A Simple Key Reuse Attack on Ntru Cryptosystem

Jintai Ding, Joshua Deaton, Zheng Zhang, Kurt Schmidt, and Vishakha

Department of Mathematical Sciences, University of Cincinnati

Abstract. In 1998, Jerey Hostein, 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 [18] 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. People realize 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 [14]. 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][2].

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 [16]. 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 Key Reuse Attack

Key reuse actually is commonly used in the internet standard. For example, the pre-shared keys in TLS 1.3 [17] 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 2005, NSA warns NIST Post-Quantum candidates against active attacks[13]. 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 an key leakage attack on the LWE key exchange[6]. Besides, Ding et al also introduced an 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 [1]. Yue Qin el al. then proposed an optimized key mismatch attack on NewHope that improves Bauer's method [15]. Most recently, Ciprian Băetu et al. extended the key reuse attack to quantum variant where the adversary has quantum access to a decryption oracle [3].

1.5 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 attacker to create equivalent private keys. First, we will recall the original design of Ntru due to Jerey Hostein, Jill Pipher, and Joseph H. Silverman. Next, we will describe the method to obtain the longest consecutive nonzero chain in the coefficients 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 by using the effective choices of ephemeral keys. In this step, one may get several ambiguous cases, but we will explain why these ambiguous case does not matter in finding the next coefficient and how we fill them. To find an equivalent key for the other private key, we will simply solve a linear equation by the design construction. Last but not least, we will provide the experimental success rate of our method, and analyze that why in very low probability, our method fails.

2 The Ntru cryptosystem [11]

2.1 Definitions

For the rest of the paper, we assume that n is an odd prime number.
The representatives of $\mathbb{Z}_q$ is defined to be $\{-\frac{q-1}{2}, \cdots, 0, \cdots, \frac{q-1}{2}\}$.
R is the quotient ring $\frac{\mathbb{Z}[x]}{x^n-1}$.
R_p is the quotient ring $\frac{\mathbb{Z}_p[x]}{x^n-1}$.
R_q is the quotient ring $\frac{\mathbb{Z}_q[x]}{x^n-1}$.
$\Phi_n = 1 + x + x^2 + \cdots + x^{n-1}$.
$\Phi_1 = x - 1$.
A polynomial is ternary if its coefficients are in $\{-1, 0, 1\}$.
$\mathcal{T}$ is the set of non-zero ternary polynomials of degree at most $n-1$.
$\mathcal{T}(d_1, d_2)$ is a subset of $\mathcal{T}$ consisting of polynomials that have exactly d_1 coefficients equal to 1 and d_2 coefficients equal to -1.

Multiplication of polynomials in 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 $\mathbf{f}(x)\mathbf{g}(x)$ 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 $\mathbf{f}(x)\mathbf{g}(x)$ in R_q.

2.2 The Ntru scheme

In this section, we describe the original Ntru scheme.
Keygen:

- Choose $\mathbf{f} \in \mathcal{T}(d, d+1)$ such that $\mathbf{f}$ is invertible in R_p and R_q.
- Let $\mathbf{f}_p$ be the inverse of $\mathbf{f}$ in R_p
- $\mathbf{f}_q$ be the inverse of $\mathbf{f}$ in R_q.
- Choose $\mathbf{g} \in \mathcal{T}(d, d)$.
- Let $\mathbf{h} = p\mathbf{g}\mathbf{f}_q \mod (q)$.
- Public key: $\mathbf{h}$.
- Private key: $(\mathbf{f}, \mathbf{g})$.

Encryption:

- Let $m \in R_p$ be a message.
- Choose $\mathbf{r} \in \mathcal{T}(d, d)$
- compute $\mathbf{c} = p\mathbf{r}\mathbf{h} + \mathbf{m} \mod q$.

Decryption:

- Compute $\mathbf{a} = \mathbf{cf} \mod q$.
- Center lift $\mathbf{a}$ to R and do a $\mod p$ computation
  $\mathbf{m} = \mathbf{f}_p\mathbf{a} \mod p$.

Definition 1. *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 $\mathscr{T}(d, d)$, the decryption will be successful and no information is revealed. Hence, the attacker has to choose a special $\mathbf{r}$ outside of the set $\mathscr{T}(d, d)$ which will fail the decryption so that he can get some information about the private key.

