CR10: Cryptanalysis Autumn 2018

Lecture 6 — october 24, 2018

Lecturer G. Hanrot, E. Kirshanova *Scribe: Caroline Brosse*

1 Introduction

1.1 What is Coppersmith method about?

Coppersmith method is about finding "small" solutions to modular polynomial equations. It dates back to the mid '90s ([Cop96]).

Univariate version: Given an integer N, a bound $0 \leq X < N$, a polynomial $P \in \mathbb{Z}/N\mathbb{Z}[x]$, find all integers x such that $|x| \leq X$ and $P(x) = 0 \ mod \ N$ in time $poly(\log N, \deg P)$.

More generally, same question : given $X_1, \ldots, X_k$, $P \in \mathbb{Z}/N\mathbb{Z}[x_1, \ldots, x_k]$, find all k-tuples of integers such that for all i, $|x_i| < X_i$ and $P(x_1, \ldots, x_k) = 0 \ mod \ N$.

We cannot hope that it works for all X (hence the fact that this is about small solutions).

Example $P(x) = x^d$, $N = p^d$, p prime.

$$P(x) = 0 \ mod \ N \Leftrightarrow x = kp, \ k \in \mathbb{Z}$$

$\#\{x | P(x) = 0 \ mod \ N, \ x < X\} = \dfrac{2X}{p}(1 + o(1)), \ X \to +\infty$. In this case, the limit is $X = N^{\frac{1}{\deg P}}$. We can then see the importance of the bound in the problem.

If we know the factorization of N, we should rather use it to reduce to solving $P(x) = 0 \ mod \ p^k$, then to solving $P(x) = 0 \ mod \ p$ and recombine using Hensel lemma + CRT.

Remark If we have an algorithm which is able to return a random (uniform) solution of $x^2 = 1 \ mod \ N$ on input N, then we have a factoring algorithm. Indeed, $(x+1)(x-1) = 0 \ mod \ N$. If N is not p^k or $2p^k$, the equation $x^2 = 1 \ mod \ N$ has at least 4 solutions, so with probability greater than $\frac{1}{2}$ we get a solution $z \neq \pm 1$.

Then we claim that $N_1 := \gcd(x-1, N)$, $N_2 := \gcd(x+1, N)$ are two nontrivial factors of N and iterate the process on N_1 and N_2 if a full factorization is sought.

1.2 A few words about the strategy of Coppersmith method

In the following, let $P \in \mathbb{Z}/N\mathbb{Z}[x]$, $\deg P = d$, $|x| \leq X$.

Fact 1. *If P has coefficients with representations in $[0, N-1]$ which are $\leq B$ in absolute value, then*

$$|P(x)| \leq B \sum_{i=0}^{d} x_i \leq (d+1)BX^d \ .$$

Hence, if $(d+1)BX^d < N$, from $P(x) = 0 \ mod \ N$ we can deduce that $P(x) = 0$ over the integers.

Fact 2. *If $P(xX)$ has coefficients less than B in absolute value, then $|P(x)| \leq (d+1)B$ ($P(x) = \sum_{i=1}^{d} p_i x^i$, so the assumption implies that $|p_i X^i| \leq B$ for all i).*

Key idea: Build an auxiliary polynomial Q from P with the size property needed by Fact 2 and such that

$$P(x) \equiv 0 \ (mod \ N) \ \Rightarrow \ Q(x) \equiv 0 \ (mod \ N) \ ,$$

e.g. take $Q_1(x) = xP(x)$, or $Q_k(x) = x^k P(x)$, or any linear combination of those.

Key idea': Build an auxiliary polynomial Q from P with the size property needed by Fact 2 and such that

$$P(x) \equiv 0 \ (mod \ N) \ \Rightarrow \ Q(x) \equiv 0 \ (mod \ N^\ell) \text{ for some integer } \ell;$$

e.g. any linear combination of $x^i P(x)^{\ell-k} N^k$.

