석 사 학 위 논 문
Master's Thesis

Apon 등의 그룹키 교환 (PQCrypto 2019) 구체화 및 키 재사용 공격 취약성 연구

Instantiation of Apon et al.'s GKE (PQCrypto 2019) and its vulnerability by key-reuse attack

2020

홍 동 연 (洪 東 演 Hong, Dongyeon)

한 국 과 학 기 술 원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

Apon 등의 그룹키 교환 (PQCrypto 2019) 구체화 및 키 재사용 공격 취약성 연구

2020

홍 동 연

한 국 과 학 기 술 원

전산학부 (정보보호대학원)

Apon 등의 그룹키 교환 (PQCrypto 2019) 구체화 및 키 재사용 공격 취약성 연구

홍 동 연

위 논문은 한국과학기술원 석사학위논문으로
학위논문 심사위원회의 심사를 통과하였음

2019년 12월 16일

심사위원장 김 광 조 (인)

심 사 위 원 강 병 훈 (인)

심 사 위 원 이 주 영 (인)

Instantiation of Apon et al.'s GKE (PQCrypto 2019) and its vulnerability by key-reuse attack

Dongyeon Hong

Advisor: Kwangjo Kim

A dissertation submitted to the faculty of
Korea Advanced Institute of Science and Technology in
partial fulfillment of the requirements for the degree of
Master of Science in Computer Science (Information Security)

Daejeon, Korea
December 16, 2019

Approved by

Kwangjo Kim
Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics[1].

[1] Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MIS
20183669

홍동연. Apon 등의 그룹키 교환 (PQCrypto 2019) 구체화 및 키 재사용 공격 취약성 연구. 전산학부 (정보보호대학원) . 2020년. 36+iv 쪽. 지도 교수: 김광조. (영문 논문)
Dongyeon Hong. Instantiation of Apon et al.'s GKE (PQCrypto 2019) and its vulnerability by key-reuse attack. School of Computing (Graduate School of Information Security) . 2020. 36+iv pages. Advisor: Kwangjo Kim. (Text in English)

초 록

암호시스템은 안전하지 않은 채널에서 통신 상대방과 안전한 소통을 하기 위해 사용되며 암호시스템을 사용하기 전 두 사용자 혹은 그룹 내 동일한 비밀키의 교환이 우선 수행되어야 한다. 근래 두 명이 아닌 그룹 단위의 작업 환경이 많아지면서 그룹 내 비밀키 교환이 중요해지고 있다. 1994년 Burmester와 Desmedt가 그룹키 교환 프로토콜을 제안한 이후 현재까지 많은 연구가 진행되었다. 최근 PQCrypto에 Apon 등은 동일한 라운드의 격자 기반 그룹키 교환 프로토콜을 최초로 제안하였다. 그러나 Apon 등은 프로토콜의 키 조정 메커니즘을 구체적으로 설계하지 않고 일반적인 설계로 제안하여 실제 구현에 추가적인 연구가 필요하다. 본 논문에서는 Apon 등의 프로토콜에 NewHope키 조정 메커니즘을 도입하여 프로토콜 설계를 완성하고 이에 따른 정확성과 안전성을 살펴보도록 한다. 그 후 키 재사용 공격을 통해 키 조정 메커니즘 적용 시 주의를 기울여야 함을 제안하고자 한다.

핵 심 낱 말 키 교환 프로토콜, 격자 기반 암호, 양자 내성 암호, 키 재사용 공격

Abstract

A cryptosystem is used to communicate securely with the intended party on insecure channels. Before using a cryptographic system, two users or group members should perform a key exchange protocol, and then users or members can initiate a cryptographic system.

Recently, as the group-based working environment, not two people, increases, the key exchange within a group becomes important. Since Burmester and Desmedt proposed group key exchange protocols in 1994, numerous researches have been conducted. Apon *et al.* first presented the constant round group key exchange protocol based on the lattice problem. However Apon *et al.* proposed the key reconciliation mechanism of protocol as general design, rather than specifically designed it. In this paper, we instantiate the Apon *et al.*'s protocol using the key reconciliation mechanism of NewHope, which is a two-party key exchange protocol, and analyze its correctness and security. Also, we suggest that we should be careful when applying the key reconciliation mechanism through a key-reuse attack.

Keywords Key exchange protocol, lattice-based cryptography, post-quantum cryptography, key-reuse attack

Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1 Motivation

In everyday life, we communicate with or send files to other people through various messengers or mail. Often, classified files are also exchanged through same applications. According to [41], we can recover some parts of chattings, photos, and files sent by the KakaoTalk application from our Windows computers and Android phones. Also, an adversary can easily eavesdrop any communications between two or more users over an insecure channel. To prevent this, we use symmetric key cryptography or public-key cryptography. Before using symmetric key cryptography protocols, the same secret key have to be shared with the intended two users. Two-party key exchange protocols have been proposed to share such a secret key.

However, as increasing the number of group-based applications, not between two users, a group key exchange (GKE) protocol has become important. As we know, by repeating a two-party key exchange protocol between two users in a group, all group participants can share a common secret key (group key). However, this repeating method is inefficient as the number of users increases. Since [16, 17] presented GKE protocols, numerous researches have been conducted to date. [13, 14, 15, 28, 27, 29, 18, 1, 2, 9] proposed a GKE protocol based on the discrete logarithm problem (DLP), which is challenging to solve mathematically. (Mathematically hardness refers that given problem is computationally infeasible to solve with the current computer, and we call such a problem as a hard problem or problem is hard)

However, after Simon's algorithm [39] in 1994, [38, 25] showed that the mathematically hard problem such as DLP could be solved in polynomial time using a quantum computer, which means that if quantum computers become commercially available, then cryptosystems we use currently will break down. Recently, Google presented the Sycamore, 53-qubit quantum computer, and showed that Sycamore resolves a problem that takes 10,000 years to solve in 200 seconds. [6] This shows the rapid development of quantum computers, and to prevent such threats post-quantum cryptography is ongoing worldwide.

National Institute of Standards and Technology (NIST) is currently working on the standardization of entire areas such as encryption, signing, and key exchange protocols. Including lattice-based key exchange protocols [3, 10, 11, 22, 12], many protocols have been submitted.

However, [3, 10, 11, 22, 12] are two-party key exchange protocols, not group based one. In 2019, Apon *et al.* firstly proposed the constant-round lattice-based group key exchange protocol with the generic design of the key reconciliation mechanism.

Therefore, in this paper, we instantiate the Apon *et al.*'s protocol using the key reconciliation

mechanism of NewHope and examine its correctness and security. After that, we suggest that we should be careful when applying the key reconciliation mechanism through a key-reuse attack.

1.2 Organization

The rest of this thesis is organized as follows: Chapter 2 describes notations and definitions as preliminaries. The related work, which consists of group key exchange, lattice-based key exchange, and key-reuse attack, is described in Chapter 3. In Chapter 4, we describe our instantiation of ADGK19 protocol. The completion of correctness and security proof is presented in Chapter 5. Vulnerability by a key-reuse attack is described in Chapter 6. Finally, the conclusion and future work are discussed in Chapter 7.

Chapter 2. Preliminaries

2.1 Notations

Let $\mathbb{Z}$ be the ring of integers and $\mathbb{Z}_q$ be a quotient ring $\mathbb{Z}/q\mathbb{Z}$ for $q \geq 1$. Define the round function for $x \in \mathbb{R}$, $\lfloor x \rceil = \lfloor x + \frac{1}{2} \rfloor \in \mathbb{Z}$. For a even (resp. odd) integer m, define $\alpha' = \alpha \bmod^{\pm} m$ be the unique element α' in the range $\frac{m}{2} < \alpha' \leq \frac{m}{2}$ (resp. $\frac{m-1}{2} \leq \alpha' \leq \frac{m-1}{2}$). Let $[N] = \{0, 1, \cdots, N-1\}$. We use $log(x), exp(x)$ to denote $log_2(x)$ and e^x for $x \in \mathbb{R}$, respectively. Write (column) vectors as bold characters $\mathbf{a} = (a_0, a_1, \cdots, a_{m-1})^T$, where $\mathbf{a}^T$ denotes the transpose of the vector, and matrices as capital bold characters $\mathbf{A}$. For a vector $\mathbf{a} \in \mathbb{R}^m$, we define the l_1-norm as $\|\mathbf{a}\|_1 = \sum_{i=0}^{m-1} |a_i|$ and the l_2-norm as $\|\mathbf{a}\|_2 = \left(\sum_{i=0}^{m-1} |a_i|^2\right)^{1/2}$. In this paper, $\|\cdot\|$ is l_2-norm.

If S is a set, $s_1, s_2, \cdots, s_k \longleftarrow S$ denotes that uniformly sampling each s_i from S and if χ is a distribution over S, then $s_1, s_2, \cdots, s_{k'} \longleftarrow \chi$ denotes that independently sampling each $s_i \in S$ according to χ. In case $\mathcal{A}$ is a probabilistic algorithm, $a_1, a_2, \cdots, a_{k''} \longleftarrow \mathcal{A}$ denotes that the output of $\mathcal{A}$ is assigned to each a_i randomly and independently. Let $\chi(E)$ be a probability that an event E occurs under distribution χ and let $\mathrm{Supp}(\chi) = \{s \in S \mid \chi(s) \neq 0\}$.

Define a ring of integer polynomials $R = \mathbb{Z}[x]/(f(x))$ with modulo $f(x)$ where $f(x)$ is an irreducible polynomial. In this paper, we set $f(x) = x^n + 1$, where n is a power of 2. For any positive integer q, define a quotient ring of integer polynomials $R_q = \mathbb{Z}_q[x]/(f(x)) = R/qR$ where every coefficient is reduced modulo q. We restrict to the case where q is prime with $q = 1 \mod 2n$. Also we define the ring of integer polynomials $\bar{R} = \mathbb{Z}[x]/(x^4 + 1)$. Denote for $v = v_0 + v_1 x^1 + v_2 x^2 \cdots + v_{n-1} x^{n-1}$, $v[i]$ as an i-th coefficient of v. We can define the coefficient embedding mapping σ from R to $\mathbb{R}^n$ by $\sigma(v = \sum_{i=0}^{n-1} v_i x^i) = (v[0], v[1], v[2], \cdots, v[n-1])$. Also we can define the splitting mapping $\phi : R \longrightarrow \bar{R}^{\frac{n}{4}}$ by $\phi(v) = (\bar{v}_0, \bar{v}_1, \cdots, \bar{v}_{\frac{n}{4}-1})$ where $\bar{v}_i = v[i] + v[i + \frac{n}{4}]\, x + v[i + 2 \cdot \frac{n}{4}]\, x^2 + v[i + 3 \cdot \frac{n}{4}]\, x^3 \in \bar{R}$ for any $v \in R$. When we embed $\bar{v}_i$ to $\mathbf{v}_i = (v[i],\, v[i + \frac{n}{4}],\, v[i + 2 \cdot \frac{n}{4}],\, v[i + 3 \cdot \frac{n}{4}])^T$, $\mathbf{v}_i$ is called an *i-th split vector of* v.

We let λ denote a *computational security parameter* and ρ a *statistical security parameter*.

2.2 Definitions

2.2.1 Error Distributions

In this section, we deal with error distributions which are used commonly in lattice based cryptography.

