Lattices in Cryptography #2

The NTRU encryption scheme

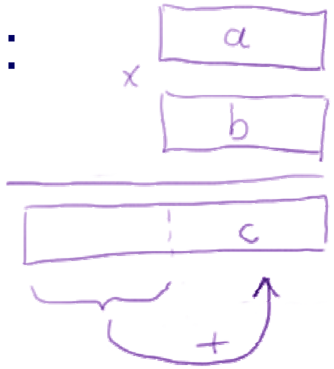
[Hoffstein, Pipher, Silverman 1998]

- Fast
- Has small keys
- Different
- Secure?

NTRU: preliminaries

- Fix n=167, q=128, p=3 (important: $q \gg p$, $\gcd(p,q)=1$).
- We work in the ring $\mathbb{Z}[x]/(x^n - 1)$
  whose elements are polynomials of degree<n
  (which we will often write as n-vectors).
- Additional is component-wise.
- Multiplication of polynomials is done modulo x^n-1:

$$c = a * b \;\leftrightarrow\; c_i = \sum_{j=0}^{n-1} a_j b_{i-j\,(\mathrm{mod}\ n)}$$



  i.e., normal polynomial multiplication followed by "folding" the coefficients vector modulo n and summing its entries.
- Sometimes will work modulo p or modulo q – this means taking all coefficient values modulo p or q.

NTRU: the keys

- Private key:
  - f – a polynomial with coefficients in {-1,0,1}
    (61 1's, 60 -1's and 46 0's)
  - g – a polynomial with coefficients in {-1,0,1}
    (20 1's, 20 -1's and 127 0's)
  - f_p^{-1}, f_q^{-1} – polynomials fulfilling

$$f_p^{-1} * f \equiv 1 \pmod{p}$$

$$f_q^{-1} * f \equiv 1 \pmod{q}$$

  f,g chosen randomly subject to the above.
- Public key: $h \leftarrow f_q^{-1} * g \pmod{q}$

NTRU: encryption

- Encryption:
  - Message is given as a polynomial m with with coefficients in {-1,0,1}.
  - Choose r, a random polynomial with 18 1's, 18 -1's and 131 0's.
  - Ciphertext: $c \leftarrow p \cdot r * h + m \pmod q$

NTRU: decryption

reminder

$$f_p^{-1} * f \equiv 1 \pmod p$$
$$f_q^{-1} * f \equiv 1 \pmod q$$
$$h \equiv f_p^{-1} * g \pmod q$$
$$c \equiv p \cdot r * h + m \pmod q$$

$$\left.\begin{aligned} a &\leftarrow c * f \\ &\equiv f * (p \cdot r * h + m) \\ &\equiv p \cdot r * g * f_q^{-1} * f + m * f \\ &\equiv p \cdot r * g + m * f \end{aligned}\right\} \pmod q$$

The polynomials r,g,m,f all have tiny coefficients, and p is small.
So if we take the coefficients of a in {-$q/2+1$,...,$q/2$} it is likely that

$$a = p \cdot r * g + m * f \quad (\text{over } \mathbb{Z})$$

$$\Rightarrow a \equiv p \cdot r * g + m * f \pmod p$$

and then:

$$\left.\begin{aligned} a * f_p^{-1} &\equiv (p \cdot *r * g + m * f) * f_p^{-1} \\ &\equiv m * f * f_p^{-1} \\ &\equiv m \end{aligned}\right\} \pmod p$$

Lattice attack on NTRU

[Coppersmith, Shamir 1997]

