Lattice Reduction Meets Key-Mismatch: New Misuse Attack on Lattice-Based NIST Candidate KEMs

Ruiqi Mi[1,2], Haodong Jiang[1,2], and Zhenfeng Zhang *[1,2]

[1]Trusted Computing and Information Assurance Laboratory,
Institute of Software,
Email: {ruiqi2017, haodong2020}@iscas.ac.cn
[2]University of Chinese Academy of Sciences,Beijing 100049, China

April 2022

1 Abstract

Resistance to key misuse attacks is a vital property for key encapsulation mechanisms(KEMs) in NIST-PQC standardization process. In key mismatch attack, the adversary recovers reused secret key with the help of an oracle $\mathcal{O}$ that indicates whether the shared key matches or not. Key mismatch attack is more powerful when fewer oracle queries are required. A series of works tried to reduce query times, Qin et al. [AISACRYPT 2021] gave a systematic approach to finding lower bound of oracle queries for a category of KEMs, including NIST's third-round candidate Kyber and Saber.

In this paper, we found the aforementioned bound can be bypassed by combining Qin et al. (AISACRYPT 2021)'s key mismatch attack with a standard lattice attack. In particular, we explicitly build the relationship between the number of queries to the oracle and the bit security of the lattice-based KEMs. Our attack is inspired by the fact that each oracle query reveals partial information of reused secrets, and affects the mean and the covariance parameter of secrets, making the attack on lattice easier. In addition, We quantify such an effect in theory and estimate the security loss for all NIST second-round candidate KEMs.Specifically, Our improved attack reduces the number of queries for Kyber512 by 34% from 1312 queries with bit security 107 to 865 with bit security 32. For Kyber768 and Kyber1024, our improved attack reduces the number of queries by 29% and 27% with bit security is 32.

*Corresponding author: zhenfeng@iscas.ac.cn

2 Introduction

Current Diffie-Hellman key exchange and other widely used factoring or discrete-log-based public key cryptography are no longer safe if quantum computers become practical. According to the roadmap released by the US National Institute and Technology(NIST) and the Department of Homeland Security[1], the transition to post-quantum standards should be finished by 2030, by which cryptographically relevant quantum computers are potentially available.

NIST began the call for post-quantum cryptography algorithms from all over the world in February 2016. In the third round, there are 4 finalists and 5 alternative candidates for Public Key Encryption(PKE) or Key Encapsulation Mechanism(KEM). There are 3 lattice-based KEMs among the 4 finalists.After careful analysis, NIST has selected one finalist and four alternates to move on to the fourth round. Crystals-Kyber[2] is the first PKE/KEM candidate to be standardized, which is based on lattice assumption[3].

Most of the NIST CCA-secure candidates are designed in a similar way: given a chosen-plaintext(CPA) secure construction, such a construction can be converted into a chosen-ciphertext(CCA) secure one by using some transformations like Fujisaki-Okamoto transformation[4]. When the public key is reused, there is no security guarantee on the CPA secure KEMs. If one participant reuses its public key, the adversary can recover the corresponding secret key by comparing if the shared keys between two participants' matches or not. Such attack is called Key Mismatch Attack[5, 6, 7, 8, 9, 10, 11, 12]. Qin et al. [13] give a unified method to find lower bounds for all lattice-based NIST candidate KEMs. They convert the problem into finding an optimal binary recovery tree(BRT). The optimal binary recovery tree can be constructed using the technique of Huffman Coding[14]. Compared to existing results[12, 8], the lower bounds found in [13] improved attacks against Frodo640 and LightSaber with 72 % and 28 % . They also confirm their lower bounds through experiments.

Fujisaki-Okamoto transformation[4] requires re-encryption of the decrypted message to protect the validity of the ciphertext. Such re-encryption is the main cost of the encapsulation process. To improve efficiency, CPA-secure KEMs without FO transformation has been used in designing authenticated key exchanges. In these cases, understanding the effect of key reuse on the concrete security of the scheme is no doubt essential.

The key mismatch attack can also be applied in CCA-secure KEMs. Ravi et al. [15] showed that the restrictions of FO transform can be bypassed with the help of side-channel information. Ravi's side-channel attack(SCA) utilizes Welch's test-based template approach [16]. Such an attack consists of two stages: pre-processing and template matching operation. However, Ravi's attack has to select parameters in a brute-force way. The optimal BRT approach can be directly applied in Ravi's attack. The needed number of queries is reduced significantly. Take Kyber512 as an example, Ravi's CCA attack requires 2560 queries. By adopting the optimal BRT approach, only 1183 times of queries is required when the coefficients of reused secret s_A are drawn from $[-2, 2]$.

Side-channel information from oracle queries can be integrated as side infor-

mation into a standard lattice attack, thus reducing the cost of attacking the underlying Learning with Errors problem.

Suppose Alice reuses her public key P_A. The adversary $\mathcal{A}$'s goal is to recover each coefficient of Alice's secret key $\boldsymbol{s_A}$. Set $H(\boldsymbol{S})$ the Shannon Entropy for S. According to Qin's analysis, A single coefficient block s_i of $\boldsymbol{s_A}$ can be recovered by at most $H(\boldsymbol{S})+1$ queries to the oracle. Such queries lead to a perfect hint in the form of $< \boldsymbol{s}, \boldsymbol{v} >= s_i$, where $\boldsymbol{v}$ is an all-zero vector except the i-th coefficient is 1. Integrating such a hint into the lattice reduces the dimension and increases the volume of the lattice, reducing the cost of standard lattice attack.