Table 2.1: Notations and Variables

Variables	Description
N	number of parites
P_i	ith party of a protocol
n	degree of polynomial ring
$R = \mathbb{Z}[x]/(x^n + 1)$	the ring of integers with dimension n
$R_q = R/qR$	quotient ring of R for a modulus integer q
$a[j]$	j-th coefficient of a polynomial $a \in R$
$\mathbf{a}_{k,l}$	l-th split vector of $\mathbf{a}_k$ in k-th session i.e. $\mathbf{a}_{k,l} = (a_k[l], a_k[l\frac{n}{4}], a_k[l + 2 \cdot \frac{n}{4}], a_k[l + 3 \cdot \frac{n}{4}])$
χ_σ	discrete Gaussian distribution with variance σ^2
$a \longleftarrow S$	uniformly and independently sampling from a set S
$a \longleftarrow \mathsf{A}$	assigning output from a algorithm or function A
ρ	statistical security parameter
λ	computational security parameter
$H()$	cryptographic hash function

Discrete Gaussian Distribution

For $\sigma \in \mathbb{R}$, denote $\rho_\sigma(\mathbf{x}) = exp\left(-\frac{\pi \cdot \|\mathbf{x}\|^2}{\sigma^2}\right)$ as the Gaussian function scaled by σ where $\mathbf{x} \in \mathbb{R}^m$ and let $\rho_\sigma(\mathbb{Z}^m) = \sum_{\mathbf{x} \in \mathbb{Z}^m} \rho_\sigma(\mathbf{x})$. Define the *m-dimensional discrete Gaussian distribution* $D_{\mathbb{Z}^m,\sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\mathbb{Z}^m)}$ for $\mathbf{x} \in \mathbb{Z}^m$. To sample a polynomial of degree n, (1) take $m = 1$ and sample each coefficient n times from $D_{\mathbb{Z},\sigma}(x)$ or (2) take $m = n$ and sample coefficient vectors from $D_{\mathbb{Z}^n,}(\mathbf{x})$. We write $D^m_{\mathbb{Z},\sigma}$ to denote sampling m times from $D_{\mathbb{Z},\sigma}$.s We can restrict the sample domain to $\mathbb{Z}_q$, in other words, i-th coordinate x_i of the input $\mathbf{x}$ is in the range $-\frac{q}{2} < x_i \leq \frac{q}{2}$. Figure 2.1 shows a 2-dimensional discrete Gaussian distribution with standard deviation $\sigma = 2\sqrt{8}$

Remark that the ring-Learning with Errors (ring-LWE) problem is still hard if the secret $s \in R_q$ is sampled from an error distribution instead of sampled from a uniform distribution of R_q. [5, 32, 33]

But the discrete Gaussian distribution has some issues on its implementation and efficiency.

Centered Binomial Distribution

To resolve such problem of the discrete Gaussian distribution, [3] used a centered binomial distribution which is more simpler design and efficient than Gaussian distribution. For $\eta \in \mathbb{Z}$, a *centered binomial distribution* ψ_η is defined as

$$\psi_\eta = \sum_{i=0}^{\eta-1} (b_i - b'_i)$$

where b_i, $b'_i \longleftarrow \{0, 1\}$.

[3] proved that if an error distribution replaced by centered binomial distribution instead of Gaussian distribution, still ensure the security level. (See theorem 1

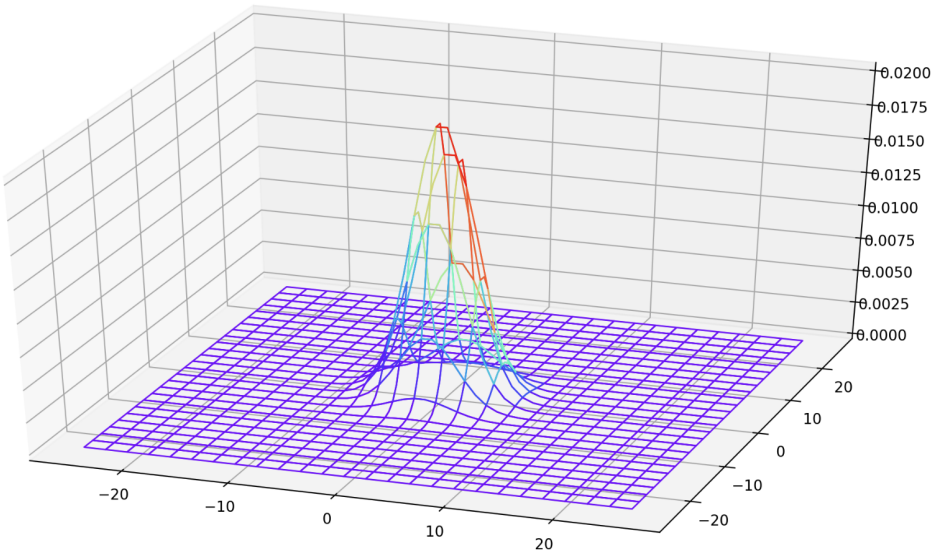

Figure 2.1: Discrete Gaussian distribution with $\sigma = 2\sqrt{8}$

Theorem 1 (restated [3])**.** Let ξ be the rounded Gaussian distribution of parameter $\sigma = \sqrt{8}$, that is, the distribution of $\lfloor \sqrt{8} \cdot x \rceil$ where x follows the standard normal distribution. Let $\mathscr{P}$ be the idealized version of [3], where the distribution $\psi_1 6$ is replaced by ξ. If an (unbounded) algorithm, given as input the transcript of an instance of [3] succeeds in recovering the pre-hash key ν with probability p, then it would also succeed against $\mathscr{P}$ with probability q at least

$$q \geq p^{9/8}/26.$$

To best our knowledge, there are no known attacks using difference in error distributions.

2.2.2 Rènyi Divergence

Bai showed that the *Rènyi divergence* can be used lattice-based cryptography when analyzing closeness between two distributions. [8] The Rènyi divergence is defined as follow: For $k > 1$, and two distributions P and Q,

$$RD_k(P\|Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^k}{Q(x)^{k-1}} \right)^{\frac{1}{k-1}}$$

In this paper, we set $k = 2$.

Proposition 1 ([40])**.** For two discrete distributions P and Q with $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$, let $E \subseteq \mathrm{Supp}(Q)$ be an arbitrary event. Then we have

$$Q(E) \geq P(E)^2 / RD_2(P\|Q).$$

This **Proposition 1** indicates that as long as $RD_2(P\|Q)$ is bounded by some polynomial, then any event $E \subseteq \mathrm{Supp}(Q)$ that occurs with negligible probability under Q also occurs with negligible probability under P.

Theorem 2. Fix $m, q, \lambda \in \mathbb{Z}$, a bound $\beta_{\text{Rènyi}}$, and σ with $\beta_{\text{Rènyi}} < \sigma < q$. Let $e \in \mathbb{Z}$ be such that $|e| \leq \beta_{\text{Rènyi}}$. Then

$$RD_2\left(\left(e + D_{\mathbb{Z}_q,\sigma}\right)^m \| D^m_{\mathbb{Z}_q,\sigma}\right) \leq exp\left(2\pi m \left(\beta_{\text{Rènyi}}/\sigma\right)^2\right).$$

2.2.3 Voronoi Cell

For a given lattice L, *Voronoi cell* is $V(L) = \{\mathbf{x} \in \mathbb{R}^m \mid \|x\| < \|x - v\| \text{ for all } v \in L\backslash\{\mathbf{0}\}\}$ which indicates that a set of vectors closer to origin than any other lattice point. We can define Voronoi cell centered at other lattice point not only the origin. Figure 2.2 shows Voronoi cell in 2-dimension centered at the origin, $(0,1)$, $(1,0)$, $(1,1)$, and $(1/2, 1/2)$ from [3]. In figure 2.2, grey line denote a border of each Voronoi cell.

Let H_v be the set of all vectors closer to origin that to $v \in L$. Then we can define a Voronoi relevant vector as follow: VR is *Voronoi relevant* if $V(L) = \cap_{v \in VR} H_v$ and VR is minimal. We call a vector in VR a *Voronoi relevant vecotr*.

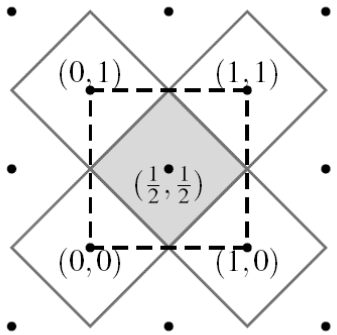

Figure 2.2: Voronoi cell in 2 dimension [3]

2.2.4 Key Transport and Exchange

Key transport is a key-sharing protocol in which one of two users or a representative user in a group selects a session key and sends it to another user. The representative party has more authority or power than any other user, and hence the security of key transport entirely depends on the representative party. Key transport can be used in a centralized topology or in an environment where trust is established, and national or certification organizations play the role of representative users. Otherwise, selecting a representative user may be another issue. However, it is beyond the scope of this paper and is not covered here.

Key transport may fall into a wrong cycle. Typically, we think about how to prevent eavesdropping when transmitting session keys. Assuming we send the encrypted session key k, we need another session

key k' to encrypt and decrypt. Then we need another session k'' to transmit k'. As mentioned before, we fall into a vicious circle. To resolve a wrong cycle, hybrid cryptography was designed. However, this hybrid method is not efficient in an environment such that the session key is changed frequently.

Unlike key transport, *key exchange* occurs in an environment where there is a lack of trust between users. In other words, there is any representative user, and all users have almost equal authority. Key exchange can be viewed as adding a step in key transport. Rather than the representative user sending the session key, each user sends a subkey to all other users and creates a session key by combining the received subkeys. At this time, we can divide key exchange into an implicit and explicit key exchange, depending on whether or not who has the session key. In implicit key exchange, we believe that users who not included in a group cannot compute session key, and explicit key exchange is to execute an authentication process to verify that the intended user has the session key.

Also, the properties of the session key are as follows: (1) *Key integrity* means that the generated session key has not been modified by the adversary and is created using the intended subkey of the intended user. (2) Each session has a *key freshness* that generates a new key, a key that has never been used before, (3) *key independence* that ensures that the generated session key is unrelated, and (4) a *key contributiveness* that says each user contributes to the session key generation.

2.3 Lattice-based Mathematical Hard Problems

2.3.1 Closest Vector Problem

Closest Vector Problem (CVP) is one of mathematical hard problem based on lattice and defined as follow: Given a vector $w \in \mathbb{R}^m$ that is not in a lattice L, find a nonzero vector $v \in L$ that is closest to w. Babai presented the algorithm to solve CVP in 1985. [7]

Theorem 3 (Babai's algorithm [7])**.** Let $L \subset \mathbb{R}^n$ be a lattice with basis $\{\boldsymbol{v_1}, \boldsymbol{v_2}, \cdots, \boldsymbol{v_n}\}$ and let $w \in \mathbb{R}^n$ be an arbitrary vector. If the vectors in the basis are sufficiently orthogonal to one another, then the following algorithm solves CVP.

Algorithm 1: Babai's algorithm

1. Express $w = t_1\boldsymbol{v_1} + t_2\boldsymbol{v_2} + \cdots t_n\boldsymbol{v_n}$ where $t_i \in \mathbb{R}$
2. Set $a_i = \lfloor t_i \rceil$ for $i = 1, 2, \cdots, n$
3. Return the vector $\boldsymbol{v} = a_1\boldsymbol{v_1} + a_2\boldsymbol{v_2} + \cdots a_n\boldsymbol{v_n}$

From now, introduce definitions about mathematical hard problem based on lattice described in [35].

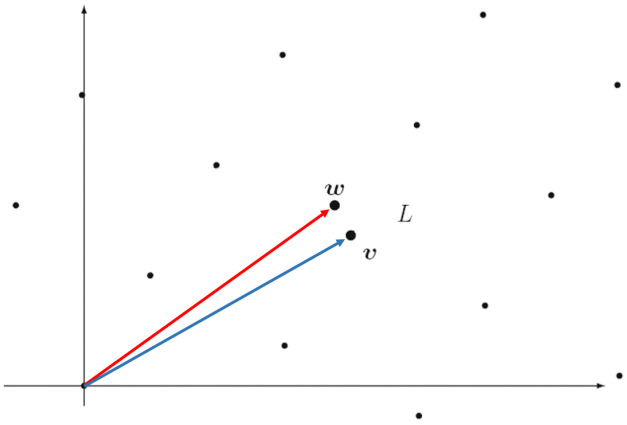

Figure 2.3: Closest vector problem [26]

2.3.2 Learning With Errors

Definition 1 (LWE distribution)**.** For a vector $\mathbf{s} \in \mathbb{Z}_q^n$ called the secret, *the LWE distribution* $A_{s,\chi}$ over $\mathbb{Z}_q^n \times Z_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \longleftarrow \chi$, and outputting $(\mathbf{a}, b =< \mathbf{s}, \mathbf{a} > + e \mod q)$.

Definition 2 (Search LWE problem)**.** *Search-LWE problem* is parameterized by (n, q, χ, l) and defined as follow: given l independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ from $A_{s,\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ but fixed for all samples, find the secret $\mathbf{s}$.

Definition 3 (Decision-LWE problem)**.** *Decision-LWE problem* is parameterized by (n, q, χ, l) and defined as follow: given l independent samples $(\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where every sample is distributed according to either: (1) $A_{s,\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ but fixed for all samples, or (2) the uniform distribution, distinguish which is the cased with non-negligible advantage.

Theorem 4 ([36, 37])**.** For any $m = poly(n)$, any modulus $q \leq 2^{poly(n)}$, and any (discretized) Gaussian error distribution χ of parameter $\alpha q \geq 2\sqrt{n}$ where $0 < \alpha < 1$, solving the decision-LWE$_{n,q,\chi,m}$ problem is at least as hard as *quantumly* solving GapSVP$_\gamma$ and SIVP$_\gamma$ on arbitrary n-dimensional lattices, for some $\gamma = O(n/\alpha)$.

