An Improved Practical Key Mismatch Attack Against NTRU

Zhen Liu, Vishakha FNU, Jintai Ding, Chi Cheng and Yanbin Pan

PQCrypto 2024, University of Oxford

NTRU

- Introduced by Hoffstein, Pipher and Silverman in 1996.
- A lattice-based public key encryption scheme.
- Standardized by IEEE 1363.1-2008.
- Commercialized: Security Innovation.
- No provable security

NTRU – KMAs before

- In 2000, Hoffstein, Pipher and Silverman firstly proposed a reaction attack against the original NTRU reling on a strong assumption that the upper (lower) wrapping failure only occurs at one coefficient.
- In 2003, Howgrave et al. successfully gave a reaction attack against the padded NTRUs, a infeasible large number of queries to the oracle.
- In 2019, Ding et al. proposed a key mismatch attack on the original NTRU scheme with a linear number of queries.
- In 2021, Zhang et al. successfully mounted a key mismatch attack against NTRU-HRSS based on searching for the optimum binary recovery tree, which has the minimum number of queries.

NTRU cryptosystem

Public Parameter: (N, p, q, d_f, d_g, d_s), $\mathcal{R} = \frac{\mathbb{Z}[X]}{X^N - 1}$ and $\gcd(p, q) = 1$

$\mathcal{T}_{(d_1, d_2)} = \{$trinary polynomials of $\mathcal{R}$ with d_1 entries equal to 1 and d_2 entries equal to $-1\}$

Alice

Bob

Alice:

$$f \xleftarrow{\$} \mathcal{T}_{(d_f+1, d_f)}$$

$$\exists\ f_q^{-1} \in \mathcal{R}_q,\ f_q^{-1} * f = 1 \bmod q$$

$$\exists\ f_p^{-1} \in \mathcal{R}_p,\ f_p^{-1} * f = 1 \bmod p$$

$$g \xleftarrow{\$} \mathcal{T}_{(d_g, d_g)}$$

Alice → Bob: $h = p * g * f_q^{-1}$ (public key)

Bob:

$$m \in \mathbb{Z}_3^N$$

$$s \xleftarrow{\$} \mathcal{T}_{(d_s, d_s)}$$

Bob → Alice: $c = h * s + m$ (ciphertext)

Alice:

$$a = c * f \bmod q$$

$$m = a * f_p^{-1} \bmod p$$

Why it works?

- $$\begin{aligned} a &= c * f \mod q \\ &= p * g * s + m * f \mod q \end{aligned}$$

  If every coefficient of $p * g * s + m * f$ lies in $[-q/2, q/2)$, then

  $$a = p * g * s + m * f$$

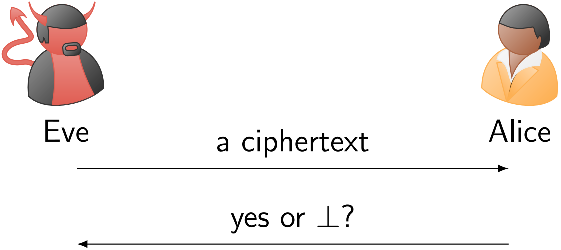
- $$\begin{aligned} a * f_p^{-1} &= m * f * f_p^{-1} \mod p \\ &= m \end{aligned}$$

- $x^i * f$ **is an equivalent private key**, for $0 \leq i \leq N - 1$.

Key Mismatch Attack

Basic Scenario

The attacker in a Key Mismatch Attack has access to a **weaken decryption oracle**, which only tells the ciphertext can be decrypted correctly or not.

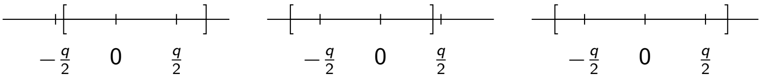


Decryption failure

- **For a ciphertext c that can be decrypted correctly**, construct ciphertexts $\boldsymbol{c_i = c + n * p * x^i}$, $0 \leq i \leq N - 1$, n is a positive integer, we have

$$
\begin{aligned}
c_i * f &= c * f + n * p * x^i * f \bmod q \\
&= a + n * p * x^i * f \bmod q
\end{aligned}
$$

- If **every coefficient** of $\boldsymbol{a + n * p * x^i * f}$ lies in $[-q/2, q/2)$, then $(\boldsymbol{c_i * f} \textbf{ mod } \boldsymbol{q}) * \boldsymbol{f_p^{-1}} \textbf{ mod } \boldsymbol{p = a * f_p^{-1}} \textbf{ mod } \boldsymbol{p = m}$.
- Otherwise we say $\boldsymbol{c_i}$ causes a decryption failure, and define



upper bound overflow **lower bound overflow** overflow on both sides

Hoffstein et al.'s attack [1]

- **Find the smallest** n that there exists a $\boldsymbol{c_i} = \boldsymbol{c} + \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i}$ that causes a decryption failure, for some $0 \leq i \leq N-1$.
- **Assume that only the u-th position of** $\boldsymbol{a} + \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{f}$ exceeds the upper bound $q/2$, for some i and u **is unknown**, then the u-th position of $\boldsymbol{x^i} * \boldsymbol{f}$ is equal to 1.

