CS641

Modern Cryptology

Lecture 15

Post Quantum Cryptography

- With full-scale quantum computers in near future increasingly likely, in 2017, NIST started a worldwide contest to identify a new public-key encryption algorithm that is resistant against quantum computers.
- It received 59 submissions, of which 17 were shortlisted after one round of evaluation.
- After another round, completed in July 2020, the number got reduced to four.
- The final result is expected this year.
- Out of four remaining candidates, three are based on hardness of finding shortest vector in integer lattices.
- We describe one of these three: the NTRU cryptosystem.

NTRU Cryptosystem

- Proposed by Hoffstein, Pipher, and Silverman in 1996.
- Since then it has undergone significant analysis and revision.
- Today, there exist multiple versions of the algorithm.
- Even in the shortlist, there are multiple variants: we describe the simplest one.

Key Generation

- Fix numbers n and q where n is prime, q a power of two with $n/3 \leq q/8 - 2 \leq 2n/3$, and 3 is a generator of F_n^*.
- Define ring $R = \mathbb{Z}_q[x]/(x^n - 1)$ where:
  - $\mathbb{Z}_q$ is the ring of integers modulo q,
  - $\mathbb{Z}_q[x]$ is the ring of polynomials in x with coefficients from $\mathbb{Z}_q$, and
  - $\mathbb{Z}_q[x]/(x^n - 1)$ is the ring of remainder polynomials obtained by dividing polynomials in $\mathbb{Z}_q[x]$ by $x^n - 1$.
- Let $T \subset R$ be the set of polynomials with coefficients in $\{-1, 0, 1\}$.
- Let $T(d) \subset T$ be the set of polynomials with exactly $d/2$ coefficients $+1$ and $d/2$ coefficients -1.

Key Generation

- Pick a random $f \in T$ such that f is invertible in the ring R as well as in the ring $\mathbb{Z}_3[x]/(x^n - 1)$.
- Pick a random $g \in T(q/8 - 2)$ such that g is invertible in the ring R.
- Let $f_q = f^{-1} \pmod{q, x^n - 1}$ and $f_p = f^{-1} \pmod{3, x^n - 1}$.
- Compute $h = 3f_q g \pmod{q, x^n - 1}$.
- Public key: (n, q, h)
- Private key: (f, f_p)

Encryption

- Let $m \in T(q/8 - 2)$ be a plaintext block.
- Pick a random $r \in T$.
- Compute $c = rh + m \pmod{q, x^n - 1}$.
- Output c.

Decryption

- Let c be the ciphertext polynomial.
- Compute $e = cf \pmod{q, x^n - 1}$, and write e such that its coefficients are in the range $[-q/2 + 1, q/2]$.
- Compute $m' = ef_p \pmod{p}$.
- Output m'.

Correctness

- When $c = rh + m \pmod{q, x^n - 1}$, we have, in the ring R:

$$\begin{aligned} e &= cf \\ &= rhf + mf \\ &= 3rgf_q f + mf \\ &= 3rg + mf \end{aligned}$$

- Since $r \in T$ and $g \in T(q/8 - 2)$, each coefficient of rg is in the range $[-(q/8 - 2), q/8 - 2]$.
- In fact, since r is chosen randomly at the time of encryption, each coefficients will be in the range $[-q/24, q/24]$ with high probability.
- Since rg is a polynomial of degree at most $2n - 2$, coefficients of $rg \pmod{x^n - 1}$ will be in the range $[-(q/12), q/12]$ with high probability.

Correctness

- Same analysis holds for the product mf: all the coefficients of $mf \pmod{x^n - 1}$ are in the range $[-(q/4 - 4), q/4 - 4]$.
- Therefore, coefficients of $3rg + mf \pmod{x^n - 1}$ are in the range $[-(q/2 - 4), q/2 - 4]$ with high probability.
- This implies that

$$e = 3rg + mf \pmod{q, x^n - 1} = 3rg + mf \pmod{x^n - 1}.$$

- Therefore,

$$\begin{aligned} m' &= ef_p \pmod{3, x^n - 1} \\ &= 3rgf_p + mff_p \pmod{3, x^n - 1} \\ &= m \pmod{3, x^n - 1} \\ &= m \end{aligned}$$

with high probability.