2.3.3 Ring-Learning With Errors

Similar to LWE case, we can define ring version of LWE but over R_q

Definition 4 (Ring-LWE distribution)**.** For a $s \in R_q$ called the secret, the *ring-LWE distribution* $A'_{s,\chi}$ over $R_q \times R_q$ is sampled by choosing $a \in R_q$ uniformly at random, choosing $e \longleftarrow \chi$, and outputting $(a, b = a \cdot s + e \mod q)$.

Definition 5 (Search-ring-LWE problem)**.** *Search-ring-LWE problem* is parameterized by (n, q, χ, l) and defined as follow: given l independent samples $(a_i, b_i) \in R_q \times R_q$ from $A'_{s,\chi}$ for a uniformly random $s \in R_q$ but fixed for all samples, find the secret s.

Definition 6 (Decision-ring-LWE problem)**.** *Decision-ring-LWE problem* is parameterized by (n, q, χ, l) and defined as follow: given l independent samples $(a_i, b_i) \in R_q \times R_q$ where every sample is distributed

according to either: (1) $A'_{s,\chi}$ for a uniformly random $s \in R_q$ but fixed for all samples, or (2) the uniform distribution, distinguish which is the cased with non-negligible advantage.

Theorem 5 ([32])**.** For any $m = poly(n)$, cyclotomic ring R of degree n (over $\mathbb{Z}$), and appropriate choices of modulus q and error distribution χ of error rate $\alpha < 1$, solving the R-LWE$_{q,\chi,m}$ problem is at least as hard as *quantumly* solving the SVP$_\gamma$ problem on arbitrary ideal lattices in R, for some $\gamma = poly(n)/\alpha$.

Module-LWE problem is introduced by Langlois *et al.* [30] in 2015. Module-LWE is also a quantum-resistant mathematical hard problem against the quantum adversary

2.3.4 Module-Learning With Errors

Definition 7 (Module-LWE distribution)**.** For a $\mathbf{s} \in R_q^k$ called the secret, the *module-LWE distribution* $A''_{\mathbf{s},\chi}$ over $R_q^k \times R_q$ is sampled by choosing $\mathbf{a} \in R_q^k$ uniformly at random, choosing $e \longleftarrow \chi$, and outputting $(\mathbf{a}, b =< \mathbf{s}, \mathbf{a} > + e \mod q)$.

Definition 8 (Search-module-LWE problem)**.** *Search-module-LWE problem* is parameterized by (n, k, q, χ, l) and defined as follow: given l independent samples $(\mathbf{a}_i, b_i) \in R_q^k \times R_q$ from $A''_{\mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in R_q^k$ but fixed for all samples, find the secret $\mathbf{s}$.

Definition 9 (Decision-module-LWE problem)**.** *Decision-ring-LWE problem* is parameterized by (n, k, q, χ, l) and defined as follow: given l independent samples $(\mathbf{a}_i, b_i) \in R_q^k \times R_q$ where every sample is distributed according to either: (1) $A''_{\mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in R_q^k$ but fixed for all samples, or (2) the uniform distribution, distinguish which is the cased with non-negligible advantage.

Chapter 3. Related Work

3.1 Group Key Exchange

Since [16] proposed the first GKE, there are numerous variants are shown up. [14, 15, 28, 27, 29, 18, 1, 9]. **Algorithm** 2 describes group key exchange protocol of [16, 17] which is basic form of the group key exchange protocl.

Algorithm 2: BD94($P[0, 1, \cdots, N-1]$)

(Round 1) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. selects uniform $r_i \in \mathbb{Z}_q$ and broadcasts $z_i = g^{r_i}$

(Round 2) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. broadcasts $X_i = (z_{i+1}/z_{i-1})^{r_i}$

(Key Computation) For each party P_i,

1. computes the group key sk_i,

$$\begin{aligned} sk_i &= (z_{i-1})^{N \cdot r_i} \cdot X_i^{N-1} \cdot X_{i+1}^{N-2} \cdot \cdots \cdot X_{i-2} \\ &= g^{r_0 r_1 + r_1 r_2 + \cdots + r_{N-1} r_0} \end{aligned}$$

However, almost group key exchange protocols are based on DLP which could be broken when the quantum-computer is commercialized. We describe protocols which are resistant to quantum computer attacks.

3.2 Lattice-based Key Exchange

3.2.1 NIST Standard Candidates

As quantum computer becomes realistic in the near future, National Institute of Standards and Technology (NIST) has been selecting standard post-quantum cryptographic algorithms like key exchange, encryption, and signature schemes.

Frodo

Bos *et al.* proposed the two-party key exchange based on LWE assumption and we call this protocol Frodo [10]. Bos *et al.* chose four distributions which are instantiation of inversion sampling and closely approximate the rounded continuous Gaussian distribution as the error distribution. To construct the

Table 3.1: Algorithms of liboqs

Primitive		Protocol
Lattice-based	LWE	Frodo
	Ring-LWE	BCNS
		NewHope
		MSrln
	Module-LWE	Kyber
	NTRU	
Supersingular Elliptic Curve	SIDH	IQC Reference
		MSR SIDH
Code-based	Error-correcting codes	McBits

protocol of [10], some functions are needed and defined as follows: let B be the number of bits that from one coefficient in $\mathbb{Z}_q$ such that $B < log(q) - 1$ and $\bar{B} = log(q) - B$.

For any $v \in \mathbb{Z}_q$, the rounding function $\lfloor \cdot \rceil_{2^B} : \mathbb{Z}_q \longrightarrow [0, 2^B)$ is defined by $\lfloor v \rceil_{2^B} = \lfloor 2^{-\bar{B}} \cdot v \rceil \mod 2^B \rceil$ and cross-rounding function $\langle \cdot \rangle_{2^B} : \mathbb{Z}_q \longrightarrow \mathbb{Z}_2$ is defined by $\langle v \rangle_{2^B} = \lfloor 2^{-\bar{B}+1} \cdot v \rfloor \mod 2$. Note that the rounding function outputs the B most significant bits of $(v + 2^{\bar{B}-1} \mod q)$ which partitioning $\mathbb{Z}_q$ into 2^B intervals of integers and the cross-rounding function divides $\mathbb{Z}_q$ into two subsets according to their $(B+1)$-th most significant bit.

Using two functions, $\lfloor \cdot \rceil_{2^B}$ and $\langle \cdot \rangle_{2^B}$, rec function $\mathsf{rec} : \mathbb{Z}_q \times \mathbb{Z}_2 \longrightarrow [0, 2^B)$ is defined by $\mathsf{rec}(w, b) = \lfloor v \rceil_{2^B}$ where v is the closest element to w such that $\langle v \rangle_{2^B} = b$ for all $w \in \mathbb{Z}_q$ and $b \in \mathbb{Z}_2$.

Protocol 1 describes the key exchange protocol of [10] with parameters $n = 752, q = 2^{15}, B = 4$

Protocol 1: Frodo

Alice		**Bob**
$\mathsf{seed}_{\mathbf{A}} \longleftarrow U(\{0,1\}^\lambda)$		
$\mathbf{A} \longleftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$		
$\mathbf{S}, \mathbf{E} \longleftarrow \chi(\mathbb{Z}_q^{n \times \overline{n}})$		
$\mathbf{B} = \mathbf{AS} + \mathbf{E}$	$\xrightarrow{\mathsf{seed}_{\mathbf{A}}, \mathbf{B} \in \{0,1\}^\lambda \times \mathbb{Z}_q^{n \times \overline{n}}}$	$\mathbf{A} \longleftarrow \mathsf{Gen}(\mathsf{seed}_{\mathbf{A}})$
		$\mathbf{S}', \mathbf{E}' \longleftarrow \chi(\mathbb{Z}_q^{n \times \overline{n}})$
		$\mathbf{B}' = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
		$\mathbf{C} \longleftarrow \langle \mathbf{V} \rangle_{2^B}$
	$\xleftarrow{\mathbf{B}',\ \mathbf{C} \in \mathbb{Z}_q^{\overline{m} \times n} \times \mathbb{Z}_2^{\overline{m} \times \overline{n}}}$	
$sk' \leftarrow \mathsf{rec}(\mathbf{B}'\mathbf{S}, \mathbf{C})$		$sk \leftarrow \lfloor \mathbf{V} \rceil_{2^B}$

where $\bar{n}, \bar{m} \in \mathbb{Z}$ are protocol specific parameters.

NewHope

Alkim *et al.* proposed the two-party key exchange protocol based on RLWE assumption. [3] [3] chose the centered binomial distribution ψ_η rather than the discrete Gaussian distribution for efficient implementation. To construct the protocol of [3], two functions, $\mathsf{HelpRec}$ and Rec, are needed to be defined. Firstly, let $\mathsf{CVP}_{\tilde{D}_4} : \mathbb{R}^4 \longrightarrow \mathbb{Z}_4^4$ be the function outputting an integer vector $\boldsymbol{z} \in \mathbb{Z}^4$ such that $\boldsymbol{Bz}$ is a closest vector of input $\boldsymbol{x} \in \mathbb{R}^4$ (i.e. $\boldsymbol{x - Bz} \in \mathcal{V}$) where $\boldsymbol{B} = [\boldsymbol{e_1}\ \boldsymbol{e_2}\ \boldsymbol{e_3}\ \boldsymbol{g}]$ is a basis lattice $\tilde{D}_4$, $\boldsymbol{e_i}$ is the standard basis vectors of $\mathbb{Z}^4$ and $\boldsymbol{g} = (1/2,\ 1/2,\ 1/2,\ 1/2)^T$. For $w \in \mathbb{R}^4$ and $b \longleftarrow \{0,1\}$ the $\mathsf{HelpRec}$ function is defined as

$$\mathsf{HelpRec}(\boldsymbol{x}; b) = \mathsf{CVP}_{\tilde{D}_4}\left(\frac{2^r}{q}(\boldsymbol{x} + b\boldsymbol{g})\right) \mod 2^r,$$

where r is the number of discretization of Voronoi cell $\mathcal{V}$. Output of $\mathsf{HelpRec}$ is called a reconciliation vector or signal vector.

For $\boldsymbol{x} \in \mathbb{R}^4$ and a reconciliation vector $\boldsymbol{r} \in \mathbb{Z}_4^4$, the corresponding function

$$\mathsf{Rec}(\boldsymbol{x},\ \boldsymbol{r}) = \mathsf{Decode}\left(\frac{1}{q}\boldsymbol{x} - \frac{1}{2^r}\boldsymbol{Br}\right),$$

where Decode is the function outputting a bit k such that $k\boldsymbol{g}$ is a closest vector to $\boldsymbol{x} + \mathbb{Z}^4$.

Protocol 2 describes the key exchange protocol of [3] with parameters $n = 1024, q = 12289$ and $\eta = 16$

Protocol 2: NewHope

Alice		**Bob**
$\mathsf{seed} \longleftarrow \{0,1\}^{256}$		
$a \longleftarrow \mathsf{Parse}\,(\mathsf{SHAKE\text{-}128}(\mathsf{seed}))$		
$s, e, \longleftarrow \psi_{16}^n$		$s', e', e'' \longleftarrow \psi_{16}^n$
	$\xrightarrow{(b, \mathsf{seed})}$	$a \longleftarrow \mathsf{Parse}(\mathsf{SHAKE\text{-}128}(\mathsf{seed}))$
		$u = as' + e'$
		$v = bs' + e''$
	$\xleftarrow{(u, r)}$	$r \longleftarrow \mathsf{HelpRec}(v)$
$v' = us$		
$\nu' \longleftarrow \mathsf{Rec}(v', r)$		$\nu \longleftarrow \mathsf{Rec}(v, r)$
$sk' \longleftarrow \mathsf{SHA3\text{-}256}(\nu')$		$sk \longleftarrow \mathsf{SHA3\text{-}256}(\nu)$

BCNS

Bos *et al.* implemented the two-party key exchange protocol of [34]. Bos *et al.* chose the discrete Gaussian distribution as an error distribution. To complete the protocol of [34, 12], we define some

functions, the modular rounding function, cross-round function, dbl and rec, as follows: the modular rounding function $\lfloor \cdot \rceil_{q,2} : \mathbb{Z} \longrightarrow \mathbb{Z}$ is defined by $\lfloor x \rceil_{q,2} = \lfloor \frac{2}{q} x \rceil_{q,2} \mod 2$ for all $x \in \mathbb{Z}$ and cross-rounding function $\langle \cdot \rangle_{q,2} : \mathbb{Z} \longrightarrow \mathbb{Z}$ is defined by $\langle x \rangle_{q,2} = \lfloor \frac{4}{q} \cdot x \rfloor \mod 2$.