The above motivates the focus of this work: can we estimate the concrete relationship between the number of queries and by how much does such queries reduce the cost of lattice attack? If the cost of standard lattice attack is reduced to an acceptable range, we can bypass the lower bound of query numbers in key reuse attack.

Contributions We build the relationship between the number of queries and the concrete security loss in Section 5. Our theorem can be applied to all lattice-based NIST candidate KEMs since the underlying attacks are unified. We found that for at most $H(\boldsymbol{S}) + 1$ queries to the oracle, the dimension of the lattice is decreased by one and the volume is increased by a factor $\sqrt{1 + s_i^2}$, leading to lower cost of standard lattice attack.

We test the concrete bit security for lattice-based NIST round 2 KEM candidates under key mismatch attack: Kyber[2], FrodoKEM[17], Saber[18], NewHope[19],LAC[20],ThreeBear[21],Round5[22]. Take Kyber512 as an example, the classical bit security is reduced from 119 to 64 after querying the oracle for 533 times, and further reduced to 32 after querying the oracle for 867 times. Compared to the number of queries needed without lattice technique, we reduced the number of queries by at least 34 %.

Lattice techniques can also be applied to side-channel attacks against CCA-secure NIST candidate KEMs in a similar way. [15] showed that with the help of side-channel information, the adversary can also launch chosen ciphertext attack on CCA-secure NIST candidate KEMs in a direct way to [13]. The only difference between CPA and CCA versions of the key mismatch attack is that the adversary can physically access devices performing decapsulation to know whether the shared messages match or not.

According to our experimental results, we found that larger ranges of coefficients require more number of queries to reach the same bit security. On the other side, encoding/decoding several coefficients at one time reduces the number of queries and block sizes, lowering the cost of lattice attacks.

Organizations We start with some preliminaries in Section 2. In particular, we introduced the model of key mismatch attack and how to estimate exact security under standard lattice attacks. Section 3 gives a brief summary of key mismatch attack and optimal BRT approach. Section IV analyzes the relationship between bit security and the number of oracle queries in theory. Section V describes the experimental results for bit security and oracle queries for each NIST second round KEM candidates.

Independent and Concurrent Work Very recently, Guo and Mårtensson

[23] showed an improved key mismatch attack that recovers multiple secret coefficients in a parallel way. The comparisons are summarized below:

1 Guo and Mårtensson showed how to recover partial information of multiple secret entries in each oracle call. The adversary split the two-dimensional plane for two secret coefficients and decides from the mismatch oracle call which part the two coefficients belong to. Compared to the lower bound given in [13], the attack given in [23] reduces the number of queries needed by 0.08%,10.6%, 10.6% for Kyber512, Kyber768, Kyber1024, and 3.4%, 5.01%, 8.1% for LightSaber, Saber, FireSaber.

2 In the discussion part, they give a rough estimation of the query sample complexity for Kyber and Saber when post-processing is allowed. They employ the lattice estimator given in [24]. They did not give concrete relationship between the query times and the geometry of the lattice in theory.

In our work, we investigate the form of side information the adversary get from oracle queries and the volume and dimension change after integrating such information in Section 5. Besides, we also found that removing some special short vectors can further reduces the bit security after each oracle call. We give a quantitative and detailed analysis between query times and bit security in theory. Besides,we applied our theory to all NIST second-round KEM candidates except NTRU[25] and NTRU Prime[26].

3 Preliminaries

3.1 CPA-secure lattice-based KEMs and its construction

Definition 1 (KEM)**.** *A Key-Encapsulation Mechanism(KEM) is a tuple of three algorithms:*

Gen: Gen is a probabilistic algorithm that takes a security parameter and returns a keypair $pk, sk \in \mathcal{PK} \times \mathcal{SK}$*.*

Enc: Enc is a probabilistic algorithm that takes a public key pk and returns a ciphertext c from a ciphertext space C and a shared secret ss from a secret space SS.

Dec: Dec takes a secret key $sk \in SK$ *and a ciphertext* $c \in C$ *and returns a shared secret* $ss \in SS$*.*

Definition 2 (IND-CPA)**.** *We say that a KEM* KEM *offers IND-indistinguishability under Chosen Plaintext Attacks (IND-CPA) if and only if:*

$$\begin{aligned}\forall \mathcal{A} \in \mathrm{QPT}, \lambda \in \mathbb{N} : &\left| \Pr\left[\mathrm{Exp}^{\mathrm{IND-CPA}}_{KEM,\mathcal{A},q}\left(1^{\lambda}\right) = 1\right] - \frac{1}{2}\right| \\ =: &\,\mathrm{Adv}^{\mathrm{IND-CPA}}_{KEM,\mathcal{A},q}\left(1^{\lambda}\right) \leq \mathrm{negl}(\lambda)\end{aligned}$$

where $\mathrm{Exp}^{IND\text{-}CPA}_{KEM,\mathcal{A},q}$ *is defined in Experiment 1.*