Theorem 3 (LLL - [LLL82])**.** *There exists a deterministic polynomial time algorithm which on input $(b_1, \ldots, b_d) \in \mathbb{Z}^d$ $\mathbb{R}$-linearly independent, returns a nonzero $\mathbb{Z}$-linear combination of $b_1, \ldots, b_d$ (call it x) such that*

$$\|x\|_2 = 2^{\mathcal{O}(d)} . (\det L(b_1, \ldots, b_d))^{\frac{1}{d}} \ .$$

2 Coppersmith's theorem(s)

Theorem 4. *There exists a deterministic polynomial time algorithm, which on input N an integer and $P \in \mathbb{Z}/N\mathbb{Z}[x]$ a polynomial of degree d, returns either a nontrivial factor of N, or all integers x such that $|x| \leq N^{\frac{1}{d}}$ and $P(x) = 0 \ mod \ N$.*

Theorem 5. *Given $\beta \in]0,1[$, $\varepsilon > 0$, there exists a deterministic polynomial time algorithm in $(\log N, \deg P, \frac{1}{\varepsilon})$ which on input N, $P \in \mathbb{Z}/N\mathbb{Z}[x]$, $\deg P \leq d$, returns either a nontrivial factor of N, or all x such that $|x| \leq N^{\frac{\beta^2}{d} - \varepsilon}$ and $\gcd(P(x), N) \geq N^\beta$.*

The difference between Theorems 4 and 5 lies on the dependency on a parameter β.

Theorem 6. *We can remove ε from Theorem 5.*

Proof of Theorem 4. Step 1: reduce to monic P.

Let $P = \sum_{i=0}^{d} p_i x^i$. If $\gcd(p_d, N) = 1$, then consider $\tilde{P} = p_d^{-1} P$, which is monic. Otherwise, we have two divisors of N :

$N_1 = \gcd(N, p_d)$
$N_2 = \dfrac{N}{N_1}$.

Fix ℓ to be chosen later, and define $Q_{i,j}(x) = x^i P(x)^j N^{\ell-j}$, $0 \le i < d$ (to avoid linear dependence).

$$\deg Q_{i,j} = i + jd, \quad 0 \le j \le \ell,$$

so under our assumptions on i, all $Q_{i,j}$ have distinct degrees and must be linearly independent.

$$\begin{aligned} P(x) = 0 \ mod\ N \Rightarrow \forall i,j, \quad & Q_{i,j}(x) = 0 \ mod\ N^\ell \\ \Rightarrow \forall (\alpha_{i,j})_j \in \mathbb{Z}^{(\ell+1)d}, \quad & \sum_{j=0}^{\ell} \alpha_{i,j} Q_{i,j}(x) = 0 \end{aligned}$$

Finally, the linear independence of the $Q_{i,j}$ implies the linear independence of $Q_{i,j}(xX)$, for X an integer bound to be chosen later. Let

$$\begin{array}{rccc} \varphi: & \mathbb{Z}/N\mathbb{Z}[x]_{\le(\ell+1)d-1} & \longrightarrow & \mathbb{Z}^{(\ell+1)d} \\ & \sum_{i=0}^{(\ell+1)d-1} u_i x^i & \longmapsto & \begin{pmatrix} u_0 \\ \vdots \\ u_{(\ell+1)d-1} \end{pmatrix} \end{array}.$$

Then, if $b_{i,j} = \varphi(Q_{i,j}(xX))$, the matrix of the $b_{i,j}$ (in a suitable order) is lower-triangular.

$$\begin{pmatrix} N^\ell & 0 & 0 & & \\ 0 & XN^\ell & 0 & & \\ & & XN^{2\ell} & 0 & \\ \vdots & & & & \\ 0 & & & & N^{\ell-1}X^d \end{pmatrix}$$

On the diagonal of the matrix, we have $N^\ell, \ldots, X^{d-1}N^\ell$,$X^d N\ell - 1, \ldots, X^{2d-1}N\ell - 1, \ldots,$ $X^{\ell d}, \ldots, X^{(\ell+1)d-1}$.

