A Simple and Efficient Key Reuse Attack on NTRU Cryptosystem

Abstract. In 1998, Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman introduced the famous NTRU cryptosystem, and called it "A ring-based public key cryptosystem". Actually, it turns out to be a lattice based cryptosystem that is resistant to Shor's algorithm. There are several modifications to the original NTRU and two of them are selected as round 2 candidates of NIST post quantum public key scheme standardization.
In this paper, we present a simple attack on the original NTRU scheme. The idea comes from Ding et al.'s key mismatch attack. Essentially, an adversary can find information on the private key of a KEM by not encrypting a message as intended but in a manner which will cause a failure in decryption if the private key is in a certain form. In the present, NTRU has the encrypter generating a random polynomial with "small" coefficients, but we will have the coefficients be "large". After this, some further work will create an equivalent key.

Keywords: Lattice · Encryption · NTRU · Cryptanalysis · KEM.

1 Introduction

1.1 Background

Public-key cryptosystems have undergone a revolutionary breakthrough in cryptography since its invention in 1976 [5]. Today, public-key cryptosystems have become an indispensable part of modern communication systems. RSA, DSA, ECDSA, and similar cryptosystems are widely in use providing a secure way of exchanging keys to be used by the more efficient symmetric-key cryptosystems. Hence, the security of data relies on its weakest part, which is the transfer of the symmetric key by the public-key cryptosystem. The security of those systems are based on the hardness of classical number theory problems such as integer prime factorization or discrete logarithm. These problems are thought difficult enough to resist attack from classical computing technology. However, Peter shor [19] from Bell Laboratories theoretically showed that some hard number theory problems such as Integer Prime Factorization Problem and the Discrete Logarithm Problem could be solved if a quantum computer were built. Peter Shor's polynomial-time integer factorization algorithm has led a potential crisis to crytopraphy. It is now evident that new public-key cryptosystems that have potential to resist quantum algorithms are urgently needed.

1.2 Post-Quantum Cryptography Standardization

Due to the rapid development of quantum computers, NIST believes that it is prudent to begin developing standards for post-quantum cryptography. Moreover, it is reasonable to plan ahead because a transition to post-quantum cryptography will not be simple. A significant effort will be required in order to develop, standardize, and deploy new post-quantum cryptosystems. The call for proposals started in December 2016. NIST expects to perform multiple rounds of evaluation over a period of three to five years. The goal of this process is to select a number of acceptable candidate cryptosystems for standardization. These new standards will be used as quantum resistant counterparts to existing standards. The evaluation will be based on the following three criteria: security, cost, and algorithm implementation characteristics [15]. By the end of 2017, 23 signature schemes and 59 encryption/KEM schemes were submitted, of which 69 participated in the first round, 26 of these survived the second round. Two of these submissions to the second round are based on the original NTRU scheme, with some modifications [4][3].

1.3 Lattice based Cryptosystem

Lattice-based public-key cryptosystems are believed to be one of the candidates that have potential to resist quantum attack. The most important computational problem in lattice-based cryptosystems is the shortest vector problem (SVP) which asks to find the length of the shortest non-zero vector in a lattice. This problem is believed hard to solve efficiently even with a quantum computer. SVP also derives other interesting problems such as the learning with error (LWE) problem introduced by Oded

Regev in 2005 along with an encryption system [17]. In 2012, Ding et al. published the first key exchange system based on LWE problem that is provably secure [9]. It can be easily proven that the security of NTRU depends on the difficulty to solve the SVP in NTRU lattice. In this paper, we try to find short vectors in NTRU lattice that can be used as equivalent keys if both private and public keys are reused. We hope to show that certain implementations of NTRU are breakable due to their use in symmetric key exchanges.

1.4 Previous Attacks against NTRU

There have been several attacks on NTRU, namely brute force attack, multiple transmission attack etc. The closest attack to ours is by Jaulmes and Joux [13]. This is a chosen ciphertext attack in which they use intersection polynomial of the private key polynomials f and g. The i^{th} coefficient of this intersection polynomial is defined to be 1 if f and g both have their i^{th} coefficient equal to 1, -1 if f and g both have their i^{th} coefficient equal to -1, and 0 otherwise.