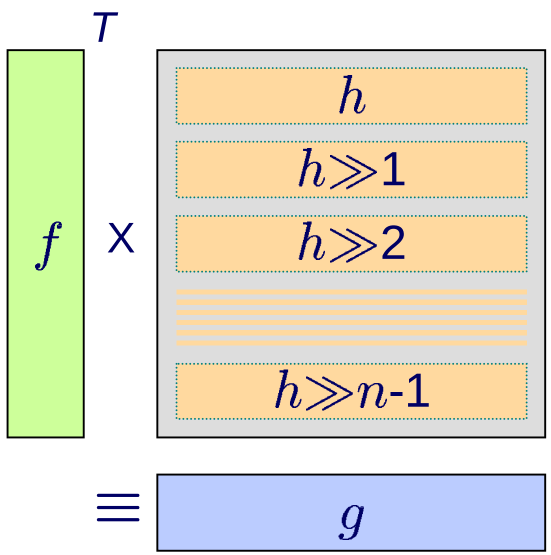
$$h \equiv f_q^{-1} * g \pmod q$$

$$\Rightarrow f * h \equiv g \pmod q$$

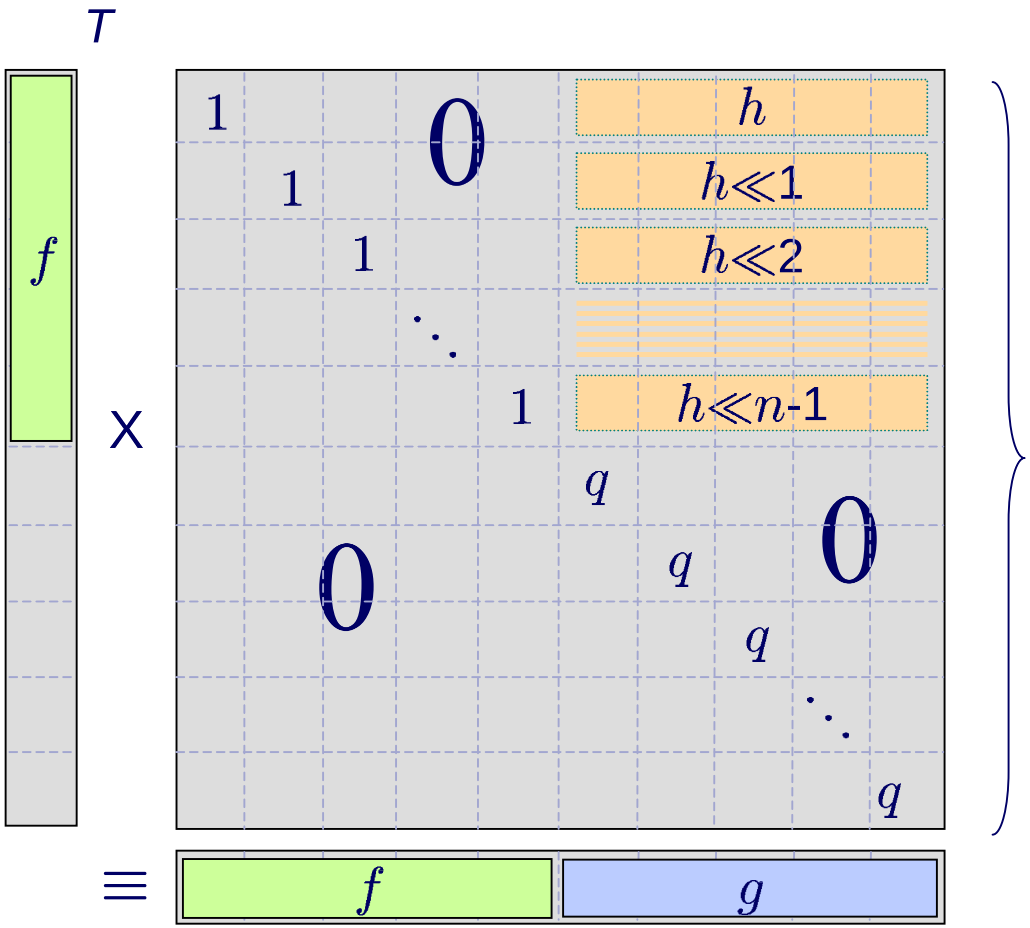
where h is known and f,g have tiny coefficients.