The design of the CPA-secure KEMs can be roughly divided into two categories: The first category follows the work of Regev[27], Lyubashevsky-Peikert-Regev[28] and the lattice-based key exchange scheme proposed by Ding, Xie

Experiment 1: $Exp^{IND-CPA}_{KEM,\mathcal{A},q}$

$pk, sk := KEM.gen(1^\lambda)$;
$c^*, k_0 := KEM.enc(pk)$;
$queries := 0$;
$k_1 \xleftarrow{\$} \{0,1\}^\lambda$;
$b \longleftarrow \{0,1\}$;
Oracle: Enc(m)
    $queries+ = 1$;
    $abort\_if(queries > q \vee m = m^*)$;
    return $KEM.Enc(pk, m)$;
$b' := \mathcal{A}^{Enc}(c^*, k_b)$;
Result: $b = b'$

and Lin[29]. The other is NTRU[30] and NTRU Prime[26]. Many NIST candidate KEMs are designed as the first category:FrodoKEM[17], NewHope[19], LAC[20], Kyber[2], ThreeBears[21], Round5[22], Saber[18]. In Figure 1 we give the meta structure of CPA-secure KEMs in the fist category.

Figure 1: The structure of CPA-secure LWE-based KEM

Alice		**Bob**
$\triangleright KEM.Gen()$		
$a \leftarrow\$ \mathcal{R}$		
$s_A, e_A \leftarrow\$ \chi$		$m \leftarrow\$ \{0,1\}^\lambda$
$P_A \leftarrow a \circ s_A + e_A$		$\triangleright KEM.Enc(P_A, m)$
$Output \quad (P_A, s_A)$		$a \leftarrow\$ \mathcal{R}, s_B, e_B, e'_B \leftarrow\$ \chi$
	$P_A \longrightarrow$	$P_B \leftarrow s_B \circ P_A + e'_B + e_B$
		$k \leftarrow Encode(m)$
		$c \leftarrow s_B \circ P_A + e'_B + k$
$\triangleright KEM.Dec(P_B, \bar{c}, s_A)$		$\bar{c} \leftarrow Compress(c)$
$c' \leftarrow Decompress(\bar{c}))$		$Output : (P_B, \bar{c})$
$k' = c' - P_B \circ s_A$		$K_B \leftarrow H(m\|(P_B, \bar{c}))$
$m \leftarrow Decode(k')$	$\longleftarrow (P_B, \bar{c})$	
$Output : m'$		
$K_A \leftarrow H(m'\|(P_B, \bar{c}))$		

3.2 Model of Key Mismatch Attacks

Suppose Alice reuses her public key P_A. The adversary $\mathcal{A}$ has access to an oracle $\mathcal{O}_s$ that decides if the two shared keys match or not. Then adversary $\mathcal{A}$ can impersonate Bob to recover Alice's secret key s_A.

The oracle $\mathcal{O}_s$ simulates Alice's $KEM.Dec$ part as shown in Algorithm 3. The input to the oracle $\mathcal{O}_s$ includes the public key of Bob P_B, the ciphertext $\bar{c}$ chosen by the adversary $\mathcal{A}$ and the shared key K_B. For each query, the oracle calls the function $Dec(P_B)$ and gets the returned shared key K_A. If K_A and K_B matches, $\mathcal{O}_s$ outputs 1 , otherwise $\mathcal{O}_s$ returns 0.

Algorithm 2: Key Mismatch Attack

Input: Alice's P_A and Oracle $\mathcal{O}_s$
Output: 0 or 1
$s'_A \leftarrow \mathcal{A}^{\mathcal{O}_s}(P_A)$;
if $s'_A = s_A$ **then**
  return 1
else
  return 0
end

Algorithm 3: oracle $\mathcal{O}_s(P)$

Input: $P := (P_B, \bar{c}, K_B)$
Output: 0 or 1
$K_A \leftarrow KEM.Dec(P_B, \bar{c})$;
if $K_A = K_B$ **then**
  return 1
else
  return 0
end

3.3 Lattice

A lattice is a discrete additive subgroup of $\mathbb{R}^m$,denoted as Λ. Lattice Λ is generated by a set of linearly independent *basis* $\{\boldsymbol{b_j}\} \subset \mathbb{R}^m$,that is $\Lambda := \{\Sigma_j z_j \boldsymbol{b_j} : z_j \in \mathbb{Z}\}$.The i-th *successive minimum* of a lattice,$\lambda_i(L)$,is the radius of the smallest ball centered at the origin containing at least i linearly independent lattice vectors.

We denote the dimension of lattice Λ as m and the rank as n. If $n = m$,the lattice is full rank. Matrix $\boldsymbol{B}$ having all basis vectors as rows can be called a *basis*. The volume of the lattice is defined as $Vol(\Lambda) := \sqrt{det\,(\boldsymbol{B}\boldsymbol{B}^{\boldsymbol{T}})}$. The dual lattice of Λ in $\mathbb{R}^n$ is defined as:

$$\Lambda^* := \{\boldsymbol{y} \in Span(\boldsymbol{B}) \mid \forall x \in \Lambda, \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}\} \tag{1}$$

The dual of Λ^* is Λ,$(\Lambda^*)^* = \Lambda$, and $\mathrm{Vol}\,(\Lambda^*) = 1/\,\mathrm{Vol}(\Lambda)$.