Let $\delta := (\ell+1)d$ the dimension of the lattice. The determinant of the corresponding lattice is $X^{\frac{\delta(\delta-1)}{2}} N^{d\frac{\ell(\ell+1)}{2}}$. LLL guarantees the existence of (and computes) a vector v (hence a polynomial) with $\|v\|_2 \le 2^{\mathcal{O}(\delta)} X^{\frac{\delta-1}{2}} N^{\frac{\ell}{2}}$ (up to a constant).

Summary: From LLL we obtain the coefficient vector of a polynomial $R(xX)$ such that:

- the coefficients of $R(xX)$ are $\le 2^{\mathcal{O}(\delta)} X^{\frac{\delta-1}{2}} N^{\frac{\ell}{2}}$
- $\forall x,\ P(x) = 0 \ mod\ N \ \Rightarrow\ R(x) = 0 \ mod\ N^\ell$.

For $|x| \le X$, $|R(x)| \le (\delta+1)2^{\mathcal{O}(\delta)} X^{\frac{\delta-1}{2}} N^{\frac{\ell}{2}}$.

Say $2^{\mathcal{O}(\delta)} \le 2^{c\delta}$ for some c. Then $\underbrace{(\delta+1)2^{c\delta} X^{\frac{\delta-1}{2}} N^{\frac{\ell}{2}} < N^\ell}_{(\#)}$, so if (#) holds, $R(x) = 0$ for all $|x| < X$ solution of $P(x) = 0 \ mod\ N$.

$$(\#) \Rightarrow X^{\frac{\delta-1}{2}} < N^{\frac{\ell}{2}} \frac{2^{-c\delta}}{\delta+1}$$

$$X < N^{\frac{\ell}{\delta-1}} \quad \underbrace{\frac{2^{-\frac{-2c\delta}{\delta-1}}}{(\delta+1)^{\frac{2}{\delta-1}}}}_{\mathcal{O}(1) \text{ as } \delta\to+\infty \text{ so } \leq c}$$

$$X < cN^{\frac{\ell}{\delta-1}}$$

With $\delta = (\ell+1)d$, choose $\ell = \frac{(\log N)+1}{d} - 1$. Then $\delta - 1 = \log N$.

The bound is $X < cN^{\frac{1}{d}} \left(N^{\frac{-1}{d \log N} - \frac{1}{\log N}} \right)$, but $N^{\frac{-1}{d \log N} - \frac{1}{\log N}} = \mathcal{O}(1)$, $N \to +\infty$ so $< c'$.

$$X < cc' N^{\frac{1}{d}}$$

Now, from an algorithm working for $X < c'' N^{\frac{1}{d}}$, it is easy to deduce an algorithm working for $X < N^{\frac{1}{d}}$ by applying the first one to the $\mathcal{O}\left(\frac{1}{c''}\right)$ polynomials

$$\underbrace{P(x)}_{\text{roots in } [-c''N^{\frac{1}{d}}, c''N^{\frac{1}{d}}]} \quad , \quad \underbrace{P(x \pm c'' N^{\frac{1}{d}})}_{\text{roots in } [-2c''N^{\frac{1}{d}}, -c''N^{\frac{1}{d}}] \cup [c''N^{\frac{1}{d}}, 2c''N^{\frac{1}{d}}]} \quad , \; P(x \pm 2c'' N^{\frac{1}{d}}), \; \ldots$$

□

Remark: The proof can be adapted to obtain always (removing the case *returns a nontrivial factor of N*) the solutions x such that $|x| \leq \left(\frac{N}{\gcd(N, p_d)} \right)^{\frac{1}{d}}$.

The dependence on p_d cannot be avoided. Indeed, for $N = 2^k$, $P(x) = 2^j x$ (so $d = 1$), when $N \to +\infty$, for all $\varepsilon > 0$, there are $\mathcal{O}(N^\varepsilon)$ solutions x such that $|x| \leq N^{k-j+\varepsilon}$.

Proof of Theorem 5. Keep the $Q_{i,j}$ for $i < d$, $j \leq \ell$. Add $x^k P^\ell$, $k = 0, \ldots, t$, t to be chosen. The determinant is $x^{\frac{\delta(\delta-1)}{2}} N^{\frac{d\ell(\ell-1)}{2}}$, $\delta = d\ell + t + 1$.

