The art of cryptography, summer 2013

Lattices and cryptography

Prof. Dr. Joachim von zur Gathen

ALGORITHM 1. Small polynomial with high-order roots.

Input: A monic linear polynomial $f \in \mathbb{Z}[t]$, positive integers N, c, and k, and real μ with $0 < \mu \leq 1$.
Output: $g \in \mathbb{Z}[t]$.

1. $n \longleftarrow \lceil k/\mu \rceil$.
2. $h_i \longleftarrow \begin{cases} N^{k-i} f^i & \text{for } 0 \leq i \leq k, \\ t^{i-k} f^k & \text{for } k < i < n. \end{cases}$
3. Form the $n \times n$ matrix A whose rows are the coefficient vectors of $h_0(ct), \ldots, h_{n-1}(ct)$.
4. Apply the basis reduction algorithm to the rows of A, with output $B = UA$ and $U \in \mathbb{Z}^{n \times n}$ unimodular. Let $(u_0, \ldots, u_{n-1}) \in \mathbb{Z}^n$ be the top row of U.
5. Return $g = \sum_{0 \leq i < n} u_i h_i$.

LEMMA 2. *The output of Algorithm 1 satisfies* $\deg g < n$ *and*

$$\|g(ct)\| \leq 2^{(n-1)/4} N^{k(k+1)/2\ell} c^{(n-1)/2}.$$

For any $r \in \mathbb{Z}$ *and a divisor* m *of* N, *we have*

$$f(r) = 0 \text{ in } \mathbb{Z}_m \Longrightarrow g(r) = 0 \text{ in } \mathbb{Z}_{m^k}.$$

The algorithm uses time polynomial in $\log(N \cdot \|f\|)$ *and* k/μ.

Theorem 3. *Let f, N, c, k, and μ be an input for Algorithm 1, g the output, and*

$$\delta \geq \frac{1}{2} \log_N((\frac{k}{\mu} + 1)2^{k/2\mu}), \tag{4}$$

$$c \leq N^{\mu^2 - \mu(\mu+2\delta)/k}, \tag{5}$$

and $m \geq N^\mu$ be a divisor of N. Then the set R of all integer roots of g has at most $\lceil k/\mu \rceil$ elements and contains all $r \in \mathbb{Z}$ with $f(r) = 0$ in $\mathbb{Z}_m$ and $|r| \leq c$. R can be computed in polynomial time.

Suppose that an attacker discovers in RSA the most significant half of the bits of p. At first sight, it is not clear how to use this. We will now show how to factor N completely and efficiently with this partial information.

THEOREM 6. *Let $p < q$ be primes, $N = pq \geq 2653$, and $v \in \mathbb{Z}$ with $|q - v| \leq N^{1/4}/2$. Given N and v, one can compute q in polynomial time.*

EXAMPLE 7. We take $N = 2183$ $(= 37 \cdot 59)$. Then $N^{1/4}/2 < 3.42$ and $\alpha = \log_2 N \approx 11.09$. Thus $k = \lceil \alpha \rceil = 12$, $\mu = 1/2$,

$$\begin{aligned} \delta &= \frac{1}{2} + \frac{\log_2(4n+6)}{2n} \approx 0.75, \\ c^* &= N^{1/4-(1/4+\delta)/k} \approx 3.59, \\ c &= 3. \end{aligned}$$

We are also given $v = 56$ with the guarantee that $|q - v| \leq N^{1/4}/2 < 3.42$. Then $|q - v| \leq c$. We call Algorithm 1 with inputs $f = t + 56$, N, c, k, and $\mu = 1/2$. In step Algorithm 1 step 1, we have $n = \lceil k/\mu \rceil = 24$, and form a 24×24 matrix A. The output is a polynomial $g \in \mathbb{Z}[x]$ of degree 23 which factors over $\mathbb{Z}$ as $g = (x - 3) \cdot (x - 56) \cdot h$, where $h \in \mathbb{Z}[x]$ is irreducible. Thus $Z = \{3\}$ and $\gcd(56 + 3, N) = 59$. We have found the factor $q = 59$ of N, and then $p = N/59 = 37$.

Example 8 (cont.).

k	v	c	roots
12	56	3	$(t-3)^3$
	55	4	$(t-4)^3$
	54	5	$(t-5)^2$
	53	6	$(t-6)(t+53)^2$
	52	7	$(t-7)(t+52)^{10}$
	51	8	$(t-8)(t+51)^{10}$
11	56	3	$(t-3)^3$
	52	7	$(t-7)(t+52)^9$
5	56	3	$t-3$
	53	6	$(t-3)(t+53)^3$
4	56	3	$(t-3)(t+56)$
	55	4	$t-4$
	54	5	$t-5$
	53	6	$(t-6)(t+53)$

Table : Some experiments for factoring 2183.

COROLLARY 9. *Let $p < q$ be primes and $N = pq \geq 2653$. If N is hard to factor, then it is hard to find an approximation to q to within $N^{1/4}/2$.*