$$\left.\begin{aligned} f * h &= g \\ \sum_{j=0}^{n-1} f_j h_{i-j \pmod n} &= g_i \\ \sum_{j=0}^{n-1} f_j(\vec{h} \gg j) &= \vec{g} \end{aligned}\right\} \pmod q$$

cyclic shift



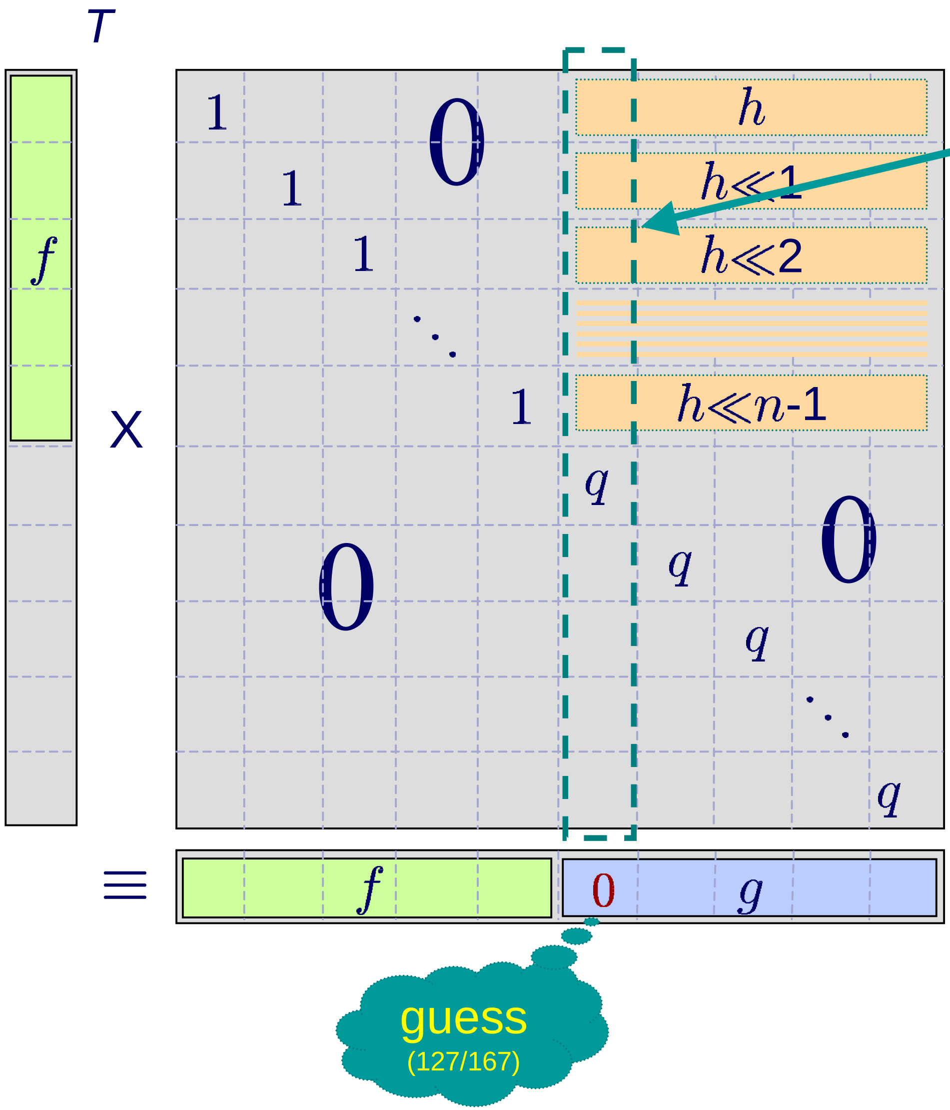
Lattice attack on NTRU (cont.)



row vectors generating the lattice

Improvement: “zero-run lattice”

[May, 1999]



Multiply values in this column by a large number

⇓

All short vectors in the lattice will have “0” at the guessed coordinate.

⇓

Larger gap

⇓

LLL performs better

Forcing several coordinates to zero: tradeoff between LLL performance and probability of good guess.