Lemma 1. *[31, proposition 1.3.4] Let Λ be a lattice and let F be a subspace of $\mathbb{R}^n$. If $\Lambda \cap F$ is a lattice, then the dual of $\Lambda \cap F$ is the orthogonal projection onto F of the dual of Λ. In other words, each element of Λ^* is multiplied by the projection matrix Π_F:*

$$(\Lambda \cap F)^* = \Lambda^* \cdot \Pi_F \tag{2}$$

Lemma 2. *[31, Proposition 1.2.9] Let Λ be a lattice in $\mathbb{R}^n$,such that $\Lambda \cap F$ is a lattice and let be the orthogonal projection onto $F^\perp$. Then*

$$\mathrm{Vol}\left(\Lambda \cdot \Pi_F^\perp\right) = \mathrm{Vol}(\Lambda)\left(\mathrm{Vol}(\Lambda \cap F)^{-1}\right) \tag{3}$$

Definition 3 (**Primitive Vectors**). *A set of vector $\boldsymbol{y_1}, \ldots, \boldsymbol{y_k} \in \Lambda$ is said primitive with respect to Λ if $\Lambda \cap Span(\boldsymbol{y_1}, \ldots, \boldsymbol{y_k})$ is equal to the lattice generated by $\boldsymbol{y_1}, \ldots, \boldsymbol{y_k}$.Equivalently,it is primitive if it can be extended to a basis of Λ. If $k = 1, \boldsymbol{y_1}$, this is equivalent to $\boldsymbol{y_1}/i \notin \Lambda$ for any integer $i \geq 2$.*

To predict the hardness of the lattice reduction on altered instances after several queries to oracle $\mathcal{O}_s$, we need to compute the volume of the final transformed lattice. [31] gives a highly efficient way to predict the volume. The volume of the final lattice is updated given the volume of the previous lattice and the hint constructed through queries.

Lemma 3. *[31, Lemma 12] Given a lattice Λ with volume $Vol(\Lambda)$ and a primitive vector $\boldsymbol{v}$ with respect to Λ^*.Let $\boldsymbol{v}^\perp$ denote subspace orthogonal to v. Then $\Lambda \cap \boldsymbol{v}^\perp$ is a lattice with volume $Vol\left(\Lambda \cap \boldsymbol{v}^\perp\right) = \|\boldsymbol{v}\| \cdot Vol(\Lambda)$.*

Fact 1 (**Volume of a projected lattice**). *Let Λ be a lattice, $\boldsymbol{v}$ be a primitive vector of Λ. Let $\Lambda' = \Lambda \cdot \Pi_{\boldsymbol{v}}^\perp$ be a sublattice of Λ. Then $\mathrm{Vol}(\Lambda') = Vol(\Lambda)/\|\boldsymbol{v}\|$.More generally, if $\boldsymbol{V}$ is a primitive set of vectors of Λ, then $\Lambda' = \Lambda \cdot \Pi_{\boldsymbol{v}}^\perp$ has volume $\mathrm{Vol}(\Lambda') = \mathrm{Vol}(\Lambda)/\sqrt{det\,(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}})}$.*

Fact 2 (**Lattice volume under linear transformation**). *Let Λ be a lattice in $\mathbb{R}^n$, and $M \in \mathbb{R}^{n \times n}$ a matrix such that $\boldsymbol{M} = Span(\Lambda)^\perp$.Then we have $\mathrm{Vol}(\Lambda \cdot \boldsymbol{M}) = \det(\boldsymbol{M})\,\mathrm{Vol}(\Lambda)$.*

Definition 4 (**search-LWE problem with short secrets**). *Let n, m, q be positive integers, and let χ be a distribution over $\mathbb{Z}$. The search LWE problem (with short secrets) for parameters (n, m, q, χ) is:*

***Given** the pair $(\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}, \boldsymbol{b} = \boldsymbol{z}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{e} \in \mathbb{Z}_q^m)$ where:*

1. $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$ is sampled uniformly at random.

2. $\boldsymbol{z} \leftarrow \chi^n$, and $\boldsymbol{e} \leftarrow \chi^m$ are sampled with independent and identically distributed coefficients following the distribution χ.

***Find** $\boldsymbol{z}$.*

The complexity of solving (search-)LWE against primal attack consists of viewing the LWE as an instance of (Distorted-)Bounded Distance Decoding problem, reducing DBDD to uSVP(via Kannan's Embedding [32], and finally applying lattice reduction algorithm to solve the uSVP instance [33]. DBDD accounts for potential distortion in the distribution of the secret noise vector that is to be recovered, and the secret noise vector is found at a lower cost.

Definition 5 (γ**-uSVP**). *given a lattice* Λ *such that* $\lambda_2(\Lambda) > \gamma\lambda_1(\Lambda)$,*find a shortest nonzero vector in* Λ.