The key difference between Jaulmes et al.'s and ours attack is that while our attack could be viewed as a ciphertext attack, we only actually care about if the key is successfully transferred, not the actual key that would get transferred. And, this is where we use a key mismatch attack, which is explained in the following sections. For our attack does not need the cleartexts corresponding to the feeded ciphertexts as this works just on the basis of the decryption being successful or not. On contrary, the attack by Jaulmes et al. works under the assumption that the attacker has an access to a decryption oracle to create ciphertext/cleartext pairs, that is, this attack works on the communication platform, whereas, although our attack could be viewed as a chosen ciphertext attack, it works on a key mismatch level.

1.5 Key Reuse Attack

Key reuse actually is commonly used in the internet standard. For example, the preshared keys in TLS 1.3 [18] are allowed to be reused. However, key reuse in lattice based cryptosystem has high potential of risk due to the key reuse attack. There are currently two types of key reuse attack, signal leakage attack and key mismatch attack. In this paper, we will focus on key mismatch attack. The goal of key mismatch attack is to create an equivalent private key by verifying if the shared information generated by two parties agrees or not several times.

In 2015, NSA warned NIST Post-Quantum candidates against active attacks[14]. The first key resue attack was proposed by Fluhrer on the leakage of secret keys of ring-LWE key exchange when one party reuses the public key [10]. Later, Ding et al. gave a key leakage attack on the LWE key exchange[6]. Besides, Ding et al. also introduced a key mismatch attack on RLWE key exchange without signal leakage [8]. In 2019, Bauer et al. analyzed the case when public key is resued in NewHope which is a second round candidate of NIST post quantum standard process [2]. Yue Qin et al. then proposed an optimized key mismatch attack on NewHope that improves Bauer's method [16]. Most recently, Ciprian Băetu et al. extended the key reuse attack to quantum variant where the adversary has quantum access to a decryption oracle [1].

1.6 Our contribution

We will present an attack on original NTRU (1998) [11] based on the fact that key mismatch is accessible to the attacker. We will show that by choosing certain ephemeral keys, the result of the decryption will make it possible for the attacker to create equivalent private keys. First, we will recall the original design of NTRU due to Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Next, we will describe the method to obtain the longest chain of consequent coefficients that consists of either consecutive 1's or consecutive -1's of a private key polynomial. This step can be done due to the special structure of the ring and the construction of the private key. Having the longest chain, one can guess the remaining coefficients of a private key polynomial by using the longest chain as an anchor and by using the effective choices of ephemeral keys. Last but not least, we will provide the experimental success rate of our method.

2 Description of the NTRU Cryptosystem [11]

2.1 Notations and Definitions

For the rest of the paper, we assume that N is an odd prime number, and q is a even integer.
R, R_p and R_q denote the quotient rings $\frac{\mathbb{Z}[x]}{x^N-1}$, $\frac{\mathbb{Z}_p[x]}{x^N-1}$ and $\frac{\mathbb{Z}_q[x]}{x^N-1}$ respectively.
A polynomial is ternary if its coefficients are in $\{-1,0,1\}$.
Let $\mathscr{T}$ denote the set of non-zero ternary polynomials of degree at most $N-1$.
$\mathscr{T}(d_1,d_2)$ denotes a subset of $\mathscr{T}$ consisting of polynomials that have exactly d_1 coefficients equal to 1 and d_2 coefficients equal to -1.

Centerlift Let $a(x) \in R_q$. The centered lift of $a(x)$ to R is the unique polynomial $a'(x) \in R$ satisfying $a'(x)\ mod q = a(x)$ whose coefficients are chosen in the interval from $-q/2$ to $q/2-1$.

Multiplication of polynomials in $\boldsymbol{R_q}$ Let $\mathbf{f}(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{N-1}x^{N-1}$ and $\mathbf{g}(x) = b_0+b_1x+b_2x^2+\cdots+b_{N-1}x^{N-1}$ be two polynomials in the ring R_q. The product $\mathbf{f}(x) \star \mathbf{g}(x)$ in R_q can be expressed in the matrix form:

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} \end{bmatrix} \begin{bmatrix} b_0 & b_1 & \cdots & b_{N-1} \\ b_{N-1} & b_0 & \cdots & b_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ b_1 & b_2 & \cdots & b_0 \end{bmatrix}$$

The resultant vector gives the coefficients of $\mathbf{f}(x) \star \mathbf{g}(x)$ in R_q.

2.2 The NTRU scheme

In this section, we describe the NTRU scheme.