Define $\mathsf{dbl} : \mathbb{Z}_q \longrightarrow \mathbb{Z}_{2q}$ as $\mathsf{dbl}(x) = 2x - e$ where e is sampled from $-1, 0, 1$ with probabilities $p_{-1} = p_1 = \frac{1}{4}$ and $p_0 = \frac{1}{2}$. Let $I_0 = \{0, 1, \cdots, \lfloor \frac{q}{2} \rceil - 1\}$ and $I_1 = \{-\lfloor \frac{q}{2} \rceil, \cdots, -1\}$ and $E = [-\frac{q}{4}, \frac{q}{4})$. Then the reconciliation function $\mathsf{rec} : \mathbb{Z}_{2q} \times \mathbb{Z}_2 \longrightarrow \mathbb{Z}_2$ is defined by

$$\mathsf{rec}(w, b) = \begin{cases} 0 & \text{if } w \in I_b + E \mod 2q, \\ 1 & \text{otherwise .} \end{cases}$$

Protocol 3 describes the key exchange protocol of [12] with parameters $n = 1024, q = 2^{32} - 1, \sigma = 8/\sqrt{2\pi}$

Protocol 3: BCNS

Alice		**Bob**
$s, e \longleftarrow \chi$		$s', e' \longleftarrow \chi$
$b = as + e$	$\xrightarrow{\quad b \quad}$	$b' = as' + e'$
		$e'' \longleftarrow \chi$
		$v = bs' + e''$
		$\overline{v} \longleftarrow \mathsf{dbl}(v) \in \mathcal{R}_{2q}$
	$\xleftarrow{\quad b', c \quad}$	$c \longleftarrow \langle \overline{v} \rangle_{2q,2} \in \{0,1\}^n$
$sk' \longleftarrow \mathsf{rec}(2b's, c) \in \{0,1\}^n$		$sk \longleftarrow \lfloor \overline{v} \rceil_{2q,2} \in \{0,1\}^n$

Kyber

Bos *et al.* proposed the two-party key exchange protocol based on module-LWE assumption. Same with [3], they chose the centered binomial distribution to sample secret and error values. To construct the protocol of [11], some functions are needed and defined as follows: let $d < \lceil log_2(q) \rceil$. The Compress function $\mathsf{Compress}_q : \mathbb{Z}_q \times \mathbb{Z} \longrightarrow [2^d]$ is defined by $\mathsf{Compress}_q(x, d) = \lfloor \frac{2^d}{q} \cdot x \rceil \mod 2^d$ and the Decompress function $\mathsf{Decompress}_q : \mathbb{Z}_q \times \mathbb{Z} \longrightarrow \mathbb{Z}$ is defined by $\mathsf{Decompress}_q(x, b) = \lfloor \frac{q}{2^d} \cdot x \rceil$.

Using two functions,Compress and Decompress, **Algorithm** 3 and 4 describe the public key encryption scheme of **Kyber**. **Protocol 4** describes the protocol of [11].

Algorithm 3: Kyber.Enc $(pk = (\mathbf{t}, \rho), m \in \mathcal{M})$

$r \longleftarrow \{256\}^2 56$
$t \longleftarrow \mathsf{Decompress}_q(\mathbf{t}, d_t)$
$\mathbf{A} \longleftarrow Sam(\rho) \in R_q^{k \times k}$
$(\mathbf{r}, \mathbf{e}_1, e_2) \longleftarrow Sam(r)$
$\mathbf{u} \longleftarrow \mathsf{Compress}_q(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1, d_u)$
$v \longleftarrow \mathsf{Compress}_q(\mathbf{t}^T\mathbf{r} + e_2 + \lfloor \frac{q}{2} \rceil \cdot m, d_v)$
Return $(\mathbf{u}, v)$

Algorithm 4: Kyber.Dec $(sk = \mathbf{s}, (\mathbf{u}, v))$

$\mathbf{u}' \longleftarrow \mathsf{Decompress}_q(\mathbf{u}, d_u)$
$v' \longleftarrow \mathsf{Decompress}_q(v, d_v)$
Return $\mathsf{Compress}_q(v - \mathbf{s}^T\mathbf{u},\ 1)$

Protocol 4: Kyber

Alice		Bob
$\rho, \sigma \longleftarrow \{0,1\}^{256}$		
$\mathbf{A} \longleftarrow \mathsf{Sam}(\rho) \in \mathcal{R}_q^{k \times k}$		
$(\mathbf{s}, \mathbf{e}) \longleftarrow \mathsf{Sam}(\sigma) \in \psi_\eta^k \times \psi_\eta^k$		$m \longleftarrow \{0,1\}^{256}$
$\mathbf{t} \longleftarrow \mathsf{Compress}_q(\mathbf{As} + \mathbf{e}, d_t)$	$\xrightarrow{(\mathbf{t}, \rho)}$	$(\hat{K}, r) \longleftarrow G((\mathbf{t}, \rho), m)$
	$\xleftarrow{(\mathbf{u}, v)}$	$(\mathbf{u}, v) \longleftarrow \mathsf{Kyber.Enc}((\mathbf{t}, \rho), m; r))$
$m' \longleftarrow \mathsf{Dec}(\mathbf{s}, (\mathbf{u}, v))$		
$(\hat{K}', r') \longleftarrow G(H((\mathbf{t}, \rho)), m')$		
$(\mathbf{u}', v') \longleftarrow \mathsf{Kyber.Enc}((\mathbf{t}, \rho), m'; r')$		
$(\mathbf{u}', v') = (\mathbf{u}, v);$		
$sk' \longleftarrow H(\hat{K}', H((\mathbf{u}, v)))$		$sk \longleftarrow H(\hat{K}, H((\mathbf{u}, v)))$
$(\mathbf{u}', v') \neq (\mathbf{u}, v);$		
$sk' \longleftarrow H(z, H((\mathbf{u}, v)))$		

3.2.2 Lattice-based Group Key Exchange

Ding's Group Key Exchange

[22] proposed the first two-party key exchange protocol based on LWE and RLWE assumption and the **Protocol 5** describes how [22] runs.

Protocol 5: Ding Key Exchange

Alice		Bob
$\mathbf{s}_A, \mathbf{e}_A \longleftarrow D_{\mathbb{Z}^n,\alpha q}$		
$\mathbf{p}_A = \mathbf{M}\mathbf{s}_A + 2\mathbf{e}_A \mod q$	$\xrightarrow{\mathbf{p}_A}$	$\mathbf{s}_B \longleftarrow D_{\mathbb{Z}^n,\alpha q}, e_B \longleftarrow D_{\mathbb{Z},\alpha q}$
		$K_B = \mathbf{p}_A^T \cdot \mathbf{s}_B + 2e'_B$
		$\mathbf{p}_B = \mathbf{M}^T \cdot \mathbf{s}_B + 2\mathbf{e}_B \mod q$
	$\xleftarrow{(\mathbf{p}_B,\ \sigma)}$	$\sigma \longleftarrow S(K_B)$
$e'_A \longleftarrow D_{\mathbb{Z},\alpha q}$		
$K_A = \mathbf{s}_A^T \mathbf{p}_B + 2e'_A \mod q$		
$sk_A \longleftarrow E(K_A, \sigma)$		$sk_B \longleftarrow E(K_B, \sigma)$

[22] also extended **Protocol 5** into group setting which is similar to [14] and **Algorithm** 5 describes group version of Ding Key Exchange. But Ding *et al.* did not describe exact security proof of the **Algorithm** 5.

Apon *et al.*'s Protocol

Apon *et al.* proposed the first constant round group key exchange protocol and **Algorithm** 6 describes the process of [4].

Proof of correctness and security of [4] are describes in Chapter 6. The parameters $N, n, \sigma_1, \sigma_2, \lambda, \rho$ of the protocol are required to satisfy some constrains described as follows:

$$(N^2 + 2N) \cdot \sqrt{n} \cdot \rho^{3/2} \cdot \sigma_1^2 + (\frac{N^2}{2} + 1) \cdot \sigma_1 + (N-2) \cdot \sigma_2 \leq \beta_{Rec}$$

$$2N\sqrt{n}\lambda^{3/2}\sigma_1^2 + (N-1)\sigma_1 \leq \beta_{\text{Rènyi}}$$

$$\sigma_2 = \Omega(\beta_{\text{Rènyi}}\sqrt{n/log(\lambda)})$$

Apon *et al.* did not construct the key reconciliation mechanism, executed as subroutine in 6, [4] in detail. They only mentioned that any key reconciliation mechanism can be applied as subroutine.

Algorithm 5: Ding Group Key Exchange $(P[0,1,\cdots,k-1], q, n, \chi, a)$

1. For a party P_i,
   (i) $s_i, e_i \longleftarrow \chi$ and computes $p_i^0 = a \cdot s_i + e_i^0 \in R_q$;
   (ii) Send p_i^0 to $P(i+1)$;
2. For P_{i+j}, $(1 \leq j \leq k-2)$,
   (i) Computes $p_i^j = p_i^{j-1} \cdot s_{i+j} + 2e_i^j \mod q$ where $e_i^j \longleftarrow \chi$
   (ii) Sends p_i^j to a party P_{i+j+1}
3. For the party P_0,
   (i) $\hat{e_0} \longleftarrow \chi$ and computes $K_0 = p_1^{k-2} \cdot s_0 + 2\hat{e}_0$
   (ii) Computes and broadcast a signal $\sigma \longleftarrow S(K_0)$
   (iii) $sk_0 \longleftarrow E(K_0, \sigma)$
4. For a party P_j except P_0,
   (i) $\hat{e}_j \longleftarrow \chi$ and computes $K_i = p_{i+1}^{k-2} \cdot s_i + 2\hat{e}_j$
   (ii) $sk_i \longleftarrow E(K_i, \sigma)$

Algorithm 6: ADGK19$(P[0,1,\cdots,N-1], a, \mathcal{H}, \sigma_1, \sigma_2)$

(Round 1) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. Computes $z_i = as_i + e_i$ where $s_i, e_i \leftarrow \chi_{\sigma_1}$;
2. Broadcasts z_i;

(Round 2) For $i = 0$ to $N-1$, do the following in parallel.

1. If $i = 0$, party P_0 samples $e_0' \leftarrow \chi_{\sigma_2}$ and otherwise, party P_i samples $e_i' \leftarrow \chi_{\sigma_1}$;
2. Each party P_i broadcasts $X_i = (z_{i+1} - z_{i-1})\, s_i + e_i'$;

(Round 3) For a party P_{N-1}, do the following.

1. Samples $e_{N-1}'' \leftarrow \chi_{\sigma_1}$ and calculates
   $b_{N-1} = z_{N-2} N s_{N-1} + (N-1)X_{N-1} + (N-2)X_0 + \cdots + X_{N-3} + e_{N-1}''$;
2. Runs $\mathsf{recMsg}(\cdot)$ to output $(\mathsf{rec}, k_{N-1}) = \mathsf{recMsg}(b_{N-1})$;
3. Broadcasts rec and gets session key as $\mathsf{sk}_{N-1} = \mathcal{H}(k_{N-1})$;

(Key Computation) For each party P_i $(i \neq N-1)$.

1. Computes $b_i = z_{i-1} N s_i + (N-1)X_i + (N-2)X_{i+1} + \cdots + X_{i+N-2}$;
2. Runs $\mathsf{recKey}()$ to output $k_i = \mathsf{recKey}\,(b_i, \mathsf{rec})$ and gets session key as $\mathsf{sk}_i = \mathcal{H}(k_i)$;

3.2.3 Key Reconciliation Mechanism

A typical group key exchange protocol based on DLP, such as [16, 23], helps parties share the group key computed by

$$ek_i = (z_{i-1})^{Nr_i} \cdot X_i^{N-1} \cdot X_{i+1}^{N-2} \cdot \cdots \cdot X_{i+N-2}.$$

The group key is the form of $g^{s_0 s_1 + s_1 s_2 + \cdot s_{N-1} s_0}$ since group multiplication is commutative. However, multiplication in LWE or RLWE is not commutative because of errors, so that we cannot induce the same group key by following the same protocols [16, 23]. Hence, [22] proposed the first key reconciliation mechanism. After that, Peikert also presented another key reconciliation mechanism [34] but almost the same with [22]. Ding *et al.* and Peikert's key reconciliation mechanism return 0 and 1 depending on which interval includes an input. Figure 3.1 describes such interval domains.