Definition 6 (**Distorted Bounded Distance Decoding Problem,DBDD**). *Let* $\Lambda \subset R^d$ *be a lattice,*$\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ *be a symmetric matrix and* $\boldsymbol{\mu} \in Span(\Lambda) \subset \mathbb{R}^d$ *such that* $Span(\boldsymbol{\Sigma}) \subsetneq Span\left(\boldsymbol{\Sigma} + \boldsymbol{\mu}^T\boldsymbol{\mu}\right) = Span(\Lambda)$

The Distorted Bounded Distance Decoding Problem $\boldsymbol{DBDD}_{\Lambda,\boldsymbol{\mu},\boldsymbol{\Sigma}}$ *is:*

Given $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ *and a basis of* Λ.

Find *the unique vector* $\boldsymbol{x} \in \Lambda \cap E(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Where $E(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *denotes the ellipsoid*

$$E(\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \left\{\boldsymbol{x} \in \boldsymbol{\mu} + Span(\boldsymbol{\Sigma}) \mid (\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{\Sigma}^{\sim} \cdot (\boldsymbol{x} - \boldsymbol{\mu})^T \leq rank(\boldsymbol{\Sigma})\right\} \quad (4)$$

The triple $I = (\Lambda, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ *will be referred to as the instance of the* $DBDD_{\Lambda,\boldsymbol{\mu},\boldsymbol{\Sigma}}$ *problem .*

Special Cases of DBDD DBDD problem is a generalization of BDD problem. When $\boldsymbol{\Sigma} = \boldsymbol{I_d}, DBDD_{\Lambda,\boldsymbol{\mu},\boldsymbol{I_d}}$ is BDD instance with shifted hyperball with center $\boldsymbol{\mu}$.In addition, if we have $\boldsymbol{\mu} = \boldsymbol{0}, DBDD_{\Lambda,\boldsymbol{0},\boldsymbol{I_d}}$ becomes a uSVP instance on Λ.

3.4 Embedding LWE into DBDD

Given a lattice $\Lambda = \left\{(\boldsymbol{x}, \boldsymbol{y}, w) \mid \boldsymbol{x} + \boldsymbol{y}A^T - \boldsymbol{b}w = 0 \bmod q\right\}$ .The lattice Λ is of full rank in $\mathbb{R}^d$ and has volume q^m.Λ's lattice basis is given by the row vectors of

$$\begin{bmatrix} q\boldsymbol{I_m} & 0 & 0 \\ \boldsymbol{A^T} & -\boldsymbol{I_n} & 0 \\ \boldsymbol{b} & 0 & 1 \end{bmatrix} \quad (5)$$

Λ has a short vector $\bar{s} := (\boldsymbol{e}, \boldsymbol{z}, 1)$.

Given an LWE instance with solution $\boldsymbol{s} := (\boldsymbol{e}, \boldsymbol{z})$.We denote the average and variance of the LWE distribution χ as μ_χ and σ_χ^2. Such LWE instance can be converted to a $DBDD_{\Lambda,\mu,\Sigma}$ instance with $\boldsymbol{\mu} = [\mu_\chi, \cdots, \mu_\chi, 1]$,$\Sigma = \begin{bmatrix} \sigma_\chi^2\boldsymbol{I}_{m+n} & 0 \\ 0 & 0 \end{bmatrix}$.

3.5 Converting DBDD to uSVP

We give the definition of restricted inverse(pseudoinverse) and restricted determinant of a given matrix $\boldsymbol{\Sigma}$. Such definition gives invertibility property to any given matrix by adding the orthogonal parts and then removing them.

Definition 7 (**Restricted Inverse**). *Given a symmetric matrix* $\boldsymbol{\Sigma}$. *The orthogonal projection matrix of* $\boldsymbol{\Sigma}$ *is* $\Pi_{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}^T \cdot (\boldsymbol{\Sigma} \cdot \boldsymbol{\Sigma}^T)^{-1} \cdot \boldsymbol{\Sigma}$. *The projection*

orthogonal to $\mathbf{\Pi_\Sigma}$ *is denoted as* $\mathbf{\Pi_\Sigma^\perp} := \boldsymbol{I}_d - \mathbf{\Pi_\Sigma}$. *The restricted inverse of* $\mathbf{\Sigma}$ *is defined as*

$$\mathbf{\Sigma}^\sim := (\mathbf{\Sigma} + \mathbf{\Pi_\Sigma^\perp})^{-1} - \mathbf{\Pi_\Sigma^\perp} \tag{6}$$

Denote the restricted inverse of $\mathbf{\Sigma}$ *as* $\mathbf{\Sigma}^\sim$. *The restricted inverse satisfies* $Span(\mathbf{\Sigma}^\sim) = Span(\mathbf{\Sigma})$ *and* $\mathbf{\Sigma} \cdot \mathbf{\Sigma}^\sim = \mathbf{\Pi_\Sigma}$.

The restricted determinant of matrix Σ *is defined as* $rdet(\mathbf{\Sigma}) := det(\mathbf{\Sigma} + \mathbf{\Pi_\Sigma^\perp})$.

The following theorem explains how to convert a DBDD instance $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ into a uSVP instance $uSVP_{\Lambda,\boldsymbol{M}}$.