3 Our Attack

The general strategy is inspired by Ding's key reuse attack to LWE and ring LWE [7]. Under the assumption that both public and private keys are unchanged, an adversary can obtain some information about the private key by choosing a particular message and ephemeral key due to the fact that the result of decryption is accessible to the adversary. Our attack consists of two parts. The first part is to obtain an equivalent $\mathbf{g}$. Our strategy is to find the longest chain of consecutive nonzero coefficients in $\mathbf{g}$. This step can be done by attempting the first several coefficients of ephemeral keys from large values to small values and see if mismatch happens in decryption. Afterward, we will guess the coefficients that are next to the endpoint of such longest chain one by one until we get all the coefficients. We can verify our guess by assigning some values to the corresponding coefficients of ephemeral key and see if the decryption goes through. The second part of our attack is to find an equivalent key for $\mathbf{f}$. Since we have the equality $\mathbf{h} = p\mathbf{g}\mathbf{f}_q$ in R_q. We can find an equivalent $\mathbf{f}_q$ by figuring out the kernel of $\mathbf{g}$, which has a special form with high probability. Once we have an equivalent $\mathbf{f}_q$, it is easy to obtain equivalent $\mathbf{f}$ and $\mathbf{f}_p$.

3.1 Finding an Equivalent g

Finding the longest Chain We first assume that our message is equal to **0**, the case $\mathbf{m} \neq \mathbf{0}$ will be discussed later. In the decryption process, $\mathbf{a} := \mathbf{fc}$ in R_q, so $\mathbf{a} := \mathbf{f}(p\mathbf{hr}) = \mathbf{f}(p\mathbf{f}_q\mathbf{gr}) = p\mathbf{gr}$ in R_q. Hence, $\mathbf{a}$ can be expressed as a multiplication of matrices:

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

Assume that **g** has the longest chain of consecutive non-zero coefficients $g_{i+1}, \cdots, g_{i+k}$. It follows that the first k entries of $(i+k)^{th}$ column of this matrix

$$\begin{bmatrix} pg_0 & pg_1 & \cdots & pg_{n-1} \\ pg_{n-1} & pg_0 & \cdots & pg_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ pg_1 & pg_2 & \cdots & pg_0 \end{bmatrix}$$

are either $p, \cdots, p$ or $-p, \cdots, -p$.

$$\begin{bmatrix} pg_0 & \cdots & \pm p & \cdots & pg_{n-1} \\ pg_{n-1} & \cdots & \pm p & \cdots & pg_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_{i+k+1} & \cdots & \pm p & \cdots & \pm p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_1 & \cdots & pg_{i+k+1} & \cdots & pg_0 \end{bmatrix}$$

If we set our ephemeral key with coefficients $r_0 = r_1 = \cdots = r_k = \lceil \frac{q-1}{2pk} \rceil$, and $r_{k+1} = \cdots = r_{n-1} = 0$, we have that

$$\begin{bmatrix} \lceil \frac{q-1}{2pk} \rceil \cdots \lceil \frac{q-1}{2pk} \rceil \; 0 \cdots 0 \end{bmatrix} \begin{bmatrix} pg_0 & \cdots & \pm p & \cdots & pg_{n-1} \\ pg_{n-1} & \cdots & \pm p & \cdots & pg_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_{i+k+1} & \cdots & \pm p & \cdots & \pm p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_1 & \cdots & pg_{i+k+1} & \cdots & pg_0 \end{bmatrix}$$

It is clear that the $(i+k)^{th}$ position of the resultant vector is either larger than $\frac{q-1}{2}$ or less than $-\frac{q-1}{2}$ which goes outside the boundary of $\mathbb{Z}_q$. So it will cause additional modulus in the decryption process, and therefore leads to mismatch. However if the ephemeral key **r** has coefficients $r_0 = r_1 = \cdots = r_{k+1} = \lceil \frac{q-1}{2p(k+1)} \rceil$ and $r_{k+2} = \cdots = r_{n-1} =$

0, we have that

$$\left[\lceil\tfrac{q-1}{2p(k+1)}\rceil \cdots \lceil\tfrac{q-1}{2p(k+1)}\rceil \; 0 \cdots 0\right]\begin{bmatrix} pg_0 & \cdots & \pm p & \cdots & pg_{n-1} \\ pg_{n-1} & \cdots & \pm p & \cdots & pg_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_{i+k+1} & \cdots & \pm p & \cdots & \pm p \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ pg_1 & \cdots & pg_{i+k+1} & \cdots & pg_0 \end{bmatrix}$$