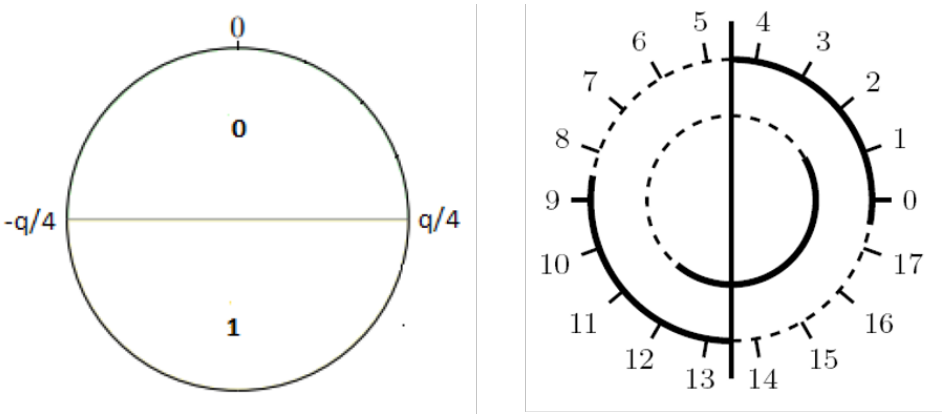

Figure 3.1: Ding Key Exchange [22] and Peikert [34]

However, [3] proposed a different type of key reconciliation mechanism using Voronoi cell and CVP, and we described the key reconciliation mechanism of [3] in Chapter 3.2 briefly. In this section, we describe more details about CVP and Decode algorithm used in NewHope

Algorithm 7: CVP ($\mathbf{x} \in \mathbb{R}^4$)

$\mathbf{v}_0 \longleftarrow \lfloor \mathbf{x} \rceil$
$\mathbf{v}_1 \longleftarrow \lfloor \mathbf{x} - \mathbf{g} \rceil$
$k \longleftarrow (\|\mathbf{x} - \mathbf{v_0}\|_1 < 1) \ ? \ 0 \ : \ 1$
$(v_0, v_1, v_2, v_3)^T \longleftarrow \mathbf{v}_k$
Return $(v_0, v_1, v_2, k)^T + v_3 \cdot (-1, -1, -1, 2)^T$

Algorithm 8: Decode ($\mathbf{x} \in \mathbb{R}^4/\mathbb{Z}^4$)

$\mathbf{v} \longleftarrow \mathbf{x} - \lfloor \mathbf{x} \rceil$
Return $(\|\mathbf{v}\|_1 < 1) \ ? \ 0 \ : \ 1$

CVP returns an integer vector related to a closest vector $\mathbf{v}$ given a basis $\mathbf{B}$, and Decode show whether input vector and its closest vector are in the same Voronoi cell.

[3] proved that the group keys computed by two parties through two key reconciliation functions HelpRec and Rec are same if inputs of each party are close enough. We discuss more details in Chapter 6.

3.3 Key-Reuse Attack

Attacks using keys are divided into two types: key-misuse attack and key-reuse attack. We do not cover the key-misuse attack in this paper because it is out of our scope. In 2016, Fluhrer proposed that a secret key computed through the signal function from [22] could be leaked. [24, 19] Surprisingly this key-reuse attack also can be applied to Peikert's key reconciliation mechanism since the key reconciliation function in [34] behaves like [22]. Hence, protocols based on key reconciliation function from [22, 34] is vulnerable to key-reuse attack. 9 showed steps of key-reuse attack under some assumptions that $\mathcal{A}$ can execute many sessions with Bob and set his/her public key deviated from the protocol.

Algorithm 9: Key-reuse attack [24, 19]

1. For $k = 0, 1, \cdots, q-1$, $\mathcal{A}$ does followings:
   (i) take $s_i = 0, e_i = 1$ and sets $pk_{\mathcal{A},k} = k$
   (ii) invoke the oracle $\mathcal{S}$ with $pk_{\mathcal{A},k}$ and obtains output from $\mathcal{S}$;
   (iii) analyze the number of changing signal values
   (iv) guess an absolute value of i-th coefficient of Bob's secret $s'_B[i]$
2. For $k = 0, 1, \cdots, q-1$, $\mathcal{A}$ does followings:
   (i) set $pk_{\mathcal{A},k} = (1 + x) \cdot k$,
   (ii) invoke the oracle $\mathcal{S}$ with $pk_{\mathcal{A},k}$ and obtains output from $\mathcal{S}$;
   (iii) analyze relations between two adjacent coefficients $s_B[i]$ and $s_B[i+1]$
   (iv) guess a sign of $s'_B[i]$ and analyzes a distribution of $p_B - a \cdot s'_B$

   If $p_B - a \cdot s'_B$ follows the discrete Gaussian distribution, then $s'_B = s_B$, and if $p_B - a \cdot s'_B$ looks random, then $s'_B = -s_B$.

Such attack like [24, 19] cannot reveal any information of secret key since [3] proposed the different type of key reconciliation mechanism from [22, 34]. However, recently Liu *et al.* presented a key-reuse attack on NewHope. [31] **Algorithm** 10 shows how attack carries out in briefly.

In step 2-(ii)-(b) from **Algorithm** 10, the adversary $\mathcal{A}$ computes the $l + i \cdot \frac{n}{4}$-th coefficient of Bob's

Algorithm 10: Key-reuse attack on NewHope [31]

1. For $k = 0, 1, \cdots, q-1$, $\mathcal{A}$ does followings:
   (i) take $s_i = 0, e_i = 1$ and sets $pk_{\mathcal{A},k} = k$
   (ii) invoke the oracle $\mathcal{S}$ with $pk_{\mathcal{A},k}$ and obtains output $\mathbf{r}_k = (r_{k,0}, r_{k,1}, \cdots, r_{k,n-1})$ from $\mathcal{S}$;
2. For $l = 0, 1, \cdots, \frac{n}{4}$, $\mathcal{A}$ does followings:
   (i) For $k = 0, 1, \cdots, q-1$, set $\mathbf{r}_{k,l} = \left(r_{k,l}, r_{k,l+\frac{n}{4}}, r_{k,l+2\cdot\frac{n}{4}}, r_{k,l+3\cdot\frac{n}{4}}\right)$, and set vectors $\mathbf{u}_{k,l} = \mathbf{B}\mathbf{r}_{k,l} \mod 4$ and $\mathbf{v}_{k,l} = (\mathbf{B}\mathbf{r}_{k,l} + 4\mathbf{g}) \mod 4$
   (ii) For $i = 0, 1, 3, 4$,
      (a) set two sequences $U_{l+i\cdot\frac{n}{4}} = (\mathbf{u}_{k,l}[i])_{k=0,1,\cdots,q-1}$ and $V_{l+i\cdot\frac{n}{4}} = (\mathbf{v}_{k,l}[i])_{k=0,1,\cdots,q-1}$
      (b) using $U_{l+i\cdot\frac{n}{4}}$ and $V_{l+i\cdot\frac{n}{4}}$, compute $s_B[l + i\cdot\frac{n}{4}]$

secret key. Let's see more details. Note that l-th split vector in k-th session $\mathbf{x}_{k,l} = (k\cdot s_B[l], k\cdot s_B[l+1\cdot\frac{n}{4}], k\cdot s_B[l+2\cdot\frac{n}{4}], k\cdot s_B[l+3\cdot\frac{n}{4}])^T \mod q$ of v in **Protocol 2**. Suppose that $b = 0$ for convenience. Let

$$\begin{aligned}\mathbf{r}_{k,l} &\longleftarrow \mathsf{HelpRec}(\mathbf{x}_{k,l}, 0) \in \mathbb{Z}_4^4\\ \hat{\mathbf{r}}_{k,l} &\longleftarrow \mathsf{CVP}_{\tilde{D}_4}\left(\frac{4}{q}\mathbf{x}_{k,l}\right) \in \mathbb{Z}^4\\ \mathbf{a}_{k,l} &= \mathbf{B}\hat{\mathbf{r}}_{k,l} \in \mathbb{R}^4.\end{aligned}$$

Since $\mathbf{r}_{k,l} = \hat{\mathbf{r}}_{k,l} \mod 4$ by definition, and we know that

$$\mathbf{a}_{k,l} \mod 4 = \begin{cases}\mathbf{B}\mathbf{r}_{k,l} \mod 4\\ (\mathbf{B}\mathbf{r}_{k,l} + 4\mathbf{g}) \mod 4.\end{cases}$$

From the definition of $\mathbf{a}_{k,l} = \mathbf{B}\hat{\mathbf{r}}_{k,l}$, $\mathbf{a}_{k,l}[i] \mod 4$ has one of the value :

$$\mathbf{a}_{k,l}[i] = \begin{cases}\lfloor\frac{4}{q}(k\cdot s_B[l+i\cdot\frac{n}{4}] \mod q)\rceil & \text{if } \|\frac{4}{q}\mathbf{x}_{k,l} - \lfloor\frac{4}{q}\mathbf{x}_{k,l}\rceil\| < 1,\\ \lfloor\frac{4}{q}(k\cdot s_B[l+i\cdot\frac{n}{4}] \mod q) - \frac{1}{2}\rceil + \frac{1}{2} & \text{otherwise.}\end{cases}$$

Liu *et al.* generalized that two values as followings:

$$\begin{aligned}f_{0,x}(h) &= \lfloor\frac{4}{q}\cdot(h\cdot x \mod q)\rceil\\ f_{1,x}(h) &= \left(\lfloor\frac{4}{q}\cdot(h\cdot x \mod q) - \frac{1}{2}\rceil + \frac{1}{2}\right) \mod 4,\end{aligned}$$

and also showed that $f_{0,x}(h)$ and $f_{1,x}(h)$ have a periodic property with period $L = \frac{q}{x \mod \pm q}$. In this case, $x = s_B[l + i\cdot\frac{n}{4}$, so that period indicates the absolute value of $s_B[l + i\cdot\frac{n}{4} \mod \pm q$. To determine

a sign of $s_B[l + i \cdot \frac{n}{4}]$, we have to analyze a behaviour of $\mathbf{a}_{k,l}[i]$. If $s_B[l + i \cdot \frac{n}{4} \mod^{\pm} q > 0$, then $\mathbf{a}_{k,l}[i]$ is increasing sequence, and if $s_B[l + i \cdot \frac{n}{4} \mod^{\pm} q < 0$, then $\mathbf{a}_{k,l}[i]$ is decreasing sequence.

Now, we want to guess $(\mathbf{a}_{k,l}[i] \mod 4)_{k=0,1,\cdots,q-1}$ using $U_{l,i\cdot\frac{n}{4}}$ and $V_{l+i\cdot\frac{n}{4}}$. Firstly, we construct two seqences $W_{l+i\cdot\frac{n}{4}}$ and $T_{l+i\cdot\frac{n}{4}}$ where $W_{l+i\cdot\frac{n}{4}}$ is for a positive sign, and $T_{l+i\cdot\frac{n}{4}}$ is for a negative sign of to guess $s_B[l + i \cdot \frac{n}{4}] \mod^{\pm} q$. The first sequence w_0 of $W_{l+i\cdot\frac{n}{4}}$ and $T_{l+i\cdot\frac{n}{4}}$ is defined as

$$w_0 = \begin{cases} \mathbf{u}_{0,l}[i] & \text{if } |\mathbf{u}_{0,l}[i] - 2| > |\mathbf{v}_{0,l}[i] - 2| \\ \mathbf{v}_{0,l}[i] & \text{otherwise,} \end{cases}$$

and then $W_{l+i\cdot\frac{n}{4}} = (w_k)_{k=0,1,\cdots,q-1}$ is constructed as following:

$$w_k = \begin{cases} w_0 & \text{if } k = 0 \\ \mathbf{u}_{k,l}[i] & \text{if } (k > 0) \wedge ((-0.5 \leq (\mathbf{u}_{k,l}[i] - w_{k-1}) \mod^{\pm} 4 \leq 0.5) \vee ((\mathbf{u}_{k,l}[i] - w_{k-1}) \mod 4 = 1)) \\ \mathbf{v}_{k,l}[i] & \text{otherwise,} \end{cases}$$

and $T_{l+i\cdot\frac{n}{4}} = (t_k)_{k=0,1,\cdots,q-1}$ is constructed as following:

$$t_k = \begin{cases} w_0 & \text{if } k = 0 \\ \mathbf{u}_{k,l}[i] & \text{if } (k > 0) \wedge ((-0.5 \leq (\mathbf{u}_{k,l}[i] - t_{k-1}) \mod^{\pm} 4 \leq 0.5) \vee ((\mathbf{u}_{k,l}[i] - w_{k-1}) \mod 4 = -1)) \\ \mathbf{v}_{k,l}[i] & \text{otherwise.} \end{cases}$$

As mentioned before, if $s_B[l+i\cdot\frac{n}{4}] \mod^{\pm} q > 0$, $W_{l+i\cdot\frac{n}{4}}$ is $(\mathbf{a}_{k,l}[i] \mod 4)_{k=0,1,\cdots,q-1}$, and if $s_B[l+i\cdot\frac{n}{4}] \mod^{\pm} q < 0$, $T_{l+i\cdot\frac{n}{4}}$ is $(\mathbf{a}_{k,l}[i] \mod 4)_{k=0,1,\cdots,q-1}$.