Correctness

- We can check correctness of encryption by computing $r' = (c - m')h^{-1} \pmod{q, x^n - 1}$.
- If $r' \notin T$, then decryption has failed.
- In that case, a re-encryption is required.

Security: Computing Private Key

- Private key (f, f_p) can be computed if f is computed.
- To compute f from (n, q, h), we need to use the relationship $h = gf_q \pmod{q, x^n - 1}$, which is equivalent to:

$$hf = 3g \pmod{q, x^n - 1}.$$

- Taking coefficients of f and g as unknowns, above gives a system of n homogeneous linear equations in $2n$ unknowns.
- This has at least q^n solutions, one of which is correct one.
- To identify the correct solution, we use the fact that both f and g are special polynomials with coefficients in $\{-1, 0, 1\}$.
- As done earlier, we construct a lattice from the linear equations where pair (f, g) viewed as a vector is a short one.

Security: Computing Private Key

- As (f, g) viewed as a vector is in $\{-1, 0, 1\}^{2n}$, we define a lattice in $\mathbb{Z}^{2n}$.
- Let

$$
\begin{aligned}
f(x) &= \sum_{i=0}^{n-1} \alpha_i x^i \\
g(x) &= \sum_{i=0}^{n-1} \beta_i x^i \\
h(x) &= \sum_{i=0}^{n-1} r_i x^i.
\end{aligned}
$$

- Then coefficient of x^i of polynomial $hf \pmod{q, x^n - 1}$ equals

$$\sum_{j=0}^{i} \alpha_j r_{i-j} + \sum_{j=i+1}^{n-1} \alpha_j r_{n+i-j} + \gamma_i q,$$

for some $\gamma_i \in \mathbb{Z}$.

Security: Computing Private Key

- Define a lattice $\mathcal{L}$ with basis being row vectors of following matrix:

$$B = \begin{bmatrix} 1 & 0 & \cdots & 0 & r_0 & r_1 & \cdots & r_{n-1} \\ 0 & 1 & \cdots & 0 & r_{n-1} & r_0 & \cdots & r_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & r_1 & r_2 & \cdots & r_0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q \end{bmatrix}$$

Security: Computing Private Key

- We have:

  $$[\alpha_0\ \alpha_1 \cdots \alpha_{n-1}\ \gamma_0\ \gamma_1 \cdots \gamma_{n-1}]B = [\alpha_0\ \alpha_1 \cdots \alpha_{n-1}\ 3\beta_0\ 3\beta_1 \cdots 3\beta_{n-1}].$$

- Therefore, lattice $\mathcal{L}$ has a vector of length at most $\sqrt{n + 9(q/8 - 2)} \leq \sqrt{3q/2 - 24}$.
- The shortest vector of $\mathcal{L}$ has length at most $\sqrt{2n}\sqrt{q} = \sqrt{2nq}$.
- Therefore, the vector $[f\ 3g] = [\alpha_0\ \alpha_1 \cdots \alpha_{n-1}\ 3\beta_0\ 3\beta_1 \cdots 3\beta_{n-1}]$ is likely to be the shortest vector in the lattice $\mathcal{L}$.
- However, this is a hard-to-solve problem.

Security: Computing Plaintext

- This problem is: given (n, q, h, c) with $c = rh + m \pmod{q, x^n - 1}$, find m.
- Considering the same lattice $\mathcal{L}$ as above, we have that vector $(r, c - m)$ is in $\mathcal{L}$.
- Note that since $(r, c - m) = (0, c) + (r, -m)$, vector $(r, c - m)$ is close to the vector $(0, c)$.
- Further, $(r, c - m)$ is likely to be the closest vector in lattice $\mathcal{L}$ to the vector $(0, c)$.
- This is Closest Vector Problem for lattices which is known to be even harder to solve that shortest vector problem.
- Therefore, finding plaintext also appears to be hard to solve.

Parameter Values

- Fastest known algorithm for finding shortest vector in a lattice in $\mathbb{R}^D$ takes time $2^{\Omega(D)}$.
- This holds true even on quantum computers.
- This allows comparatively small value of D for security.
- Suggested values for parameters in NTRU submission are:

$$n = 509, q = 2048$$
$$n = 677, q = 2048$$
$$n = 821, q = 4096$$