This time the $(i+k)^{th}$ position of the resultant vector is between $-\frac{q-1}{2}$ and $\frac{q-1}{2}$, so the decryption process should go through.
we will use the above idea to obtain the longest chain of consecutive nonzero coefficients of **g**. We just set the first j coefficients of **r** equal to $\lceil\frac{q-1}{2pj}\rceil$ and rest equal to 0, and see whether the decryption goes through or not. If decryption fails, then we set the first $j+1$ coefficients of **r** equal to $\lceil\frac{q-1}{2p(j+1)}\rceil$ and the rest equal to 0. We keep trying until decryption goes through. If the decryption succeeds when the first k coefficients of **r** are equal to $\lceil\frac{q-1}{2pk}\rceil$, we immediately know that **g** has the longest chain of $k-1$ consecutive 1's or $k-1$ consecutive -1's.

Guessing the remaining coefficients of g We may assume that the longest chain of consecutive nonzero coefficients of **g** appears in the beginning and they are equal to 1, i.e. $[g_0,\cdots,g_k,g_{k+1},\cdots,g_{n-1}] = [1,1,\cdots,1,g_{k+1},\cdots,g_{n-1}]$ since the difference is nothing but a shifting of coefficients and a positive or negative sign. We want to show that we can get the remaining coefficients of **g** but with some ambiguous case which can be solved in 3.1.3. This can be proved by mathematical induction.

It is clear that g_{k+1} is either 0 or -1. If we set $r_0 = r_1 = \cdots = r_k = \lceil\frac{q-1}{2p(k+1)}\rceil$, $r_{k+1} = -\lceil\frac{q-1}{2p(k+1)}\rceil$ and rest of the coefficients equal to 0, then in the case of $g_{k+1} = -1$, the decryption will fail, otherwise the decryption will pass. So, the decryption result will decide the value of g_{k+1}.

Assuming that we know the first $k+j$ coefficients of **g**, we want to show that we can obtain g_{k+j+1}. Let n be the number of nonzero coefficients in $\{g_0,\cdots,g_{k+j}\}$. For $i \in \{0,\cdots,k+j\}$, set $r_i = g_i * \lceil\frac{q-1}{2p(n+1)}\rceil$, and $r_{k+j+1} = \lceil\frac{q-1}{2p(n+1)}\rceil$ and rest equal to 0. If $g_{k+j+1} = 0$ or -1 then the decryption will pass through but if $g_{k+j+1} = 1$ then the decryption will fail. Whereas, if we set $r_i = g_i * \lceil\frac{q-1}{2p(n+1)}\rceil$ for $i \in \{0,\cdots,k+j\}$, and $r_{k+j+1} = -\lceil\frac{q-1}{2p(n+1)}\rceil$ and rest equal to 0. We have different decryption results this time. That is, if $g_{k+j+1} = 0$ or 1 then the decryption will pass, if $g_{k+j+1} = -1$, the decryption will fail.

Based on the above results, we attempt both the choices of **r**. If for both choices the decryption passes through then we immediately know that $g_{k+1+j} = 0$, whereas if one choice fails and the other passes then it tells us the exact value of g_{k+j+1}. However, if both the choices fail then we have an ambiguous case, but at least in this case, we know that g_{k+j+1} is not equal to 0.

Fortunately, the ambiguous case does not affect the process of finding the next coefficient of $\mathbf{g}$. For the convenience, we may simply assume that it is equal to 0 although it is indeed not. This is because no matter what value the ambiguous case g_j is, we can always set the inner product $r_0 g_0 + r_1 g_1 + \cdots + r_{j-1} g_{j-1}$ near the boundary of $\mathbb{Z}_q$ to test g_{j+1}. Therefore, we claim that we can obtain all the coefficients of $\mathbf{g}$ but with some ambiguous cases.

Completing the guess Now we finally complete the guess by assigning 1's and -1's to the positions wherever the ambiguous cases occur and see whether the decryption passes through or not. Suppose there are n ambiguous cases, then there are at most 2^n possibilities to check.

The case m ≠ 0 One may say that the attack can be easily prevented if we keep using 0 as our message all the time. However, due to [6], if we choose our message from $\mathcal{T}(d,d)$ with small d, the attack still works. Our experimental results give the same result.