Theorem 1. *[31, section 3.3] Given a DBDD instance* $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$. *The relationship between the solution* $\boldsymbol{x}$ *to the* $uSVP_{\Lambda,\boldsymbol{M}}$ *problem and the solution* $\boldsymbol{x'}$ *to the* $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ *is* $\boldsymbol{x'} = \boldsymbol{x}\boldsymbol{M}^\sim$ *with* $\mathbf{\Sigma'} := \mathbf{\Sigma} + \boldsymbol{\mu}^T \cdot \boldsymbol{\mu}$, $\boldsymbol{M} := \left(\sqrt{\mathbf{\Sigma'}}\right)^\sim$.

Remark. *The transformation can be described in three steps. First,* $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ *is contained in a homogenized and centered one* $DBDD_{\Lambda,\mathbf{0},\mathbf{\Sigma'}}$ *with* $\mathbf{\Sigma'} := \mathbf{\Sigma} + \boldsymbol{\mu}^T \cdot \boldsymbol{\mu}$. *Second,* $DBDD_{\Lambda,\mathbf{0},\mathbf{\Sigma'}}$ *can be transformed into isotropized* $DBDD_{\Lambda\cdot\boldsymbol{M},\mathbf{0},\mathbf{\Pi_\Lambda}}$ *by multiplying every element of the lattice with the restricted inverse of* $\sqrt{\mathbf{\Sigma'}}$, *which is denoted as* $\sqrt{\mathbf{\Sigma'}}^\sim$. *The new covariance matrix is* $\mathbf{\Sigma''} = \sqrt{\mathbf{\Sigma'}}^\sim \cdot \mathbf{\Sigma'} \cdot \sqrt{\mathbf{\Sigma'}}^\sim$. *The relationship between the solution* $\boldsymbol{x}$ *to the* $uSVP_{\Lambda,\boldsymbol{M}}$ *problem and the solution* $\boldsymbol{x'}$ *to the* $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ *is* $\boldsymbol{x'} = x \cdot \boldsymbol{M}^\sim$ *with* $\boldsymbol{M} := \left(\sqrt{\mathbf{\Sigma'}}\right)^\sim$. *We recommend our reader to read section 3.3 of [31] for more detail.*

3.6 Predicting concrete hardness of a uSVP instance

The attack on a uSVP instance consists of applying BKZ algorithm with block size β on the uSVP lattice Λ for an appropriate block size parameter β. For lattice Λ of dimension $dim(\Lambda)$, BKZ algorithm with block size β can solve a $uSVP_\Lambda$ instance with secret s when:

$$\sqrt{\beta} \leq \delta_\beta^{2\beta - \dim(\Lambda) - 1} \cdot \mathrm{Vol}(\Lambda)^{1/\dim(\Lambda)} \tag{7}$$

The above predictions for solving uSVP instances using BKZ is given in [34, 35].

3.7 Perfect Hints On the secret

Definition 8. *A perfect hint on the secret* $\boldsymbol{s}$ *is the knowledge of* $\boldsymbol{v} \in \mathbb{Z}^{d-1}$ *and* $l \in \mathbb{Z}$ *such that* $\langle \boldsymbol{s}, \boldsymbol{v} \rangle = l$.

Such perfect hint can be integrated into a DBDD instance. Extend the solution $s = (e, z)$ to $\bar{s} = (e, z, 1)$. Let $\bar{\boldsymbol{v}} := (\boldsymbol{v}; -l)$. The adversary $\mathcal{A}$ can integrate hint v by modifying $DBDD_{\Lambda,\boldsymbol{\mu},\mathbf{\Sigma}}$ to $DBDD_{\Lambda',\boldsymbol{\mu'},\mathbf{\Sigma'}}$,where:

$$\begin{aligned}
\Lambda' &= \Lambda \cap \left\{ x \in \mathbb{Z}^d \mid \langle \boldsymbol{x}, \bar{\boldsymbol{v}} \rangle = 0 \right\} \\
\boldsymbol{\Sigma'} &= \boldsymbol{\Sigma} - \frac{(\bar{\boldsymbol{v}}\boldsymbol{\Sigma})^T \bar{\boldsymbol{v}}\boldsymbol{\Sigma}}{\bar{\boldsymbol{v}}\boldsymbol{\Sigma}\bar{\boldsymbol{v}}^{\boldsymbol{T}}} \\
\boldsymbol{\mu'} &= \boldsymbol{\mu} - \frac{\langle \bar{\boldsymbol{v}}, \boldsymbol{\mu} \rangle}{\bar{\boldsymbol{v}}\boldsymbol{\Sigma}\bar{\boldsymbol{v}}^{\boldsymbol{T}}} \bar{\boldsymbol{v}}\boldsymbol{\Sigma}
\end{aligned} \tag{8}$$

The derivation of 8 and the computation of new lattice Λ is given in Section 4.1 [31] . A perfect hint is quite strong in terms of additional knowledge. The adversary $\mathcal{A}$ can construct perfect hints with the help of oracle $\mathcal{O}_s$ in section 3.2. We will give a detailed description about how to construct such hints in Section 5.