To determine the sign of $s_B[l + i \cdot \frac{n}{4}] \mod^{\pm} q$, we have to analyze $W_{l+i\cdot\frac{n}{4}}$ and $T_{l+i\cdot\frac{n}{4}}$. When $W_{l+i\cdot\frac{n}{4}} = T_{l+i\cdot\frac{n}{4}}$ the value *sign* indicates of the sign of $s_B[l+i\cdot\frac{n}{4}] \mod^{\pm} q$. If $s_B[l+i\cdot\frac{n}{4}] \mod^{\pm} q > 0$, then $sign > 0$, and if $s_B[l + i \cdot \frac{n}{4}] \mod^{\pm} q < 0$, then $sign < 0$.

$$sign = \sum_{k \in [0, q/period];} (w_{k+1} - w_k)$$

$$|w_{k+1} - w_k| < 3,$$

When $W_{l+i\cdot\frac{n}{4}} \neq T_{l+i\cdot\frac{n}{4}}$, compute and compare variancs $var_{W_{l+i\cdot\frac{n}{4}}}$ and $var_{T_{l+i\cdot\frac{n}{4}}}$ of each sequence, and if $var_{W_{l+i\cdot\frac{n}{4}}} < var_{T_{l+i\cdot\frac{n}{4}}}$, then we take the positive sign, and otherwise we take the negative sign.

Chapter 4. Instantiation of Apon *et al.*'s Protocol

In this chapter, we instantiate ADGK19 by applying the key reconciliation mechanism of **NewHope** to generic key reconciliation mechanism recMsg and recKey and rem differences with ADGK19.

We assume that N participants communicate in broadcasting topology and determine $R_q, \chi_{\sigma_1}, \chi_{\sigma_2}, a \longleftarrow R_q$ same as in ADGK19. If b_i's computed by i-th party P_i in **Key computation** is close enough to each other b_j for $i \neq j$, then all participants can agree on the same group key. We take $\sigma_1 = 2\sqrt{8}$ to make discrete Gaussian distribution approximate to centered binomial distribution used in **NewHope**.

Algorithm 11: Instatiated ADGK$(P[0, 1, \cdots, N-1], a, \mathcal{H}, \sigma_1, \sigma_2)$

(Round 1) For each party P_i for $i = 0$ to $N-1$, do the following in parallel.

1. Computes $z_i = as_i + e_i$ where $s_i, e_i \leftarrow \chi_{\sigma_1}$;
2. Broadcasts z_i;

(Round 2) For $i = 0$ to $N-1$, do the following in parallel.

1. If $i = 0$, party P_0 samples $e'_0 \leftarrow \chi_{\sigma_2}$ and otherwise, party P_i samples $e'_i \leftarrow \chi_{\sigma_1}$;
2. Each party P_i broadcasts $X_i = (z_{i+1} - z_{i-1})\, s_i + e'_i$;

(Round 3) For a party P_{N-1}, do the following.

1. Samples $e''_{N-1} \leftarrow \chi_{\sigma_1}$ and calculates
   $b_{N-1} = z_{N-2} \cdot N \cdot s_{N-1} + (N-1) \cdot X_{N-1} + (N-2) \cdot X_0 + \cdots + X_{N-3} + e''_{N-1}$;
2. a signal $\mathbf{r} \longleftarrow FullHelpRec(b_{N-1})$;
3. an ephemeral key $ek_{N-1} \longleftarrow FullRec(b_{N-1}, \mathbf{r})$
4. Broadcasts $\mathbf{r}$ and computes the group key as $\mathsf{sk}_{N-1} = \mathcal{H}(ek_{N-1})$;

(Key Computation) For each party P_i $(i \neq N-1)$.

1. Computes $b_i = z_{i-1} \cdot N \cdot s_i + (N-1) \cdot X_i + (N-2) \cdot X_{i+1} + \cdots + X_{i+N-2}$;
2. Computes an ephemeral key $ek_i \longleftarrow FullRec\,(b_i, \mathbf{r})$ and computes the group key by $\mathsf{sk}_i = \mathcal{H}(ek_i)$;

Note that the key reconciliation mechanism in ADGK19 consists of two functions, recMsg and recKey. In **Round 3**, recMsg outputs two value, rec and k_{N-1}. We divide recMsg into two functions in NewHope, FullHelpRec and FullRec. FullHelpRec returns a signal value $\mathbf{r}$, and FullRec takes input as b_i and $\mathbf{r}$ and returns an ephemeral key ek_i. recKey is substituted by FullRec.

Besides, we sample the public value $a \in R_q$ in every session to prevent a backdoor attack. If not,

dishonest authority can recover each party's secret key. For example, assume that a dishonest authority chooses small $f, g \equiv 1 \mod p$ for some prime $p \geq 4 \cdot 16 + 1$ and set $a = g \cdot f^{-1} \mod q$. Then, given an RLWE instance $(a, b = a \cdot s + e)$, $bf = (as + e)f = afs + ef = gs + ef \mod q$ and the authority can compute $gs + ef \in \mathbb{Z}$ since g, s, e, f, are sufficiently small. Then the authority can compute $t = s + e \mod p$, and by computing $(b - t) \cdot (a = 1)^{-1} \mod q$, s can be recovered. Therefore, by sampling a in every session, we can prevent this backdoor attack. Also, we can avoid a key-reuse attack when we sample a in every session. We describe a key-reuse attack in chapter 6.

Chapter 5. Security Analysis

5.1 Correctness Proof

Alkim *et al.* suggested an exact upper bound for correctness.

Lemma 1 (Restate in [3])**.** If $\|\mathbf{x} - \mathbf{x}'\|_1 < (1 - 1/2^r) \cdot q - 2$, then two session keys k and k' computed from protocol are same. Additionally, if $\mathbf{x}$ is uniform, then the driven k is uniform and independent to a signal r.

Protocol: Simplified NewHope		
Alice		**Bob**
$\mathbf{x} \in \mathbb{Z}_q^4$	$\mathbf{x} \approx \mathbf{x}'$	$\mathbf{x}' \in \mathbb{Z}_q^4$
	$\mathbf{r}$	$\mathbf{r} \longleftarrow \mathsf{HelpRec}(\mathbf{x})$
	$\longleftarrow$	
$k' \longleftarrow Rec(\mathbf{x}', \mathbf{r})$		$k \longleftarrow Rec(\mathbf{x}, \mathbf{r})$

In [4], Apon *et al.* suggested β_{Rec} as an upper bound for a coefficient of $(b_{N-1} - b_i)$ to achieve correctness.

Theorem 6 (Restate [4])**.** Fix a statistical security parameter ρ, and assume

$$(N^2 + 2N) \cdot \sqrt{n} \cdot \rho^{3/2} \cdot \sigma_1^2 + (\frac{N^2}{2} + 1) \cdot \sigma_1 + (N - 2) \cdot \sigma_2 \leq \beta_{Rec}.$$

Then all parties have the same session key except with probability at most $2^{-\rho+1}$.

In **Lemma** 1, $\|\mathbf{x} - \mathbf{x}'\|_1$ is the sum of 4 coordinates of $(\mathbf{x} - \mathbf{x}')$ and Alkim *et al.* suggested an upper bound for this summation, however in **Theorem** 6, Apon *et al.* set an upper bound for one coefficient, so that we cannot compare inequalities directly. To compare inequalities in **Lemma** 1 and **Theorem** 6, we need to adjust $\|\mathbf{x} - \mathbf{x}'\|_1$.

Since each coefficient of $\mathbf{x}$ and $\mathbf{x}'$ is independently sampled from ψ_η, $(\|\mathbf{x} - \mathbf{x}'\|_1)/4$ stands for one coefficient of $\|\mathbf{x} - \mathbf{x}'\|_1$. Then to satisfy **Lemma** 1 and **Theorem** 6, we have two inequalities 5.1 and 5.2. If $\|\mathbf{x} - \mathbf{x}'\|_1/4$ satisfies such inequalities, then all participants of a group can compute the same group key except with at most probability $2^{-\rho+1}$.

$$(\|\mathbf{x} - \mathbf{x}'\|_1)/4 \leq (N^2 + 2N) \cdot \sqrt{n} \cdot \rho^{3/2} \cdot \sigma_1^2 + (\frac{N^2}{2} + 1) \cdot \sigma_1 + (N - 2) \cdot \sigma_2 \tag{5.1}$$

$$< \left(\frac{3}{4} \cdot q - 2\right)/4 =: \beta_{\text{Rec}} \tag{5.2}$$

5.2 Security Proof

Theorem 7 (Restated [4])**.** Assume $2N \cdot \sqrt{n} \cdot \lambda^{3/2} \cdot \sigma_1^2 + (N-1) \cdot \sigma_1 \leq \beta_{\text{Rènyi}}$ and $\beta_{\text{Rènyi}} < \sigma_2 < q$, and let $\mathcal{H}$ be a random oracle. Then

$$\mathsf{Adv}_{\Pi}^{\mathsf{GKE}}(t, \mathsf{q}) \leq 2^{-\lambda+1} + \sqrt{\left(N \cdot \mathsf{Adv}_{n,q,\chi_{\sigma_1},3}^{\mathsf{RLWE}}(t_1) + \mathsf{Adv}_{\mathsf{KeyRec}}(t_2) + \frac{\mathsf{q}}{2^{\lambda}}\right) \cdot \frac{exp\left(2\pi n(\beta_{\text{Rènyi}}/\sigma_2)^2\right)}{1 - 2^{-\lambda+1}}},$$

where $\mathsf{Adv}_{\Pi}^{\mathsf{GKE}}(t, \mathsf{q})$ is an advantage in distinguishing a pair (transcript, sk) between execution protocol and uniform distribution, $\mathsf{Adv}_{n,q,\chi_{\sigma_1},3}^{\mathsf{RLWE}}(t_1)$ denotes the maximum advantage in distinguishing RLWE instances and $\mathsf{Adv}_{\mathsf{KeyRec}}(t_2)$ denotes the advantage in distinguishing an ephemeral key ek between $ek \longleftarrow KeyRec$ and uniform distribution, and $t_1 = t + \mathcal{O}(N \cdot t_{\mathsf{ring}})$, $t_2 = t + \mathcal{O}(N \cdot t_{\mathsf{ring}})$ and t_{ring} is the time required to perform operations in R_q.

Theorem 7 indicates the security of ADGK19, and Apon *et al.* proved that $\mathsf{Adv}_{\Pi}^{\mathsf{GKE}}(t, \mathsf{q})$ is negligible. In detail, by RLWE assumption, we know that $\mathsf{Adv}_{n,q,\chi_{\sigma_1},3}^{\mathsf{RLWE}}$ is negligible. Now we consider the second term, the advantage $\mathsf{Adv}_{\text{KeyRec}}(\mathcal{A})$ of adversary $\mathcal{A}$ about distinguishing an ephemeral key and a uniformly random value. Note that $\mathsf{Adv}_{\text{KeyRec}}(t_2)$ means the maximum advantage of any such adversary running in time t_2. In **Lemma** 1, Alkim *et al.* showed that if $\mathbf{x}$ is uniform, then the session key k computed by the key reconciliation mechanism is uniform. Then it is enough to show that b_{N-1}, which is an input of FullHelpRec in **Round 3**, is uniform or indistinguishable from the uniform distribution in our instantiation, and Apon *et al.* already proved what we have to show. However, in this paper, we propose a new approach to prove b_{N-1} is indistinguishable from the uniform distribution. Before showing b_{N-1} is uniform, we describe a theorem that is used in our protocol mainly. After that, we will prove **Lemma** 2, describing that b_{N-1} is indistinguishable from the uniform distribution.

Theorem 8 (Restate [21])**.** Let $x, y \in R_q$ such that x follows an arbitrary distribution ϕ over R_q and y is indistinguishable from a uniformly chosen value in R_q. Then $\bar{x} = x + y$ is indistinguishable from a uniformly chosen value in R_q.