3.2 Finding an Equivalent f

Suppose we obtain an equivalent key $\hat{\mathbf{g}}$ by going through 3.1. We claim that $\hat{\mathbf{g}}$ has enough information to construct an equivalent private key $\hat{\mathbf{f}}$. By nature of its construction, $\hat{\mathbf{g}}$ (if successfully generated) can only differ from $\mathbf{g}$ by a sign and a shifting of its coefficients. That is, if $\mathbf{g} = \sum_{i=0}^{n-1} a_i x^i \in \mathcal{T}(d,d)$ then for some integer m,

$$\hat{\mathbf{g}} = v \sum_{i=0}^{n-1} a_{(i+m) \mathrm{mod} n} x^i = v x^m \sum_{i=0}^{n-1} a_i x^i = v x^m \mathbf{g}$$

where v is simply 1 if the longest chain of nonzero coefficients in $\mathbf{g}$ is indeed 1's and is -1 if 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

$$\mathbf{h} = \mathbf{f}_q \mathbf{g} = \mathbf{f}_q \cdot v x^{n-m} \hat{\mathbf{g}}$$

Let us denote $\bar{\mathbf{f}_q} = v x^{n-m} \mathbf{f}_q$ so we may write $\mathbf{h} = \bar{\mathbf{f}_q} \hat{\mathbf{g}}$. Further let us write

$$\bar{\mathbf{f}} = \bar{\mathbf{f}_q}^{-1} = v x^m \mathbf{f}$$

where we view each polynomial as a member of R_q. We note that as multiplying by v is merely a potential change of sign and multiplying by x^m just shifting which coefficient belongs to which term, $\bar{\mathbf{f}}$ will be a ternary polynomial like $\mathbf{f}$.

It thus remains to find $\bar{\mathbf{f}_q}$ which results in $\bar{\mathbf{f}}$. As we already know $\hat{\mathbf{g}}$ and $\mathbf{h}$, we can turn this into solving

$$\mathbf{x}\hat{\mathbf{G}} = \mathbf{h} \tag{1}$$

Here $\mathbf{x}$ and $\mathbf{h}$ are vectors representing an unknown polynomial and $\mathbf{h}$ in R_q respectively. $\hat{\mathbf{G}}$ is the matrix corresponding to $\hat{\mathbf{g}}$ as detailed in section 2.1.1. As $\hat{\mathbf{g}}$ has a zero divisor, the same as $\mathbf{g}$. Thus $\hat{\mathbf{g}}$ will have a nontrivial nullspace N. So if $\mathbf{u}$ is a particular solution to (1), $(\mathbf{u}+N)$ where is the set of all solutions to (1). Hence $\bar{\mathbf{f}_q}$ belongs to $(\mathbf{u}+N)$. To find it one goes checking each element in $(\mathbf{u}+N)$ to see if it is invertible in R_q and importantly that its inverse is a ternary polynomial. As the rank of $\hat{\mathbf{G}}$, considered as a random matrix, has very high probability to be almost full, but can never be full, this set is small enough to search through as our computer experiments have verified. Having found $\bar{\mathbf{f}_q}$ one can then make an equivalent key.

3.3 Experimental Results

Our parameters are: $N = 61$, $p = 3$, $q = 2048$, $d = 20$. We chose our message $\mathbf{m} \in \mathcal{T}(3,3)$. The programming language we used is Magma

Attack ran	100
Ambiguous case	12
Nullity failure	2132
Total success	93

It can be seen from our experiments that the ambiguous case happened at a very low probability. We got most of the coefficients of $\mathbf{g}$ by doing 3.1.1 and 3.1.2. The main reason our attack fails is the nullity problem. In other words, the $\bar{\mathbf{g}}$ we found provides a nullity that has dimension greater than 1 or some other large nullity. Moreover, for each ambiguous case, we had to try 1 and -1 in that position. These attempts gave us different $\bar{\mathbf{g}}$, which corresponded to different $\mathbf{G}$ in (1). Hence the attempts for ambiguous case may contribute to nullity failure. Overall, Our attack came with a very high success rate.

Bibliography

[1] 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.

[2] 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.

[3] Ciprian Băetu, F. Betül Durak, Loïs Huguenin-Dumittan, Abdullah Talayhan, and Serge Vaudenay. Misuse attacks on post-quantum cryptosystems. Cryptology ePrint Archive, Report 2019/525, 2019. https://eprint.iacr.org/2019/525.

[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] 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.

[14] 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.

[15] Yue Qin, Chi Cheng, and Jintai Ding. A complete and optimized key mismatch attack on nist candidate newhope. *IACR Cryptology ePrint Archive*, 2019:435, 2019.

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

[17] Eric Rescorla. The transport layer security (tls) protocol version 1.2. 2018.

[18] 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.