  $N = 3$:

$$\boldsymbol{a} = (a_0 , a_1 , a_2)$$
$$\boldsymbol{n} * \boldsymbol{p} * \boldsymbol{f} = (n * p * f_0, n * p * f_1, n * p * f_2)$$
$$\boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x} * \boldsymbol{f} = (n * p * f_2, n * p * f_0, n * p * f_1)$$
$$\boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^2} * \boldsymbol{f} = (n * p * f_1, n * p * f_2, n * p * f_0)$$

- By recording the values of i, the attacker can **recover a shifted version of the positions of 1 in** $\boldsymbol{f}$.

[1] Hoffstein, J., Silverman, J.H.: Reaction attacks against the ntru public key cryptosystem (2000), https://ntru.org/f/tr/tr015v2.pdf

Hoffstein et al.'s attack

a special case of upper bound overflow	**a special case of lower bound overflow**	**overflow on both sides**
assume **only one** coefficient of $\boldsymbol{c_i}$ causes decryption failure, **recover** a shifted version of the positions of 1 in f	assume **only one** coefficient of $\boldsymbol{c_i}$ causes decryption failure, **recover** a shifted version of the positions of -1 in f	$\times$

table: the results of Hoffstein et al.'s attack

- **How to detect the type of a decryption failure?**

Motivation

- add the disturbed polynomials $n*p*x^i$ to $\boldsymbol{c} \Rightarrow$ the discontinuous position of $\boldsymbol{f}$.
- add other disturbed polynomials $\triangle$ to $\boldsymbol{c} \Rightarrow$ a consecutive coefficient sequence of $\boldsymbol{f}$?
  - a consecutive coefficient sequence of length k of $\boldsymbol{f}$:

$$f_{i \bmod N}, f_{(i+1) \bmod N}, \cdots, f_{(i+k-1) \bmod N}$$

    e.g., $k = N$ and $i = N-1$, $f_{N-1}, f_0, \cdots, f_{N-2} \Leftrightarrow x * f(x)$
    $k = N$ and $i = N-2$, $f_{N-2}, f_{N-1}, \cdots, f_{N-3} \Leftrightarrow x^2 * f(x)$
  - $(\boldsymbol{c} + \triangle) * \boldsymbol{f} \Rightarrow \boldsymbol{a} + \triangle * \boldsymbol{f}$
  - How to construct $\triangle$?

Observation

For a polynomial $\boldsymbol{t} \in \mathcal{R}$, $\boldsymbol{t} * \boldsymbol{f} = (t_0, t_1, \cdots, t_{N-1}) \begin{pmatrix} f_0 & f_1 & \cdots & f_{N-1} \\ f_{N-1} & f_0 & \cdots & f_{N-2} \\ \vdots & \vdots & & \vdots \\ f_2 & f_3 & \cdots & f_1 \\ f_1 & f_2 & \cdots & f_0 \end{pmatrix}$,

for $0 \le i \le N-1$, the i-th coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is

$$t_{N-1} \cdot f_{i \bmod N} + t_{N-2} \cdot f_{(i+1) \bmod N} + \cdots + t_0 \cdot f_{(i+N-1) \bmod N}.$$

The i-th coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is determined by two consecutive coefficient sequences

$$t_{N-1}, t_{N-2}, \cdots, t_0$$

and

$$f_{i \bmod N}, f_{(i+1) \bmod N}, \cdots, f_{(i+N-1) \bmod N}$$

Some Notations

- $\boldsymbol{c}$: a ciphertext that can be decrypted correctly.

$$\boldsymbol{a} = \boldsymbol{c} * \boldsymbol{f} \bmod \boldsymbol{q}.$$

- $\boldsymbol{n}$: **the smallest positive integer** that there exists a $\boldsymbol{c_i} = \boldsymbol{c} + \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i}$ that causes a decryption failure, for some $0 \leq i \leq N-1$.

$$\boldsymbol{c_i} * \boldsymbol{f} = \boldsymbol{a} + \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{f} \bmod \boldsymbol{q}$$

- $\boldsymbol{c'_i} = \boldsymbol{c} + \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{t}$, where $\sum_{j=0}^{N-1} |t_j| = n$, $0 \leq i \leq N-1$.

$$\boldsymbol{c'_i} * \boldsymbol{f} = \boldsymbol{a} + \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{t} * \boldsymbol{f} \bmod \boldsymbol{q}$$

  - decrypted correctly: $(\boldsymbol{c'_i} * \boldsymbol{f} \bmod \boldsymbol{q}) * \boldsymbol{f_p^{-1}} \bmod \boldsymbol{p} = (\boldsymbol{a} + \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{t} * \boldsymbol{f}) * \boldsymbol{f_p^{-1}} \bmod \boldsymbol{p} = \boldsymbol{m}$.

