Lattice-based cryptography V

Attacks on NTRU

Tanja Lange
(with some slides from Christine van Vredendaal)

Eindhoven University of Technology

SAC – Post-quantum cryptography

Evaluation-at-1 attack

Ciphertext equals $c = rh + m$ and g and r have t coefficients equal to 1 and t coefficients equal to -1.

Evaluation-at-1 attack

Ciphertext equals $c = rh + m$ and g and r have t coefficients equal to 1 and t coefficients equal to -1.

This means $r(1) = 0$ and $h(1) = g(1)/f(1) = 0$.

Evaluation-at-1 attack

Ciphertext equals $c = rh + m$ and g and r have t coefficients equal to 1 and t coefficients equal to -1.

This means $r(1) = 0$ and $h(1) = g(1)/f(1) = 0$.

This implies

$$c(1) = r(1)h(1) + m(1) = m(1)$$

which gives information about m, in particular if $|m(1)|$ is large.

NTRU rejects extreme messages – this is dealt with by randomizing m via a padding (not mentioned so far).

For other choices of r and h, e.g. choosing r with one fewer -1 than 1, one knows $r(1)$ and h is public, so evaluation at 1 leaks $m(1)$.

Evaluation-at-1 attack

Ciphertext equals $c = rh + m$ and g and r have t coefficients equal to 1 and t coefficients equal to -1.

This means $r(1) = 0$ and $h(1) = g(1)/f(1) = 0$.

This implies

$$c(1) = r(1)h(1) + m(1) = m(1)$$

which gives information about m, in particular if $|m(1)|$ is large.

NTRU rejects extreme messages – this is dealt with by randomizing m via a padding (not mentioned so far).

For other choices of r and h, e.g. choosing r with one fewer -1 than 1, one knows $r(1)$ and h is public, so evaluation at 1 leaks $m(1)$.

Could also replace $x^n - 1$ by $\Phi_n = (x^n - 1)/(x - 1)$ to avoid attack.

Mathematical attacks

- Meet-in-the-middle attack;
- Lattice-basis reduction (e.g. LLL, BKZ);
- Hybrid attack, combining both.

Crypto attacks:

- Chosen-ciphertext attacks;
- Decryption-failure attacks;
- Complicated padding systems.

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for f in two parts

$$h = (f_1 + f_2)^{-1} 3g$$
$$f_1 \cdot h = 3g - f_2 \cdot h.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for f in two parts

$$h = (f_1 + f_2)^{-1} 3g$$
$$f_1 \cdot h = 3g - f_2 \cdot h.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$
- Solution: look for collisions in $c(f_1 \cdot h)$ and $c(-f_2 \cdot h)$ with

$$c(a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}) = (\mathbf{1}(a_0 > 0), \ldots, \mathbf{1}(a_{n-1} > 0))$$

  using that g is small and thus $+g$ often does not change the sign.
- If $c(f_1 \cdot h) = c(-f_2 \cdot h)$ check whether $h(f_1 + f_2)$ has correct coefficients.
- Basically runs in square root of size of search space.

Odlyzko's meet-in-the-middle attack on NTRU

- Idea: split the possibilities for f in two parts

$$h = (f_1 + f_2)^{-1} 3g$$
$$f_1 \cdot h = 3g - f_2 \cdot h.$$

- If there was no g: collision search in $f_1 \cdot h$ and $-f_2 \cdot h$
- Solution: look for collisions in $c(f_1 \cdot h)$ and $c(-f_2 \cdot h)$ with

$$c(a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}) = (\mathbf{1}(a_0 > 0), \ldots, \mathbf{1}(a_{n-1} > 0))$$

  using that g is small and thus $+g$ often does not change the sign.
- If $c(f_1 \cdot h) = c(-f_2 \cdot h)$ check whether $h(f_1 + f_2)$ has correct coefficients.
- Basically runs in square root of size of search space.
- General running time / memory mitm (Christine van Vredendaal)

$$L = \sqrt{|S|}/\sqrt{s}.$$

Attackable rotations

In NTRU, $x^i f$ is simply a rotation of f, so it has the same coefficients, just at different positions.

This means, $x^i f$ also gives a solution in the mitm attack:
$hx^i f = x^i g$ has same sparsity etc., increasing the number of targets.

Decryption using $x^i f$ works the same as with f for NTRU,
so each target is valid.

Security against Odlyzko's meet-in-the-middle attack

- Number of choices for f is

$$\binom{n}{t}\binom{n-t}{t-1}$$

  because f has $2t-1$ non-zero coefficients.
- Number of rotations is n.
- Running time / memory against NTRU

$$L = \frac{\sqrt{\binom{n}{t}\binom{n-t}{t-1}}}{\sqrt{n}}.$$

Security against Odlyzko's meet-in-the-middle attack

- Number of choices for f is
$$\binom{n}{t}\binom{n-t}{t-1}$$
because f has $2t-1$ non-zero coefficients.
- Number of rotations is n.
- Running time / memory against NTRU
$$L = \frac{\sqrt{\binom{n}{t}\binom{n-t}{t-1}}}{\sqrt{n}}.$$
- Memory requirement can be reduced.

Security against lattice sieving

- Recall $h = 3g/f$ in $\mathbf{R}/q$.
- This implies that for $k \in \mathbf{R}$: $f \cdot h/3 + k \cdot q = g$.
- NTRU lattice

$$\begin{pmatrix} k & f \end{pmatrix} \begin{pmatrix} qI_n & 0 \\ H & I_n \end{pmatrix} = \begin{pmatrix} g & f \end{pmatrix}.$$

Security against lattice sieving

- Recall $h = 3g/f$ in $\mathbf{R}/q$.
- This implies that for $k \in \mathbf{R}$: $f \cdot h/3 + k \cdot q = g$.
- NTRU lattice

$$\begin{pmatrix} k & f \end{pmatrix} \begin{pmatrix} qI_n & 0 \\ H & I_n \end{pmatrix} = \begin{pmatrix} g & f \end{pmatrix}.$$

- Key pair (g, f) is a short vector in this lattice.
- Asymptotically sieving works in $2^{0.292 \cdot 2n + o(n)}$ using $2^{0.208 \cdot 2n + o(n)}$ memory.
- Crossover point between sieving and enumeration is still unclear.
- Memory is more an issue than time.
- Can use sieving and enumeration as subroutines in BKZ.

Hybrid attack

Howgrave-Graham combines lattice basis reduction and meet-in-the-middle attack.

- Idea: reduce submatrix of the NTRU lattice, then perform mitm on the rest.

Hybrid attack

Howgrave-Graham combines lattice basis reduction and meet-in-the-middle attack.

- Idea: reduce submatrix of the NTRU lattice, then perform mitm on the rest.
- Use BKZ on submatrix B to get B':

$$C \cdot \begin{pmatrix} qI_n & 0 \\ H & I_n \end{pmatrix} = \left(\begin{array}{ccc} qI_w & 0 & 0 \\ \hline * & B' & 0 \\ \hline * & * & I_{w'} \end{array}\right).$$

- Guess options for last w' coordinates of f, using collision search (as before).
- If the Hermite factor of B' is small enough, then a rounding algorithm can detect collision of "halfguesses".