4 Theoretical and Practical Lower Bound For Key Mismatch Attack

When users reuse their secret key, the CPA-secure KEMs are vulnerable to chosen-ciphertext attacks. When the adversary $\mathcal{A}$ tries to recover Alice's secret key $\boldsymbol{s_A}$, $\mathcal{A}$ craft special ciphertexts to narrow the possible range of coefficient block $\boldsymbol{s_i}$(e.g. $[-3, 3]$ in Kyber512) with the help of oracle $\mathcal{O}_s$'s response. Qin et al. gives a theoretical lower bound for a certain type of key reuse attack using (nearly) **Optimal Binary Search Tree**[36] and **Huffman Coding**[14]. In Qin's attack [13], the range of $\boldsymbol{s_i'}s$ coefficient block is divided in half for each query, thus we will refer to Qin's attack as key mismatch attack in the following parts.

Let $\boldsymbol{S} = \{\boldsymbol{S_0}, \boldsymbol{S_1}, ..., \boldsymbol{S_n}\}$ be the set of all possible values for one coefficient block(let $S' = \{-1, 0, 1\}$.If there are no Encode/Decode functions,$\boldsymbol{S} = S'$.If the scheme use $D-v$ lattice to encode, then $\boldsymbol{S} = \{(s_1, s_2, ..., s_v) | s_1, s_2, ..., s_v \in S'\}$).

For any coefficient block $\boldsymbol{s_A^b}$ of $\boldsymbol{s_A}$, let P_i be the probability that $s_A^b = S_i$,where $s_A \xleftarrow{\$} \chi$.That is,$P_i = Prob(\boldsymbol{s_A^b} = \boldsymbol{S_i} | \boldsymbol{s_A} \leftarrow \chi)$. We assume without loss of generality that $P_0 \geq P_1 \geq ... \geq P_{n-1}$. It holds that $\sum_{i=0}^{n-1} P_i = 1$.

To recover a coefficient block in key mismatch attack, adversary $\mathcal{A}$ needs to query the oracle O_s with properly crafted parameters for several times. We denote the required number of queries needed as $\boldsymbol{Q_i}$ when the coefficient block is exactly $\boldsymbol{S_i}$. The expectation of number of queries to recover one coefficient block is $E_{\mathcal{A}}(\boldsymbol{S}) = \sum_{i=0}^{n-1} P_i \boldsymbol{Q_i}$.Finding a lower bound in key mismatch attack is to find the minimum value of $E_{\mathcal{A}}(\boldsymbol{S})$ over all possible attack strategies under the attack model in Section 3.2.

The key idea of minimizing $E_{\mathcal{A}}(\boldsymbol{S})$ is to associate every attack with a binary recovery tree(BRT). A binary recovery tree is a binary tree with a root node and n leaf nodes, each leaf node corresponds to a $\boldsymbol{S_i}$. For every node that has child nodes, we denote the left child node by 1 and its right child node by 0.

The adversary get a unique binary sequence $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ from the Oracle $\mathcal{O}_s$ to recover any coefficient block. Each coefficient block $\hat{\boldsymbol{S}}_{\boldsymbol{i}}$ corresponds to different binary sequence $\hat{\boldsymbol{s}}_i$. A binary search tree can be constructed in a natural way that for

every i, the binary string consisting of the nodes from the root to the leaf $\hat{\boldsymbol{S}}_{\boldsymbol{i}}$ is the binary sequence $\hat{\boldsymbol{s}}_i$. The length of $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ also means the depth $depth_{T_{\mathcal{A}}}(\boldsymbol{S_i})$ of the BRT. The expectation of query numbers can be represented as $E_{\mathcal{A}}(\boldsymbol{S}) = \sum_{i=0}^{n-1} P_i \cdot depth_{T_{\mathcal{A}}}(\boldsymbol{S_i})$.

If we enlarge the set of BRTs to all possible BRTs, the problem of finding the lower bound of $\boldsymbol{E_A(S)}$ becomes the problem of finding an optimal BRT to minimize $\boldsymbol{E(S)}$. Huffman coding[14] is one of the optimal way to find the optimal BRT. It combines two $\boldsymbol{S_i}$ of the lowest probabilities in each step. Thus the problem becomes finding BRT with $n-1$ weights with $\{P_0, P_1, ..., P_{n-2} + P_{n-1}\}$. The optimal BRT can be constructed by repeating such process, and we can get the $minE(S)$ within $O(nlogn)$.

The following theorem gives the minimum value of $E(S)$ calculated by the optimal BRT.

Theorem 2. *[13] In key mismatch attack model described in section 3.2, the methods described above gives bounds for minimum average number of queries in launching the key mismatch attacks. Given $\boldsymbol{S} = \{\boldsymbol{S_0}, \boldsymbol{S_1}, ..., \boldsymbol{S_{n-1}}\}$ and its corresponding probabilities $\{P_0, P_1, ..., P_{n-1}\}$ in each lattice-based KEM, min $E(\boldsymbol{S})$ calculated by the optimal BRT is a lower bound for the minimum average number of queries . Moreover,set $H(\boldsymbol{S})$ the Shannon entropy for $\boldsymbol{S}$, then we have*

$$H(\boldsymbol{S}) \leq min\ E(\boldsymbol{S}) \leq H(\boldsymbol{S})\ +\ 1 \tag{9}$$

Key Reuse Attack for Kyber The main attack procedure proposed in [13] for Kyber is crafting ciphertext adaptively. The adversary selects proper ciphertext $ct = (\boldsymbol{c_1}, \boldsymbol{c_2})$ as inputs to $\mathcal{O}_s$. Then, the attacker is able to recover s from the oracle response $\hat{s}$. The recovery of each coefficient is mutually independent. We give the approach to recover the first coefficient block $\boldsymbol{s_0}[0]$ of $\boldsymbol{s}$, other coefficient blocks can be recovered similarly.