Keygen:

- Randomly choose $\mathbf{f} \in \mathcal{T}(d+1,d)$ such that $\mathbf{f}$ is invertible in both R_p and R_q. Denote the inverses of $\mathbf{f}$ by $\mathbf{f}_p$ and $\mathbf{f}_q$ in R_p and R_q respectively.
- Randomly choose $\mathbf{g} \in \mathcal{T}(d,d)$.
- Let $\mathbf{h} = p\mathbf{g} \star \mathbf{f}_q \mod q$.
- Public key is the polynomial $\mathbf{h}$, and a private key pair is $(\mathbf{f},\mathbf{g})$.

Encryption:

- Let $\mathbf{m} \in R_p$ be a message.
- Randomly choose $\mathbf{r} \in \mathcal{T}(d,d)$.
- Compute $\mathbf{c} = \mathbf{r} \star \mathbf{h} + \mathbf{m} \mod q$.

Decryption:

- Compute $\mathbf{a} = \mathbf{c} \star \mathbf{f} \mod q$.
- Center lift $\mathbf{a}$ to R and recover the message by computing $\mathbf{f}_p \star \mathbf{a} \mod p$.

Definition 1. *Two NTRU private keys* $(\mathbf{f},\mathbf{g})$ *and* $(\hat{\mathbf{f}},\hat{\mathbf{g}})$ *are said to be equivalent if they lead to the same private key, i.e.,* $p\mathbf{g} \star \mathbf{f}_q \mod q = \mathbf{h} = p\hat{\mathbf{g}} \star \hat{\mathbf{f}}_q \mod q$.

Definition 2. *The NTRU assumption is that given* $\mathbf{h}$*, it is hard to find* $\mathbf{f}$ *and* $\mathbf{g}$.

The NTRU assumption can be formulated to a SVP in the NTRU lattice which is spanned by the rows of the 2N by 2N matrix:

$$\begin{bmatrix} I & \mathbf{h} \\ 0 & qI \end{bmatrix}$$

where I is the N dimensional identity matrix, $\mathbf{h}$ stands for the cyclical permutations of the coefficients of $\mathbf{h}$. Moreover 0 represents the zero matrix, and qI is q times the indentity matrix I.

Remark 1. By Proposition 6.48 in [12], if the NTRU parameters (N,p,q,d) are chosen to satisfy $q > (6d+1)p$, the decryption process will never fail.

Remark 2. The inequality in **Remark 1** guarantees that the coefficients of $\mathbf{a}$ do not change when it moves from R_q to R. Therefore, it ensures the correctness of decryption. However, the attacker has the freedom to choose the ephemeral key $\mathbf{r}$, and if $\mathbf{r}$ is chosen honestly in $\mathcal{T}(d,d)$, the decryption will be successful and no information is revealed. Hence, the attacker has to choose a special $\mathbf{r}$ outside of the set $\mathcal{T}(d,d)$ which will fail the decryption so that he can get some information about the private key.

3 Our Attack

It is the fundamental assumption of the attack that a fixed NTRU public/private key pair will be used repeatedly as KEM (key encapsulation mechanism). The strategy, following that of Ding's key reuse attack to LWE and ring LWE [7], will be to systematically abuse the freedom given to the encrypting party to choose the coefficients of $\mathbf{r}$ during the encryption step $\mathbf{c} = \mathbf{r} \star \mathbf{h} + \mathbf{m}$. We assume the message $\mathbf{m}$ to be 0 in our attack. An appropriate choice for $\mathbf{r}$ will cause the key exchange to fail, meaning a mismatch of symmetric keys instead of a proper exchange, if and only if the private key polynomial $\mathbf{g}$ is of a specific shape. Through repeated uses, enough of the form of $\mathbf{g}$ can be recovered to create an equivalent polynomial $\hat{\mathbf{g}}$. From there it is easy to create an equivalent $\hat{\mathbf{f}}$ to $\mathbf{f}$ forming an equivalent private key $(\hat{\mathbf{f}}, \hat{\mathbf{g}})$.