Improvement: “zero-forced lattice”

[Silverman, 1999]

$$f * h = g \pmod{q}$$

$$\sum_{j=0}^{n-1} f_j h_{i-j \pmod{n}} = g_i \pmod{q}$$

Suppose we guess that the $g_0, \ldots, g_{r-1}=0$. We get r linear equations:

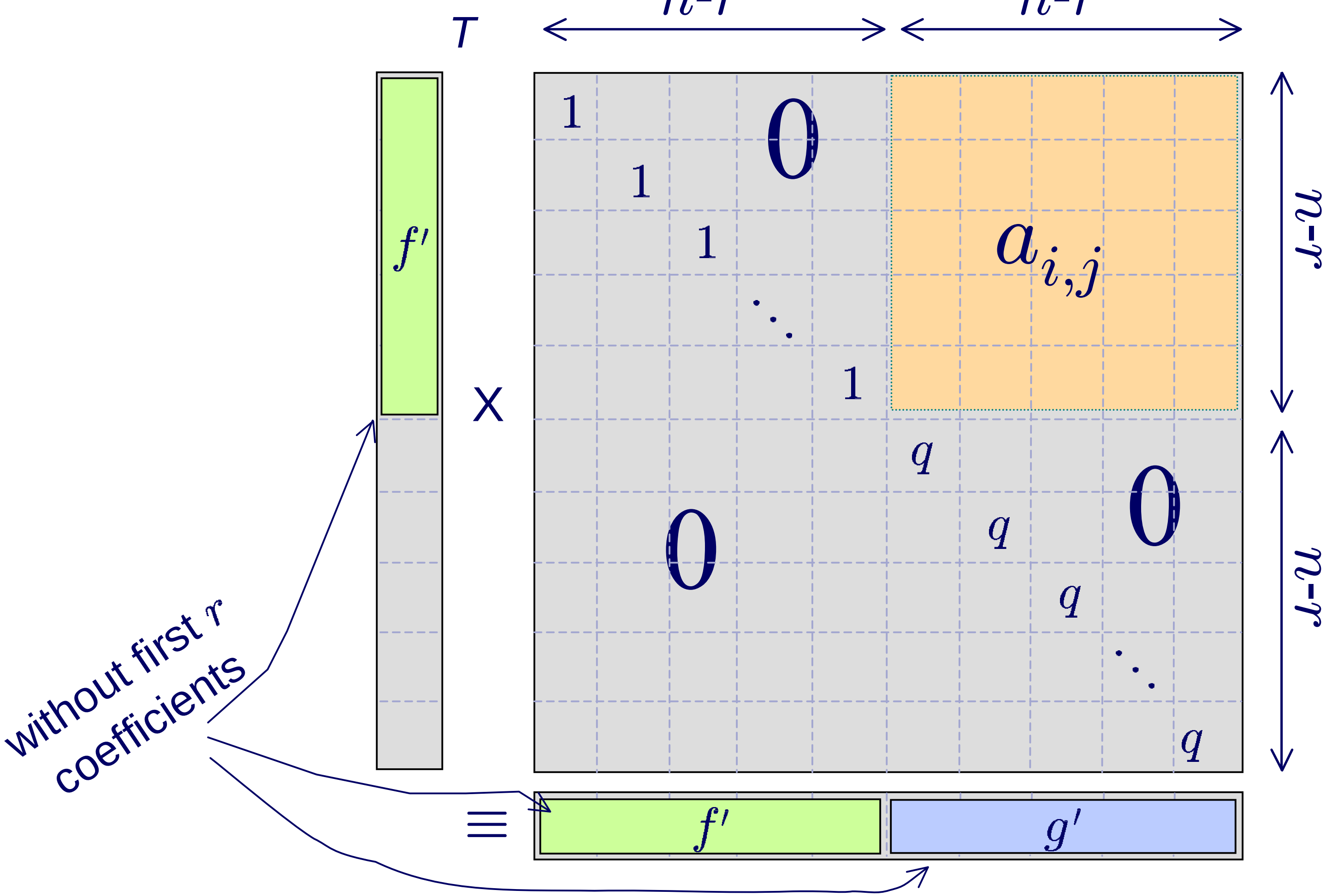
$$\sum_{j=0}^{n-1} f_j h_{i-j \pmod{n}} = 0 \pmod{q} \quad (0 \le i < r)$$