Claim: $\gcd(P(x), N) > N^\beta \Rightarrow \gcd(R(x), N^\ell > N^{\ell\beta}$, so if $|R(x)| < N^{\ell\beta}$, then $R(x) = 0$.

Now, LLL gives R such that $\|\varphi\|_2 \leq 2^{\mathcal{O}(\delta)} X^{\frac{\delta-1}{2}} N^{\frac{d\ell(\ell+1)}{2\delta}}$. So a sufficient condition to get $R(x) = 0$ (over $\mathbb{Z}$) for any x such that $|x| < X$ and $\gcd(P(x), N) > N^\beta$ is $(\delta+1)2^{c\delta} X^{\frac{\delta-1}{2}} N^{\frac{d\ell(\ell+1)}{2\delta}} < N^{\ell\beta}$, or $X < cN^{\frac{2\ell\beta}{\delta-1} - \frac{d\ell(\ell+1)}{\delta(\delta-1)}}$.

For any ε, we want to choose t_ε so that, for ℓ large enough,

$$\frac{2\ell\beta}{\delta-1} - \frac{d\ell(\ell+1)}{\delta(\delta-1)} \geq \frac{\beta^2}{d} - \varepsilon$$

$$\Leftrightarrow 2\ell d\beta(d\ell + t + 1) - d^2\ell(\ell+1) - (\beta^2 - d\varepsilon)(d\ell + t)(d\ell + t + 1) \geq 0.$$

This is a degree 2 polynomial in t with negative leading coefficient. We can find such a t as soon as $\left|\frac{disc(\theta)}{l.c.(\theta)}\right| \geq 1$, which gives the condition $4\varepsilon d^3\ell^2 + \mathcal{O}(\ell, poly(d), poly(\frac{1}{\varepsilon})) \geq 1$. This holds for some $\ell = poly(d, \frac{1}{\varepsilon})$.

□

Proof of Theorem 6. With $\varepsilon = \frac{1}{\log_2(N)}$, Theorem 5 implies that we know how to solve $\gcd(P(x), N) > N^\beta$ for $|x| \leq \frac{1}{2}N^{\frac{\beta^2}{d}}$. Conclude as in the end of the proof of Theorem 4: use the algorithm for P, $P\left(x \pm \frac{N^{\frac{\beta^2}{d}}}{2}\right)$.

□

3 Applications

3.0 (Plain) RSA

secret key: pair of distinct primes (p, q), $d \in (\mathbb{Z}/(p-1)(q-1)\mathbb{Z})^*$
public key: $N = pq$, $e = d^{-1}\ mod\ (p-1)(q-1)$
encryption: $x \mapsto x^e\ mod\ N$
decryption: $x \mapsto x^d\ mod\ N$

The security of the message rests on the hardness of inverting $x \mapsto x^e\ mod\ N$. The security of the key rests on the hardnness of factoring N.

It is commonly suggested to take $e = 3$ ($p, q = 2\ mod\ 3$).

3.1 Factoring with high bits known

Say we know some $p = p_0 + x$, $x < N^t$ for some $t < \frac{1}{2}$. Take $P(x) = x + p_0$. By theorem 6, $\gcd(P(x), N) > N^\beta$ can be solved for $x < N^{\beta^2}$ ($d = 1$). With $\beta \simeq \frac{1}{2}$, the solutions of $\gcd(P(x), N) > N^{\frac{1}{2}}$ contain the x such that $p_n + x = p$.

Hence we are able to find those x as soon as $t \leq \frac{1}{4}$.

3.2 Low encryption exponent attacks

Let C be a ciphertext for which we know most of the clear text. Then $m = m_0 + x$, m_0 known, x unknown but small, and we solve $(m_0 + x)^e = C\ mod\ N$. It can be done as soon as $|x| \leq N^{\frac{1}{e}}$ by Theorem 4.

References

[Cop96] Don Coppersmith. Finding a small root of a univariate modular equation. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 155–165. Springer, 1996.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.