For a given NTRU private key $(\mathbf{f}, \mathbf{g})$, let

$$\mathbf{g}(x) = \sum_{i=0}^{N-1} g_i x^i \text{ where } g_i \in \{-1, 0, 1\}.$$

3.1 Finding an Equivalent g

Finding a longest chain in g First, we find a longest chain of consequent coefficients of $\mathbf{g}$ that consists of either consecutive 1's or consecutive -1's. From now on, for our convenience, whenever we say a longest chain in $\mathbf{g}$, we mean a longest chain that is described above, unless otherwise mentioned. By having such a longest chain, we mean that there exists a unique integer k such that for some $i \in \{0, \cdots, N-1\}$, we have $g_{imodN} = g_{(i-1)modN} = \cdots = g_{(i-k+1)modN} = v \in \{-1, 1\}$. An appropriate choice of an ephemeral key $\mathbf{r}$ will lead us to find this chain.

In the decryption process, $\mathbf{a} := \mathbf{f} \star \mathbf{c}$ in R_q, so $\mathbf{a} := \mathbf{f} \star \mathbf{h} \star \mathbf{r} = \mathbf{f} \star (p\mathbf{f}_q \star \mathbf{g}) \star \mathbf{r} = p\mathbf{g} \star \mathbf{r}$ in R_q. Note that $\mathbf{a}$ can be expressed as a multiplication of matrices:

$$\begin{bmatrix} r_0 \ r_1 \ \cdots \ r_{N-1} \end{bmatrix} \begin{bmatrix} pg_0 & pg_1 & \cdots & pg_{N-1} \\ pg_{N-1} & pg_0 & \cdots & pg_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ pg_1 & pg_2 & \cdots & pg_0 \end{bmatrix}$$

Since it is easier to view the multiplication of two polynomials as a multiplication of two matrices like above, therefore, for our convenience we will look the polynomial multiplication as a matrix multiplication.

Now, observe that, for some $j \geq 1$, if we set $\mathbf{r}$ with coefficients $r_0 = r_1 = \cdots = r_{j-1} = \left\lceil \frac{q}{2pj} \right\rceil$ and $r_j = r_{j+1} = \cdots = r_{N-1} = 0$, then we get

$$[a_0 \cdots a_{N-1}] = \left[\left\lceil \frac{q}{2pj} \right\rceil \cdots \left\lceil \frac{q}{2pj} \right\rceil 0 \cdots 0 \right] \begin{bmatrix} pg_0 & pg_1 & \dots & pg_i & \dots & pg_{N-1} \\ pg_{N-1} & pg_0 & \dots & pg_{i-1} & \dots & pg_{N-2} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ pg_1 & pg_2 & \dots & pg_{i+1} & \dots & pg_0 \end{bmatrix}.$$

Thus, $\mathbf{a} = (a_\ell)_{0\le\ell\le N-1}$, where $a_\ell = p\left\lceil\frac{q}{2pj}\right\rceil(g_{\ell modN}+\cdots+g_{(\ell-j+1)modN})$.

Note that, if the length of a longest chain is k, then $M := \max\{|a_\ell| : 0\le\ell\le N-1\}$ is equal to $\left|j\left\lceil\frac{q}{2pj}\right\rceil pv\right| = j\left\lceil\frac{q}{2pj}\right\rceil p$ for $j\le k$,

and M belongs to the set $\{pk\left\lceil\frac{q}{2p(k+1)}\right\rceil, \, p(k-1)\left\lceil\frac{q}{2p(k+1)}\right\rceil\}$ for $j = k+1$.

Therefore, for $j\le k$, M goes outside of the interval from $-q/2$ to $q/2-1$, and for $j = k$, M lies inside of the above interval.

We say that a mismatch occurs (i.e. a decryption fails) when a coefficient of **a** goes outside of the interval from $-q/2$ to $q/2-1$ (the chosen representatives for the centerlift), and a match occurs (i.e. a decryption is successful) when all the coefficients of **a** are inside of the above interval.

Hence, a mismatch occurs for $j\le k$, and a match occurs for $j = k+1$.

We will use the above idea to obtain a longest chain of either consecutive 1's or consecutive -1's in the coefficients of **g**. We just set the first j coefficients of **r** equal to $\lceil\frac{q}{2pj}\rceil$ and rest equal to 0, and see whether the decryption goes through or not. If the decryption fails, then this tells us that the length of the chain is atleast j. We keep trying until the decryption goes through. If the first time decryption succeeds when the first $k+1$ coefficients of **r** are equal to $\lceil\frac{q}{2p(k+1)}\rceil$, then we immediately know that **g** has a longest chain of length k.