Lemma 2. The b_{N-1} is indistinguishable from uniform.

Proof. Firstly, we show that $z_{i+1} - z_{i-1}$ is indistinguishable from uniformly random. Since z_{i+1} and z_{i-1} are both RLWE instances, both z_{i+1} and z_{i-1} are indistinguishable from uniformly random. By the theorem 8, clearly $z_{i+1} - z_{i-1}$ is indistinguishable from a uniformly random value, and hence, X_i computed in **Round 2** of **Algorithm** 11 is RLWE instance.

For $k \in \mathbb{Z}_q$ and X_i, since k is invertible in R_q and X_i is indistinguishable from uniformly random one, clearly $k \cdot X_i$ is indistinguishable from uniformly random value. Then by theorem 8, for $k_i, k_j in \mathbb{Z}_q$ and $X_i, X_j \in R_q$, $k_i \cdot X_i + k_j \cdot X_j$ is also indistinguishable from uniformly random value.

In **Round 3**, $z_{N-2} \in R_q$ and $N \in Z_q$, since N is invertible and z_{N-2} is indistinguishable from uniformly random, $z_{N-2} \cdot N$ is also indistinguishable with uniformly random value, and this implies that $z_{N-2} \cdot N \cdot s_{N-1} + e''_{N-1}$ of b_{N-1} from **Round 3** of algorithm 11 is RLWE instance which means that $z_{N-2} \cdot N \cdot s_{N-1} + e''_{N-1}$ is also indistinguishable from uniformly random.

Therefore, by theorem 8, $b_{N-1} = z_{N-2} \cdot N \cdot s_{N-1} + (N-1) \cdot X_{N-1} + (N-2) \cdot X_0 + \cdots + X_{N-3} + e''_{N-1}$ is indistinguishable from uniform. □

Conditions for parameter setting

To guarantee correctness and security of our instantiation, parameters $N, n, \sigma_1, \sigma_2, \lambda, \rho, \beta_{Rec}$ and $\beta_{\text{Rènyi}}$ are also satisfied the following conditions including inequalities 5.1 and 5.2:

$$(N^2 + 2N) \cdot \sqrt{n}\rho^{3/2}\sigma_1^2 + (\frac{N^2}{2} + 1)\sigma_1 + (N-2)\sigma_2 \leq \beta_{\text{Rec}} \tag{5.3}$$

$$2N\sqrt{n}\lambda^{3/2}\sigma_1^2 + (N-1)\sigma_1 \leq \beta_{\text{Rènyi}} \tag{5.4}$$

$$\sigma_2 = \Omega(\beta_{\text{Rènyi}} \cdot \sqrt{n/log(\lambda)} \tag{5.5}$$

$$\beta_{\text{Rènyi}} < \sigma_2 < q \tag{5.6}$$

$$2n(N^2 + 2N) \leq 2^{\rho} \tag{5.7}$$

$$\rho = \lambda \tag{5.8}$$

$$|\mathsf{Error}_j| \leq \beta_{\text{Rènyi}} \tag{5.9}$$

$$\tag{5.10}$$

We compute some parameters under an assumption that $N = 10$ participants comprise a group and want to share the group key. Under $n = 512$, $N = 10$, $\lambda = 128$, $\rho = 128$, and $\sigma_1 = 2\sqrt{8}$, we have $\beta_{\text{Rec}} = 293,602,943.6875$, $\beta_{\text{Rényi}} = 20,971,572$, $\sigma_2 = 20,971,573$, and $q = 1,565,882,369$. These parameters are really big number compare to $q = 12289$ used in NewHope, and are not optimized yet.

Chapter 6. Vulnerability by Key-Reuse Attack

In this chapter, we extend [31] trying the key-reuse attack on [3], two-party key exchange protocol, into our instantiation, the group key exchange protocol. We do not consider the last error e''_{N-1} in b_{N-1} computed in **Round 3** for convenience. Then we assume followings:

1. Fix a public value $a \in R_q$.
2. P_0, P_{N-3} and P_{N-2} are manipulated by the adversary $\mathcal{A}$.
   (a) P_{N-2} sets his/her public key as some k where $k \in \mathbb{Z}_q$.
   (b) P_0 sets his/her public key z_0 is sufficiently small, and P_{N-3} broadcasts his public value X_{N-3} as always fixed and independent to z_{N-2}.
3. $\mathcal{A}$ can initiate many sessions with all other parities in group.
4. $\mathcal{A}$ can access to an oracle $\mathbb{S}$ outputting the value of *FullHelpRec*;
5. All parties except P_{N-1} fix their secret key s_i.

Now, we describe details of the extended key-reuse attack. In **Round 3**, we can rewrite b_{N-1} as

$$\begin{aligned} b_{N-1} &= z_{N-2} \cdot N \cdot s_{N-1} + (N-1) \cdot X_{N-1} + (N-2) \cdot X_0 + \cdots + X_{N-3} \\ &= z_{N-2} \cdot s_{N-1} + (N-1) \cdot z_0 \cdot s_{N-1} + \sum_{j=1}^{N-2} (N-1-j) \cdot X_{N-1+j}, \end{aligned}$$

and since the second term, $(N-1) \cdot z_0 \cdot s_{N-1} + \sum_{j=1}^{N-2}(N-1-j) \cdot X_{N-1+j}$, is independent to $z_{N-2}(= k)$, we consider this second term as a constant value, and note that $(b_{N-1})_{k,l}$ denotes l-th split vector of b_{N-1} in k-th session. The l-th split vector in k-the session $(b_{N-1})_{k,l} = (k \cdot s_{N-1})_{k,l} + \left((N-1) \cdot z_0 \cdot s_{N-1} + \sum_{j=1}^{N-2}(N-1-j) \cdot X_{N-1+j}\right)_{k,l}$, and we put $y_{k,l} = ((N-1) \cdot z_0 \cdot s_{N-1} + \sum_{j=1}^{N-2}(N-1-j) \cdot X_{N-1+j})_{k,l}$. Let

$$\begin{aligned} \mathbf{r}_{k,l} &\longleftarrow \mathsf{HelpRec}\left((z_{N-2} \cdot s_{N-1})_{k,l}, 0\right) \in \mathbb{Z}_4^4 \\ \hat{\mathbf{r}}_{k,l} &\longleftarrow \mathsf{CVP}_{\tilde{D}_4}\left(\frac{4}{q}(z_{N-2} \cdot s_{N-1})_{k,l}\right) \in \mathbb{Z}^4 \\ \mathbf{a}_{k,l} &= \mathbf{B}\hat{\mathbf{r}}_{k,l} \in \mathbb{R}^4 \end{aligned}$$

and

$$\mathbf{r}'_{k,l} \longleftarrow \mathsf{HelpRec}\left((b_{N-1})_{k,l}, 0\right) \in \mathbb{Z}_4^4$$
$$\hat{\mathbf{r}}'_{k,l} \longleftarrow \mathsf{CVP}_{\tilde{D}_4}\left(\frac{4}{q}(b_{N-1})_{k,l}\right) \in \mathbb{Z}^4$$
$$\mathbf{a}'_{k,l} = \mathbf{B}\hat{\mathbf{r}}'_{k,l} \in \mathbb{R}^4$$

Let $\mathbf{r}'_{k,l} = \mathsf{HelpRec}\left((b_{N-1})_{k,l}\right)$, $\hat{\mathbf{r}}'_{k,l} = CVP\left(\frac{4}{q}(b_{N-1})_{k,l}\right)$ and $\mathbf{a}_{k,l} = \mathbf{B}\mathbf{r}'_{k,l}$. Then similar to [31],

$$\mathbf{a}_{k,l}[i] = \begin{cases} \lfloor \frac{4}{q}(k \cdot s_{N-1}[l + i \cdot \frac{n}{4}] \mod q) \rceil & \text{if } \|\frac{4}{q}\mathbf{x}_{k,l} - \lfloor \frac{4}{q}\mathbf{x}_{k,l} \rceil\| < 1, \\ \lfloor \frac{4}{q}(k \cdot s_{N-1}[l + i \cdot \frac{n}{4}] \mod q) - \frac{1}{2} \rceil + \frac{1}{2} & \text{otherwise.} \end{cases}$$

$$\mathbf{a}_{k,l}[i] = \begin{cases} \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] + d\left[l + i \cdot \frac{n}{4}\right] \mod q\right) \rceil & \text{if } \|\frac{4}{q}(b_{N-1})_{k,l} - \lfloor \frac{4}{q}(b_{N-1})_{k,l} \rceil\|_1 < 1 \\ \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] + d\left[l + i \cdot \frac{n}{4}\right] \mod q\right) - \frac{1}{2} \rceil + \frac{1}{2} & \text{otherwise ,} \end{cases}$$

where $d[j]$ is constant value generated from $y_{k,l}$. Now we divide the ceiling-floor function by each term like $\lfloor x + y \rceil = \lfloor x \rceil + \lfloor y \rceil + \delta_y(x)$ where $\delta_y(x) \in \{-1, 0, 1\}$. Hence,

$$\begin{aligned} \mathbf{a}_{k,l}[i] &= \begin{cases} \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] + d\left[l + i \cdot \frac{n}{4}\right] \mod q\right) \rceil & \text{if } \|\frac{4}{q}(b_{N-1})_{k,l} - \lfloor \frac{4}{q}(b_{N-1})_{k,l} \rceil\|_1 < 1 \\ \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] + d\left[l + i \cdot \frac{n}{4}\right] \mod q\right) - \frac{1}{2} \rceil + \frac{1}{2} & \text{otherwise ,} \end{cases} \\ &= \begin{cases} \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] \mod q\right) \rceil + \lfloor d\left[l + i \cdot \frac{n}{4}\right] \mod q \rceil & \\ \qquad + \delta_{d[l+i\cdot\frac{n}{4}]}(\frac{4}{q}k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] \mod q) & \text{if } \|\frac{4}{q}(b_{N-1})_{k,l} - \lfloor \frac{4}{q}(b_{N-1})_{k,l} \rceil\|_1 < 1 \\ \lfloor \frac{4}{q}\left(k \cdot s_B\left[l + i \cdot \frac{n}{4}\right] \mod q\right) - \frac{1}{2} \rceil + \frac{1}{2} + \lfloor d\left[l + i \cdot \frac{n}{4}\right] \mod q \rceil & \\ \qquad + \delta_{d[l+i\cdot\frac{n}{4}]}(\frac{4}{q}(k \cdot s_{N-1}\left[l + i \cdot \frac{n}{4}\right] \mod q) - \frac{1}{2}) & \text{otherwise ,} \end{cases} \\ &= \mathbf{a}_{k,l}[i] + \lfloor d\left[l + i \cdot \frac{n}{4}\right] \rceil + \delta_{d[l+i\cdot\frac{n}{4}]}(\mathbf{a}_{k,l}[i]), \end{aligned}$$

$\delta_y(x)$ has some patterns and that patterns is determined by y. y chooses one of column vector in

matrix below and x chooses row vector.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Since $d\left[l + i \cdot \frac{n}{4}\right]$ comes from $y_{k,l}$ and $y_{k,l} = \left((N-1) \cdot z_0 \cdot s_{N-1} + \sum_{j=1}^{N-2}(N-1-j) \cdot X_{N-1+j}\right)_{k,l}$, by our assumption, that is z_0 is sufficiently small, we consider $(N-1) \cdot z_0 \cdot s_{N-1}$ is also small. Hence, we know the approximated value to $y_{k,l}$ and we can figure out what pattern of $\delta_y(x)$ is. Therefore, from

$$\left(\mathbf{a}'_{k,l}[i] \mod 4\right)_{k=0,1,\cdots,q-1} = \left(\mathbf{a}_{k,l}[i] + d[l + i \cdot \frac{n}{4}] + \delta_{d\left[l+i\cdot\frac{n}{4}\right]}(\mathbf{a}_{k,l}[i]) \mod 4\right)_{k=0,1,\cdots,q-1}$$

we can induce

$$\left(\mathbf{a}'_{k,l}[i] - \delta_{d\left[l+i\cdot\frac{n}{4}\right]}(\mathbf{a}_{k,l}[i]) \mod 4\right)_{k=0,1,\cdots,q-1} = \left(\mathbf{a}_{k,l}[i] + d[l + i \cdot \frac{n}{4}] \mod 4\right)_{k=0,1,\cdots,q-1}.$$

Since $\left(d[l + i \cdot \frac{n}{4}] \mod 4\right)_{k=0,1,\cdots,q-1}$ is just a constant value, $\left(d[l + i \cdot \frac{n}{4}] \mod 4\right)_{k=0,1,\cdots,q-1}$ only shifts the sequence $(\mathbf{a}_{k,l}[i] \mod 4)_{k=0,1,\cdots,q-1}$, and preserves $(\mathbf{a}_{k,l}[i] \mod 4)_{k=0,1,\cdots,q-1}$'s properties, period and its behaviour. Therefore, by key-reuse attack of [31], we can figure out the secret key of P_{N-1}.