Key Result

Lemma

For a polynomial $\boldsymbol{t}$ satisfying $\sum_{j=0}^{N-1} |t_j| = n$, if there exists a $\boldsymbol{c}_i'$ that causes a decryption failure, for $0 \le i \le N-1$, then $\|\boldsymbol{t} * \boldsymbol{f}\|_\infty = \boldsymbol{n}$.

- upper bound overflow: the maximal coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is n.
- lower bound overflow: the minimal coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is $-n$.
- overflow on both sides: $\|\boldsymbol{t} * \boldsymbol{f}\|_\infty = n$.

Eve

ciphertexts $\boldsymbol{c}_0', \cdots, \boldsymbol{c}_{N-1}'$ →

Alice

← $\|\boldsymbol{t} * \boldsymbol{f}\|_\infty \stackrel{?}{=} n$

The framework of our attack

1. **Choose** a ciphertext $\boldsymbol{c}$ that can be decrypted correctly.
2. **Find** the smallest n that there exists a $\boldsymbol{c_i} = \boldsymbol{c} + \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i}$ that causes a decryption failure, for some $0 \leq i \leq N-1$.
3. **Construct** different $\boldsymbol{t}$ with $\sum_{j=0}^{N-1} |t_j| = n$, and use $\boldsymbol{c}_i' = \boldsymbol{c} + \boldsymbol{p} * \boldsymbol{t} * \boldsymbol{x^i}$ to recover consecutive sequence $l_1, l_2, \cdots, l_M$ in $\boldsymbol{f}$ one position at a time.
4. **Select** a subsequence $l_m, \cdots, l_M$ to continue recovery and obtain a newly consecutive sequence $l_m, \cdots, l_M, \cdots, l_{M_1}$.
5. **Recover** the whole $\boldsymbol{f}$ by repeating this process.

$$\overbrace{l_1, l_2, \cdots, \underbrace{l_m, \cdots, l_M}_{}}^{}, \underbrace{\cdots, l_{M_1}}_{}, \cdots$$

Recover the next position

Input: $l_1, \cdots, l_{k+1}$ with $k \geq 0$

Output: l_{k+2}

1. **set** $\boldsymbol{t} = (0, \cdots, 0, n - \sum_{j=0}^{k} |l_{1+j}|, l_{k+1}, \cdots, l_2, l_1)$;
2. If there exits a $\boldsymbol{c}_i' = \boldsymbol{c} + \boldsymbol{p} * \boldsymbol{t} * \boldsymbol{x}^{\boldsymbol{i}}$ that causes a decryption failure, return $l_{k+2} = 1$;
3. Else **set** $\boldsymbol{t} = (0, \cdots, 0, -(n - \sum_{j=0}^{k} |l_{1+j}|), l_{k+1}, \cdots, l_2, l_1)$;
4. If there exits a $\boldsymbol{c}_i' = \boldsymbol{c} + \boldsymbol{p} * \boldsymbol{t} * \boldsymbol{x}^{\boldsymbol{i}}$ that causes a decryption failure, return $l_{k+2} = -1$;
5. return $l_{k+2} = 0$.

Recover a consecutive sequence of length 2

Assume $l_1 = 1$ to determine the next coefficient l_2:

1. $\boldsymbol{t} = (0, 0, \cdots, 0, n - |l_1|, l_1) \xrightarrow{\text{failure}} l_2 = 1$
2. $\boldsymbol{t} = (0, 0, \cdots, 0, -(n - |l_1|), l_1) \xrightarrow{\text{failure}} l_2 = -1$
3. The attacker will **only set** $l_2 = 0$ when **neither of the two choices for $\boldsymbol{t}$ can cause decryption failure.**

- **overflow in the upper bound** : the maximal coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is n $\Rightarrow l_1, l_2$ is in $\boldsymbol{f}$.
- **overflow in the lower bound** : the minimum coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is $-n \Rightarrow l_1, l_2$ is in $-\boldsymbol{f}$.
- **overflow on both sides** : The recovered sequence l_1, l_2 is in $\boldsymbol{f}$ or $-\boldsymbol{f}$.