Finding the remaining coefficients of g We will use the longest chain as an anchor to test the other coefficients of **g** in relation to the chain. Let's assume that a longest chain in **g** starts at $(i-k+1)^{th}$ position and ends at i^{th} position, that is, coefficients of **g** look like $[g_0,\cdots,g_{i-k},v,\cdots,v,g_{i+1},\cdots,g_{N-1}]$, where $v\in\{-1,1\}$.

To find the value of $g_{(i+t)modN}$ in relation to the longest chain, one tests two values for **r**. First, choose

$$r_0 = r_1 = \cdots = r_k = r_{(k+t)modN} = \left\lceil\frac{q}{2p(k+1)}\right\rceil \mathrm{P}$$

and $r_j = 0$ otherwise. See if a mismatch occurs.

Second, choose

$$r_0 = r_1 = \cdots = r_k = \left\lceil\frac{q}{2p(k+1)}\right\rceil, \, r_{(k+t)modN} = -\left\lceil\frac{q}{2p(k+1)}\right\rceil$$

and $r_j = 0$ otherwise. See if a mismatch occurs.

If only the first choice of **r** gives a mismatch, then $g_{(i+t)modN} = v$. And, if only the second choice of **r** gives a mismatch, then $g_{(i+t)modN} = -v$. If neither of the above choice gives a mismatch then $g_{(i+t)modN} = 0$. All three cases follow the stated rule for a mismatch.

Now it may happen that both the positive and negative choice for $r_{(k+t)modN}$ cause a mismatch. This could happen if there were two chains of length k of consecutive 1's or consecutive -1's in **g** with both coefficients at t places to the right of these chains agreed with the respective chains and had opposite signs. Now, one cannot simply pick one of these two values for we want to know the relationship of all the coefficients of $\mathbf{g}$ with a particular chain, not just one of these chains. To prevent this, one can simply enlarge the anchor that tests the coefficients of $\mathbf{g}$.

One treats the cases of when both the choices of $r_{(k+t)modN}$ cause a mismatch as undetermined for the moment. When a determined case occurs, say when we test $r_{(k+t)modN}$ and get the value of $g_{(i+t)modn} = \mu \in \{-1, 0, 1\}$, then to check the value of $g_{(i+t')modN}$ one uses

$$r_0 = r_1 = \cdots = r_k = \left\lceil \frac{q}{2p(k+2)} \right\rceil, \; r_{(k+t)modN} = \mu \left\lceil \frac{q}{2p(k+2)} \right\rceil$$

and a corresponding choice for $r_{(k+t')modN}$ to see if $g_{(i+t')modN}$ is v or $-v$. As it is less likely that there are two chains of consecutive 1's or consecutive -1's with the same relationship with the tth place to the right of the chain but differing relationships to the t'th place to the right, the number of undetermined cases will go down. In particular, we can recheck the previous undetermined cases with this new anchor. In a similar manner, one can obviously increase the size of the anchor until all the cases are determined. Due to the effect of the ceiling function, the number of nonzero elements for our choice of $\mathbf{r}$ cannot be two large for it would cause false mismatches. Our experimental results show that this is not an issue in practice for all the coefficients will be determined well before this happens.

3.2 Finding an Equivalent f

Once we obtain an equivalent key $\hat{\mathbf{g}}$ by going through the process described in Section 3.1, we can construct an equivalent key $\hat{\mathbf{f}}$ for the other private key $\mathbf{f}$. By nature of its construction, $\hat{\mathbf{g}}$ can differ from $\mathbf{g}$ only by a sign and a shifting of its coefficients. That is, if $\mathbf{g} = \sum_{i=0}^{N-1} g_i x^i \in \mathcal{T}(d,d)$ then for some integer m,

$$\hat{\mathbf{g}} = v \sum_{i=0}^{N-1} g_{(i+m)\mathrm{mod}N} x^i = v x^m \sum_{i=0}^{N-1} g_i x^i = v x^m \mathbf{g},$$

where v is simply 1 if the longest chain in $\mathbf{g}$ consists of indeed 1's, and is -1 if the said chain is actually made of -1's. We note that x^m has an inverse x^{n-m} in both R_q and R_p.

