$$
$$k = 0,\dots m,\ i_1 = k,\dots,m,\ i_2 = 2k, 2k+1,\ i_3 = m - i_1,$$
$$\mathcal{H}_{k,i_1,i_2,i_3}(x,y,z) = y^{i_2-2k}z^{i_3} f(x,y,z)^k e^{m-k},$$
$$k = 0,\dots m,\ i_1 = k,\ i_2 = 2k+2,\dots,2i_1+2+t,\ i_3 = m - i_1.$$

3.2.13 Generalized Cryptanalysis (4)

Coefficient vectors of $\mathcal{G}_{k,i_1,i_2,i_3}(xX, yY, zZ)$ and $\mathcal{H}_{k,i_1,i_2,i_3}(xX, yY, zZ)$, with X, Y, and Z denoting the upper bounds generate the basis matrix.

Coefficient Vectors

Concerning row order, precedence is given to any $\mathcal{G}_{k,i_1,i_2,i_3}$ over any $\mathcal{H}_{k,i_1,i_2,i_3}$. The polynomial order $\prec_{\mathrm{p}}$ is established as $(k, i_1, i_2, i_3) \prec_{\mathrm{p}} (k', i'_1, i'_2, i'_3)$ if

- $k < k'$; or
- $k = k'$ and $i_1 < i'_1$; or
- $k = k',\ i_1 = i'_1$ and $i_2 < i'_2$; or
- $k = k',\ i_1 = i'_1,\ i_2 = i'_2$ and $i_3 < i'_3$.

The monomial order $\prec_{\mathrm{m}}$ is defined as $x^{i_1}y^{i_2}z^{i_3} \prec_{\mathrm{m}} x^{i'_1}y^{i'_2}z^{i'_3}$ in a similar way.

Generalized Cryptanalysis

(5)

Integer Lattice

Regarding derived coefficient vectors as $\vec{b}_i$ for $i = 1, \ldots, \omega$ and construct

$$\Lambda = \left\{ \sum_{i=1}^{\omega} z_i \vec{b}_i : z_i \in \mathbb{Z} \right\}.$$

The lattice dimension ω is calculated as

$$\omega = \sum_{k=0}^{m} \sum_{i_1=k}^{m} \sum_{i_2=2k}^{2k+1} \sum_{i_3=m-i_1}^{m-i_1} 1 + \sum_{k=0}^{m} \sum_{i_1=k}^{k} \sum_{i_2=2k+2}^{2i_1+2+t} \sum_{i_3=m-i_1}^{m-i_1} 1 = (m+1)(m+t+3).$$

Generalized Cryptanalysis

A toy example of the lattice basis matrix for $m = 2$ and $t = 0$ is shown:

	z^2	yz^2	xz	xyz	x^2	x^2y	xy^2z	xy^3z	x^2y^2	x^2y^3	x^2y^4	x^2y^5	y^2z^2	xy^4z	x^2y^6
$G_{[0,0,0,2]}$	Z^2e^2														
$G_{[0,0,1,2]}$		YZ^2e^2													
$G_{[0,1,0,1]}$			XZe^2												
$G_{[0,1,1,1]}$				$XYZe^2$											
$G_{[0,2,0,0]}$					X^2e^2										
$G_{[0,2,1,0]}$						X^2Ye^2									
$G_{[1,1,2,1]}$	–		–	–			XY^2Ze								
$G_{[1,1,3,1]}$		–		–			–	XY^3Ze							
$G_{[1,2,2,0]}$			–		–	–			X^2Y^2e						
$G_{[1,2,3,0]}$				–		–			–	X^2Y^3e					
$G_{[2,2,4,0]}$	–		–	–	–	–	–		–	–	X^2Y^4				
$G_{[2,2,5,0]}$		–		–		–	–	–	–	–	–	X^2Y^5			
$H_{[0,0,2,2]}$													$Y^2Z^2e^2$		
$H_{[1,1,4,1]}$							–	–					–	XY^4Ze	
$H_{[2,2,6,0]}$							–	–	–	–	–	–	–	–	X^2Y^6

Generalized Cryptanalysis

(7)

Lattice Determinant

A lower triangular basis matrix only requires multiplication of the diagonal terms for computing the determinant:

$$\det(\Lambda) = e^{n_e} X^{n_X} Y^{n_Y} Z^{n_Z}.$$

Letting $t = \tau m$ with a real $\tau \geq 0$, we obtain $\omega = (\tau + 1)m^2 + o(m^2)$ and

$$n_e = \frac{1}{6}(3\tau + 4)m^3 + o(m^3),\ n_X = \frac{1}{6}(3\tau + 4)m^3 + o(m^3),$$
$$n_Y = \frac{1}{6}\left(3\tau^2 + 6\tau + 4\right)m^3 + o(m^3),\ n_Z = \frac{1}{6}(3\tau + 2)m^3 + o(m^3).$$

Generalized Cryptanalysis

Attack Bound

The solving condition $\det(\Lambda) < R^{\omega}$ with $R = e^m$ yields

$$N^{\beta n_e + (\beta+\delta-2)n_X + \frac{1}{2}n_Y + \gamma n_Z} < N^{\beta m \omega}.$$

Simplify the exponents over N and obtain

$$\delta < \frac{-3\tau^2 + (6-6\gamma)\tau + 12 - 4\beta - 4\gamma}{6\tau + 8}.$$

By setting $\tau = (2\sqrt{1+3\beta-3\gamma} - 4)/3$, it further leads to

$$\delta < \frac{7}{3} - \gamma - \frac{2}{3}\sqrt{1+3\beta-3\gamma}.$$

4.