Recover a consecutive sequence of length 3

the recovered $l_1 = 1, l_2 = 0$ in the case of upper bound overflow:

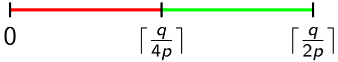
1. $\boldsymbol{t} = (0, 0, \cdots, 0, n - |l_1| - |l_2|, l_2, l_1) \xrightarrow{\text{failure}} l_3 = 1$
   - Every coefficient of $\boldsymbol{t} * \boldsymbol{f}$ has the form of
   $$1 \cdot f_j + 0 \cdot f_{j+1} + (n-1) \cdot f_{j+2}.$$
   - the maximal coefficient of $\boldsymbol{t} * \boldsymbol{f}$ is n.
   - failure $\Rightarrow f_j = 1, f_{j+1} \in \{\pm 1, 0\}, f_{j+2} = 1$, for some j.
   - $f_{j+1} \neq 0 \Rightarrow l_2 \neq 0$
   - $f_j = 1, f_{j+1} = 0, f_{j+2} = 1 \Rightarrow l_1 = 1, l_2 = 0, l_3 = 1$
2. $\boldsymbol{t} = (0, 0, \cdots, 0, -(n - |l_1| - |l_2|), l_2, l_1) \xrightarrow{\text{failure}} l_3 = -1$
3. Otherwise, $l_3 = 0$

The size of M

$$\overbrace{l_1, l_2, \cdots, \underbrace{l_m, \cdots, l_M}_{}}^{}, \cdots, l_{M_1}, \cdots$$

- When $n \geq (2d_f + 1)$, we have $M = N$, which means the recovered coefficient sequence $l_1, \cdots, l_M$ is in $\boldsymbol{f}$ or $-\boldsymbol{f}$ of length N.
- When $n < (2d_f + 1)$, by the **negative hypergeometric distribution**, the expectation of M is $\frac{n \cdot (N+1)}{2d_f+2}$.

- **binary search to find n:**
  - **upper bound** on n: $\lceil \frac{q}{2p} \rceil$
  - monitor whether there exists a $\boldsymbol{c_i}$ that causes a decryption failure or not



- a polynomial $\boldsymbol{t} \rightarrow N$ ciphertexts $\boldsymbol{c}'_i = \boldsymbol{c} + \boldsymbol{p} * \boldsymbol{t} * \boldsymbol{x^i}$.
- one coefficient $\rightarrow 2N$ ciphertexts in the worst case.
- **Complexity:** $O(N^2)$ in the worst case.

Special Case: c=0

$$\boldsymbol{c} = \boldsymbol{0} \Rightarrow \boldsymbol{c_i} * \boldsymbol{f} = \boldsymbol{n} * \boldsymbol{p} * \boldsymbol{x^i} * \boldsymbol{f} \bmod q \Rightarrow n = \lceil \frac{q}{2p} \rceil$$

- All $\boldsymbol{c_i'} = p * \boldsymbol{t} * \boldsymbol{x^i}$ cause decryption failures at the same time.
- For a polynomial $\boldsymbol{t}$ satisfying $\sum_{j=0}^{N-1} |t_j| = \lceil \frac{q}{2p} \rceil$, use $\boldsymbol{c'} = p * \boldsymbol{t}$ to recover the consecutive coefficients one by one position until the number of nonzero elements reaches

$$\min\{\lceil \frac{q}{2p} \rceil, 2d_f + 1\}.$$

Experimental Results

N	q	p	d_g	E	Q	Success Rate	Running Time(second)
443	2048	3	143	**739**	**742**	100%	48.75
743	2048	3	247	**1239**	**1238**	100%	315.80
821	4096	3	255	**1369**	**1387**	100%	455.38

- $\boldsymbol{g}$ is trinary, use $\boldsymbol{c}' = \boldsymbol{c} + \boldsymbol{h} * \boldsymbol{t} = \boldsymbol{h} * \boldsymbol{t}$ to finish the recovery of $\boldsymbol{g}$.
- Q: the corresponding number of queries in our attack:
  - one coefficient $\Rightarrow$ 2 ciphertexts in the worst case.
  - $Q \approx 2N - d_g$.
- E: the lower bound on the minimum average number of queries from Qin et al.'s work.
- When $N = 443$ and $N = 821$, we have $M = N$.
- When $N = 743$, M is about 515 in theory.

Summary

- The attack **gets rid of the assumptions** used in Hoffstein et al.'s attack.
- The attack in the special case **has the number of queries to the KMO closest to the lower bound** on the minimum average number of queries at Asiacrypt 2021.
- The attack can be **applied to any valid ciphertext**, making it difficult to be easily detected.

Thank you & Questions ?