Thus, we see that $\mathbf{h} = \mathbf{f}_q \star \mathbf{g} = v x^{n-m} \mathbf{f}_q \star \hat{\mathbf{g}}$. Let us denote $\hat{\mathbf{f}_q} = v x^{n-m} \mathbf{f}_q$. So, we may write $\mathbf{h} = \hat{\mathbf{f}_q} \star \hat{\mathbf{g}}$. Further, let us write $\hat{\mathbf{f}} = \hat{\mathbf{f}_q}^{-1} = v x^m \mathbf{f}$, where we view each polynomial as a member of R_q. Since we already know $\hat{\mathbf{g}}$ and $\mathbf{h}$, one can easily find $\hat{\mathbf{f}_q}$ which results in $\hat{\mathbf{f}}$ by solving a linear system of equations and that linear system of equations can be solved efficiently in polynomial time.

3.3 Experimental Results

Below is a table expressing some experimental results on the mismatch attack in finding an equivalent $\mathbf{g}$ for different parameters of NTRU. Each parameter set was ran 1000 times. In each instance we increased the size of the anchor until it had 12 nonzero entries. We also recorded the average time in seconds T and average number of key exchanges K needed to find the equivalent $\mathbf{g}$. The computer we used has a Intel Core i7-9700, 8 Core, 12MB Cache, 3.0Ghz. The programming language was Magma version: V2-24.

Parameters (N, p, q, d)	Success Rate	T	K
(256, 3, 2048, 75)	100%	0.15425	507.648
(512, 3, 2048, 150)	100%	0.61813	1019.051
(1024, 3, 2048, 300)	100%	2.47104	2042.381

4 Conclusion

We presented an efficient key reuse attack against an original NTRU cryptosystem proposed by Jeffrey et al. Our experimental results show that this attack is very efficient with 100 percent probability of success for the stated parameter sets.

Bibliography

[1] Ciprian Băetu, F. Betül Durak, Loïs Huguenin-Dumittan, Abdullah Talayhan, and Serge Vaudenay. Misuse attacks on post-quantum cryptosystems. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 747–776, Cham, 2019. Springer International Publishing.

[2] Aurélie Bauer, Henri Gilbert, Guénaël Renault, and Mélissa Rossi. Assessment of the key-reuse resilience of newhope. In *Cryptographers' Track at the RSA Conference*, pages 272–292. Springer, 2019.

[3] Daniel J Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime: reducing attack surface at low cost. In *International Conference on Selected Areas in Cryptography*, pages 235–260. Springer, 2017.

[4] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M Schanck, Peter Schwabe, William Whyte, and Zhenfei Zhang. Algorithm specifications and supporting documentation. 2019.

[5] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[6] Jintai Ding, Saed Alsayigh, RV Saraswathy, Scott Fluhrer, and Xiaodong Lin. Leakage of signal function with reused keys in rlwe key exchange. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[7] Jintai Ding, Chi Cheng, and Yue Qin. A simple key reuse attack on lwe and ring lwe encryption schemes as key encapsulation mechanisms (kems). Cryptology ePrint Archive, Report 2019/271, 2019. https://eprint.iacr.org/2019/271.

[8] Jintai Ding, Scott Fluhrer, and Saraswathy Rv. Complete attack on rlwe key exchange with reused keys, without signal leakage. In *Australasian Conference on Information Security and Privacy*, pages 467–486. Springer, 2018.

[9] Jintai Ding, Xiang Xie, and Xiaodong Lin. A simple provably secure key exchange scheme based on the learning with errors problem. *IACR Cryptology ePrint Archive*, 2012:688, 2012.

[10] Scott R Fluhrer. Cryptanalysis of ring-lwe based key exchange with key share reuse. *IACR Cryptology ePrint Archive*, 2016:85, 2016.

[11] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium*, pages 267–288. Springer, 1998.

[12] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[13] Éliane Jaulmes and Antoine Joux. A chosen-ciphertext attack against ntru. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 20–35, London, UK, UK, 2000. Springer-Verlag.

[14] Daniel Kirkwood, Bradley C Lackey, John McVey, Mark Motley, Jerome A Solinas, and David Tuller. Failure is not an option: standardization issues for post-

quantum key agreement. In *Workshop on Cybersecurity in a Post-Quantum World*, 2015.

[15] National Institute of Standards and Technology. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. Technical report, National Institute of Standards and Technology, 2017.

[16] Yue Qin, Chi Cheng, and Jintai Ding. A complete and optimized key mismatch attack on nist candidate newhope. In Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors, *Computer Security – ESORICS 2019*, pages 504–520, Cham, 2019. Springer International Publishing.

[17] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[18] Eric Rescorla. The transport layer security (tls) protocol version 1.3. *RFC*, 8446:1–160, 2018.

[19] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.