4.1 Limitations of Existing Strategies

Ryan[a] examines the problem of finding small solutions to systems of modular multivariate polynomials.

Limitations

- The main difficulty is shift polynomial selection stage
- Rely on heuristics: one crafts monomial sets and shifts by hand
- The manual design is time-consuming: new problem-specific strategies often take years of work to improve bounds
- Existing multivariate Coppersmith bounds lag behind potential: lattice dimensions can be huge and asymptotic analysis is hard

[a] Solving Multivariate Coppersmith Problems with Known Moduli

4.2 Main Contributions

This research develops *automated algorithms* to analyze shift polynomials, thus reducing manual work.

Advantages

- **Optimal shifts via Gröbner bases:** Provably find the best shift polynomials from the polynomial ideal
- **Graph-based monomial selection:** Heuristically identify low-rank sublattices by analyzing coefficients with a directed graph
- **Symbolic precomputation:** Precompute shift-polynomial data and use polytope analysis to automatically determine asymptotic bounds

He conducts extensive evaluation on 14 standard Coppersmith problems, new methods match or improve prior bounds and produce smaller lattices.

4.3 Shift Polynomial Selection

Involved Steps

- Form ideal J generated by input polynomials (with known modulus)
- Compute a D-Gröbner basis G of J under a weight monomial order
- For each monomial α in the finite superset of $\mathcal{M}$, find $g \in G$ with $\mathrm{LM}(g) \mid \alpha$, choose g with minimal leading coefficient, and set shift to be $g \cdot (\alpha / \mathrm{LM}(g))$
- This produces a set $\mathcal{S}$ of shift polynomials (support in monomial set $\mathcal{M}$) that is suitable and optimal

The lattice from these shifts is the optimal dual lattice for Coppersmith's bound. In practice, this lattice has the shortest possible basis vectors.

4.4 Graph-Based Monomial Selection

Many shift polynomials are sparse, implying some lattice columns are zero.

Involved Steps

- Yield a dense sublattice: dropping these columns lowers rank with little cost.
- Build a directed graph $\mathcal{G}$: vertices are monomials in $\mathcal{M}$, draw an edge $(\alpha_1 \to \alpha_2)$ if $f \in \mathcal{S}$ has $\mathrm{LM}(f) = \alpha_1$ and a nonzero coefficient on α_2
- Find a maximum-weight closure in $\mathcal{G}$ and select a subset $\mathcal{M}_{\mathsf{sub}} \subset \mathcal{M}$ (and $\mathcal{S}_{\mathsf{sub}}$) with no outgoing edges, minimizing $\det(\Lambda_{\mathcal{S}_{\mathsf{sub}}})^{1/|\mathcal{S}_{\mathsf{sub}}|}$

Exploiting sparsity helps bridge the gap between theoretical LLL bounds and observed performance.

4.5 Symbolic Precomputation

It presents a strategy based on symbolic precomputation of a set of shift polynomials and extend this precomputed set to higher multiplicities.

Involved Steps

- Symbolically represent the problem: treat unknown constants as variables and compute a generic ideal $\widehat{J}$ and its shifts
- Compute the monomial set via a convex hull (a polytope) and use Ehrhart polynomials to count monomials and bound lattice dimension
- Precompute shift polynomials for $\widehat{J}$, then specialize to each instance by interpolation and avoid repeated expensive computations
- Substitute optimizing parameters and consider asymptotic behavior

It fully automates asymptotic Coppersmith analysis for multivariate systems.

4.6 Evaluation on Existing Problems

This research test on 14 known multivariate Coppersmith instances.

Experimental Results

- **Recovery Bounds:** The provable (Gröbner) strategy always matched or improved the best-known root size bounds
- **Lattice Dimension:** The graph-based method often achieved the same bounds with significantly smaller lattices
- **Runtime:** The precomputation strategy gave similar bounds and lattice sizes with much faster solve times
- **Asymptotic:** Automated polytope analysis reproduced the strongest known asymptotic exponents.

4.7 Future Work

Recent advances introduces a suite of methods to automate multivariate Coppersmith attacks and achieve state-of-the-art results.

Remaining Challenges

- The heuristics still require parameter choices. E.g, selecting monomial vertices in the polytope must be done manually
- The graph-based methods are ineffective against integer Coppersmith problems and it requires further analysis
- How to calculate the determinant of Coppersmith lattices when that are not full-rank
- It does not capture the multi-step approaches, which construct multiple Coppersmith lattices to gain partial information of roots

Mengce Zheng

Crypto Seminar, Caen, France, April 30, 2025

mengce.zheng@gmail.com