So we can express $f_0, \ldots, f_{r-1}$ in terms of $f_r, \ldots, f_{n-1}$.
By substitution, we get coefficients $a_{i,j}$ ($r \le i, j < n-1$) such that:

$$\sum_{j=r}^{n-1} f_j a_{i,j} = g_i \pmod{q} \quad (r \le i < n-1)$$

"Zero-forced lattice" (cont.)

$$\sum_{j=r}^{n-1} f_j a_{i,j} = g_i \pmod q \quad (r \le i < n-1)$$



Lattice of lower degree

⇓

LLL performs better

As before: tradeoff in choice of r.

Lattice attacks on NTRU: conclusions

- NTRU was proposed with several parameter sets (n,p,q etc.). The smallest set (n=107) was broken using the zero-run lattice attacks.
- We have seen key-recovery attacks. Similar techniques can be used for plaintext-recovery.
- The techniques we saw are the best known passive attacks against NTRU.
- The parameter sets recommended for NTRU are pessimized for these attacks (i.e., chosen so that the gap of the lattices is very small).

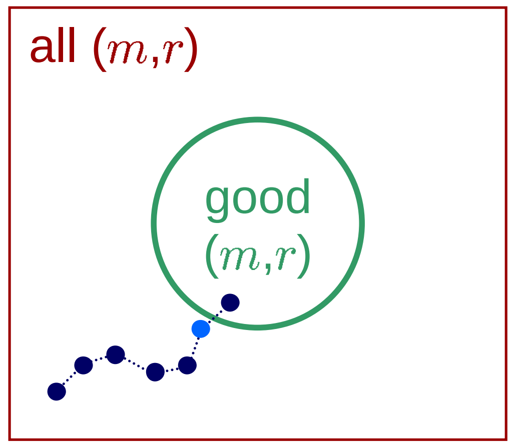
Example: choice of q. By the Guassian heuristic, the shortest vector is of length $\approx \sqrt{1/2\pi e} \cdot \sqrt{2n}(\det L)^{1/2n} = \sqrt{1/\pi e} \cdot \sqrt{nq}$

But decreasing q increases the likelyhood of decryption errors.

Imperfect Decryption Attacks on NTRU

[Proos 2003]

- Decryption failures
- Exploiting them:
  1. Find bad (m,r)
  2. Find “barely bad” (m^*,r)
  3. Find the private key
- Moral



~~encryption scheme~~ secure
against chosen-plaintext attack

↓ generic transformation

encryption scheme ~~secure~~
against chosen-ciphertext attack

NTRU (cont.)

- Fast
- Has small keys
- Different
- Secure

(other) Lattice-based cryptosystems

GGH Cryptosystem

[Goldreich, Goldwasser, Halevi 1997]

- Based directly on the Closest Vector Problem.
- Private key:
  n nearly orthogonal vectors. *
- Public key:
  A random basis $\vec{b}_1, \ldots, \vec{b}_n$ of the lattice spanned by the private key. *
- Encryption: the encryption of message $m_1,\ldots,m_n \in \mathbb{Z}^m$ is $\sum_{i=1}^{n} m_i \vec{b}_i + \vec{r}, \;\; r \in_R \{-\delta, \delta\}^n$
- Decryption: project on private key and round. *
- Breaking: solve a CVP problem. *

GGH Cryptosystem: attack

[Nguyen99]

- Attack
- Moral

Ajtai-Dwork Cryptosystem

[Ajtai, Dwork 1997]

- Like GGH, based directly on a lattice problem.
- As in GGH, key generation creates a random lattice with certain properties. The secret key is some information about the lattice, and the public key is a random basis.
- *
- Marvelous property: security proof is a reduction from *worst-case* of the lattice problem to *average-case* of breaking the scheme.
- Alas, impractical due to huge key size, ciphertext size and message expansion.