Similar to [31], when we add an error e''_{N-1} to b_{N-1}, almost nothing changes but some values is deviated from regular one. However, Liu *et al.* suggested a method to detect some irregular values so that we can induce a secret key of P_{N-1}

Countermeasure for Key-Reuse Attack

However, we can prevent such attack easily. In **Algorithm** 11, sampling a public value a in every session make a constant value in attack no more constant so that the adversary $\mathcal{A}$ cannot get periodic sequence from a signal value $\mathbf{r}$. Another method is proposed by [20].

Chapter 7. Concluding Remarks

As the use of group-based communication has increased recently, the security issue of group key exchange becomes an important task. Also, as developing the quantum computer develop very fastly, we have to consider quantum-resistant protocol. In this thesis, we instantiate the group key exchange protocol ADGK19 proposed in PQCrypto 2019 using the key reconciliation mechanism of **NewHope** and complete its correctness and security.

Since Fluhrer proposed the key-reuse attack, there are some variant researches, and one of them tried an attack on [3]. [31] We extend Liu *et al.*'s key-reuse attack into a group setting and verify that our instantiation has a vulnerability to such an attack. However, the attack could be leak information of secret key under some strong assumptions, and we show that sampling the public value $a \in R_q$ prevents our extended key-reuse attack.

As future work, we will optimize the parameters into our proposed instantiation. Our suggested parameters guarantee correctness and security but reduce performance severely. As another work, we try key-reuse attack under weaker assumptions. Our assumptions are not realistic since $\mathcal{A}$ has to manipulate at least two parties in a group , and all parties except P_{N-2} fix their secret key.

Bibliography

[1] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval. "Password-based group key exchange in a constant number of rounds". In *International Workshop on Public Key Cryptography*, pages 427–442. Springer, 2006.

[2] M. Abdalla and D. Pointcheval. "A scalable password-based group key exchange protocol in the standard model". In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 332–347. Springer, 2006.

[3] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. "Post-quantum key exchange—a new hope". In *25th USENIX Security Symposium (USENIX Security 16)*, pages 327–343, 2016.

[4] D. Apon, D. Dachman-Soled, H. Gong, and J. Katz. "Constant-round group key exchange from the ring-LWE assumption." *Post-Quantum Cryptography*, 2019:398, 2019.

[5] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. "Fast cryptographic primitives and circular-secure encryption based on hard learning problems". In *Annual International Cryptology Conference*, pages 595–618. Springer, 2009.

[6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. "Quantum supremacy using a programmable superconducting processor". *Nature*, 574(7779):505–510, 2019.

[7] L. Babai. "On Lovasz'lattice reduction and the nearest lattice point problem". In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 13–20. Springer, 1985.

[8] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. "Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance". In *21st International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT 2015*, pages 3–24. Springer, Springer Nature, 2015.

[9] J.-M. Bohli, M. I. G. Vasco, and R. Steinwandt. "Password-authenticated constant-round group key establishment with a common reference string." *IACR Cryptology ePrint Archive*, 2006:214, 2006.

[10] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila. "Frodo: Take off the ring! practical, quantum-secure key exchange from LWE". In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1006–1018. ACM, 2016.

[11] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM". In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[12] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem". In *2015 IEEE Symposium on Security and Privacy*, pages 553–570. IEEE, 2015.

[13] E. Bresson and D. Catalano. "Constant round authenticated group key agreement via distributed computation". In *International Workshop on Public Key Cryptography*, pages 115–129. Springer, 2004.

[14] E. Bresson, O. Chevassut, and D. Pointcheval. "Provably authenticated group Diffie-Hellman key exchange—the dynamic case". In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 290–309. Springer, 2001.

[15] E. Bresson, O. Chevassut, and D. Pointcheval. "Dynamic group Diffie-Hellman key exchange under standard assumptions". In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 321–336. Springer, 2002.

[16] M. Burmester and Y. Desmedt. "A secure and efficient conference key distribution system". In *Advances in Cryptology—EUROCRYPT'94*, pages 275–286. Springer, 1995.

[17] M. Burmester and Y. Desmedt. "A secure and scalable group key exchange system". *Information Processing Letters*, 94(3):137–143, 2005.

[18] K. Choi, J. Hwang, and D. Lee. "Efficient ID-based group key agreement with bilinear maps". *Public Key Cryptography–PKC 2004*, pages 130–144, 2004.

[19] J. Ding, S. Alsayigh, R. Saraswathy, S. Fluhrer, and X. 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.

[20] J. Ding, P. Branco, and K. Schmitt. "Key Exchange and Authenticated Key Exchange with Reusable Keys Based on RLWE Assumption". *IACR Cryptology ePrint Archive*, 2019:665, 2019.

[21] J. Ding, R. Saraswathy, S. Alsayigh, and C. Clough. "How to validate the secret of a Ring Learning with Errors (RLWE) key." *IACR Cryptology ePrint Archive*, 2018:81, 2018.

[22] J. Ding, X. Xie, and X. Lin. "A simple provably secure key exchange scheme based on the learning with errors problem." *IACR Cryptology ePrint Archive*, 2012:688, 2012.

[23] R. Dutta and R. Barua. "Constant round dynamic group key agreement". In *International Conference on Information Security*, pages 74–88. Springer, 2005.

[24] S. R. Fluhrer. "Cryptanalysis of ring-LWE based key exchange with key share reuse." 2016.

[25] L. K. Grover. "A fast quantum mechanical algorithm for database search". In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[26] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[27] J. Katz and J. S. Shin. "Modeling insider attacks on group key-exchange protocols". In *Proceedings of the 12th ACM conference on Computer and communications security*, pages 180–189. ACM, 2005.

[28] J. Katz and M. Yung. "Scalable protocols for authenticated group key exchange". *Advances in Cryptology-CRYPTO 2003*, pages 110–125, 2003.

[29] Y. Kim, A. Perrig, and G. Tsudik. "Simple and fault-tolerant key agreement for dynamic collaborative groups". In *Proceedings of the 7th ACM conference on Computer and communications security*, pages 235–244. ACM, 2000.

[30] A. Langlois and D. Stehlé. "Worst-case to average-case reductions for module lattices". *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[31] C. Liu, Z. Zheng, and G. Zou. "Key reuse attack on newhope key exchange protocol". In *International Conference on Information Security and Cryptology*, pages 163–176. Springer, 2018.

[32] V. Lyubashevsky, C. Peikert, and O. Regev. "On ideal lattices and learning with errors over rings". In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23. Springer, 2010.

[33] V. Lyubashevsky, C. Peikert, and O. Regev. "A toolkit for ring-LWE cryptography". In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 35–54. Springer, 2013.

[34] C. Peikert. "Lattice cryptography for the internet". In *Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, volume 8772, page 197. Springer, 2014.

[35] C. Peikert et al. "A decade of lattice cryptography". *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[36] O. REGEV. “On lattices, learning with errors, random linear dodes, and cryptography, 2005”. In *STOC*, pages 84–93. ACM, 2005.

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

[38] P. 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.

[39] D. Simon. “On the power of quantum computation”. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 116–123. IEEE, 1994.

[40] T. Van Erven and P. Harremos. “Rényi divergence and Kullback-Leibler divergence”. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[41] 이나비 and 김광조. “디지털 포렌식을 위한 Android 및 Windows 환경에서 카카오톡 메시지의 아티팩트 분석 (I)”. In *2019 년 한국정보보호학회 충청지부 학술논문발표회*. 한국정보보호학회, 2019.

Acknowledgments in Korean

본 논문을 작성하는데 많은 도움을 받았습니다. 먼저 연구의 방향과 연구자로서 갖춰야할 자세를 바르게 인도해주신 김광조 교수님께 깊은 감사를 드립니다. 석사생활 동안 연구하여 작성한 논문들을 직접 봐주시고 지도해주셔서 석사 논문까지 작성할 수 있었습니다. 또한 바쁘신 가운데 학위논문의 심사위원으로 참석해주신 강병훈 교수님과 이주영 교수님께도 감사의 말씀을 드립니다.

또한 같은 연구실에서 도움을 줬던 연구실 동료에게도 감사의 말씀을 전합니다.먼저 같이 연구를 진행하고 케어를 잘해준 락용이형, 연구실 적응을 잘하게 도와준 지은이 누나, 배려가 넘치는 Harry, 졸업 한 후에도 정신적 건강에 도움을 준 형철이형, 항상 에너지를 공급해주던 낙준이형, 취업을 비롯하여 많은 도움을 준 성호형, 함께 2년을 보낸 나비 누나 그리고 항상 행정적으로 도움을 준 홍지연 선생님께 감사드립니다. 함께 연구를 진행하고 같이 생활할 수 있어서 좋았습니다.

많은 부분이 부족하지만 그 부족한 점을 이해해주시고 감싸주신 모든 분들에게 감사드립니다. 특히 이 곳에서 흔들리지 않고 연구하고 공부하는 것을 지지해준 엄마와 형에게 큰 감사를 드립니다. 연구하고 배웠던 지식을 나누며 바른 연구자로서 성장하도록 하겠습니다.

마지막으로 이 모든 것을 하게 해주신 하나님께 감사드립니다.

Curriculum Vitae in Korean

이 름: 홍 동 연

생 년 월 일: 1993년 03월 17일

전 자 주 소: decenthong93@kaist.ac.kr

학 력

2009. 3. – 2012. 2. 서울 배명고등학교

2012. 3. – 2018. 2. 경북대학교 수학과 (B.S.)

2018. 3. – 2020. 2. 한국과학기술원 정보보호대학원 (M.S.)

경 력

2018. 3. – 2018. 6. 한국과학기술원 정보보호론 조교

2018. 3. – 2018. 6. 한국과학기술원 세미나 조교

연 구 과 제

2018. 5. – 2018. 10. 암호화폐와 스마트 컨트랙트 응용 시스템 설계 및 보안 취약성 분석 연구

2018. 6. – 2019. 7. QKD 상품성 강화 및 QRNG 취약성 연구(2018)

2018. 8. – 2019. 9. 양자 컴퓨터 환경에서 래티스 문제를 이용한 다자간 인증키 교환 프로토콜 연구 (2018)

2018. 8. – 2019. 12. 양자 난수 생성기의 보안성 및 성능 연구(2018)

2019. 6. – 2019. 10. 대칭키암호 알고리즘 양자안전성 분석 기술 동향 연구(2019)

2019. 8. – 2020. 2. 양자 컴퓨터 환경에서 래티스 문제를 이용한 다자간 인증키 교환 프로토콜 연구 (2019)

연 구 업 적

1. 한성호, **홍동연**, 최낙준, 이나비, 김광조, "(D)PoS 기반 블록체인의 거래 및 합의 방식 분석", *2018 한국정보보호학회 하계학술대회(CISC-S'18), 동신대학교, 나주*, 2018.06.21.
2. **Dongyeon Hong**, and Kwangjo Kim, "Decentralizing the Role of Root Authority in CP-ABE by the Help of Blockchain", *2019 Symposium on Cryptography and Information Security (SCIS 2019)*, Jan., 22-25, 2019, Otsu, Japan.
3. **홍동연**, 김광조, "**BFT를 이용한 악의적인 사용자를 식별하는 그룹키 알고리즘**", *2019 한국정보보호학회 하계학술대회(CISC-S'19), 동명대학교, 부산*, 2019.06.22. - [**행정안전부 장관상 수상**]
4. **홍동연**, 최락용, 김광조, "Newhope의 키 조정 메커니즘을 통한 Apon 등의 프로토콜 구체적 사례 검증", *2019 한국정보보호학회 동계학술대회(CISC-W'19), 중앙대학교, 서울*, 2019.11.30.
5. Rakyong Choi, **Dongyeon Hong**, and Kwangjo Kim, "Implementation of Tree-based Dynamic Group Key Exchange with NewHope", *2020 Symposium on Cryptography and Information Security, (SCIS 2020)*, Jan., 28-31, 2020, Kochi, Japan.
6. Rakyong Choi, **Dongyeon Hong**, and Kwangjo Kim, "Save the Key, Save the Memory: Key-reusable Group Key Exchange from RLWE Assumption", *Post Quantum Cryptography 2020* (submitted at 2019.12.04 and under reviewing)