The attacker selects $\boldsymbol{m} = (1, 0, ..., 0)$ and $\boldsymbol{u} = (\boldsymbol{u_0}, \boldsymbol{u_1}), \boldsymbol{u_0} = ([\frac{q}{16}], 0, ..., 0), \boldsymbol{u_1} = \boldsymbol{0}$. Then the attack query the oracle $\mathcal{O}_s$ with $\boldsymbol{ct} = (\boldsymbol{c_1}, \boldsymbol{c_2})$. $\boldsymbol{ct}$ is then used to generate $\boldsymbol{u} = \boldsymbol{Decomp}_q(\boldsymbol{c_1}, d_u), \boldsymbol{v} = \boldsymbol{Decomp}_q(\boldsymbol{c_2}, d_v)$. Thus, the attacker constructs a relationship between $\boldsymbol{m'}[0]$ and $\boldsymbol{s_0}[0]$ after decryption as $\boldsymbol{m'}[0] = [\frac{2}{q}([\frac{q}{16}h] - \boldsymbol{s_0}[0][\frac{q}{16}])]\ mod\ 2$, where h is a parameter chosen by the attacker.Let $h = 4$, if $\boldsymbol{s_A^T}[0] \in [0, 3]$,then the oracle will output 0. Otherwise the oracle will output 1.

The attacker could adaptively choose h to recover $\boldsymbol{s_0}[0]$ based on the sequence $\hat{s}$ from oracle $\mathcal{O}_s$. If the attacker uses well-selected h, he could recover $\boldsymbol{s_0}[0]$ with as few queries as possible. With the help of the optimal binary recovery tree, the adversary divides the range of coefficient block in half each time and tries to recover $\boldsymbol{S_i}$ with the biggest probability as soon as possible. We list the selection of h in Table 1.In such an key mismatch attack,each query divides the possible range of $\boldsymbol{s_A}[0]$ into (nearly) half. [13] gives the selection of h and the corresponding changes of states in section 4.1.

Table 1: The choice of h and the States for Kyber512

	State1	**State2**	**State3**	**State4**	**State5**	**State6**
h	**4**	**3**	**9**	**12**	**13**	**7**
$\mathcal{O}_s \to 0$	**State4**	$\boldsymbol{s_A}[0] = -1$	$\boldsymbol{s_A}[0] = -3$	$\boldsymbol{s_A}[0] = 0$	$\boldsymbol{s_A}[0] = 1$	$\boldsymbol{s_A}[0] = 3$
$\mathcal{O}_s \to 1$	**State2**	**State3**	$\boldsymbol{s_A}[0] = -2$	**State5**	**State6**	$\boldsymbol{s_A}[0] = 2$

According to Theorem 2,The lower bound for Kyber512, Kyber768 and Kyber1024 in theory is 1216, 1632, 2176. The expectation of queries needed to recover a single coefficient in $\boldsymbol{s_A}$ is $\frac{5}{16} \times 2 + \frac{15}{64} \times (3+2) + \frac{3}{32} \times (4+3) + \frac{1}{32} \times 3 = 2.56$. The average number of queries needed in key mismatch attack for Kyber512, Kyber768 and Kyber1024 in theory is 1312, 1774, 2365. The gap is less than 9%. Besides, Qin et al. also did an experiment to verify their theory. The experiment result shows that the number of queires is 1311,1777,2368 separately.

Key Reuse Attack for Saber There are three versions of Saber in NIST third round's submission: LightSaber($\boldsymbol{S_i} \in [-5, 5]$),Saber($\boldsymbol{S_i} \in [-4, 4]$) and FireSaber($\boldsymbol{S_i} \in [-3, 3]$). The key mismatch attack for them are similar. The adversary chooses $P_B = h$ and $c_m = k$ and query $\mathcal{O}_s$, then S_i can be divided into two parts based on the response of $\mathcal{O}_s$. Repeat this process will recover $\boldsymbol{S_i}$. The selection of P_B and c_m is given in Table4, section 4.2[13].

The lower bound for LightSaber, Saber and FireSaber is 1412, 1986, 2432 according to theorem 2. The average number of queries needed for LightSaber, Saber, and FireSaber in key mismatch attack described above is 1460, 2091, 2642. The gap is less than 8%. Besides, Qin et al. also did an experiment to verify their theory, the experiment result shows that the number of queries is 1476,2095,2622 separately.

Key Reuse Attack for Frodo There are three versions of Frodo in NIST second round's submission: Frodo640 ($\boldsymbol{S_i} \in [-12, 12]$),Frodo976 ($\boldsymbol{S_i} \in [-10, 10]$) and Frodo1314($\boldsymbol{S_i} \in [-6, 6]$). The selection of parameter h_i is given in Table 5,section 4.3[13].The attacker selects $h = h_1$ first, and divide the range of S_i into two parts with equal size based on O_s's return.If O_s return 0, $\boldsymbol{s_A} \in \{\boldsymbol{S_0}, \boldsymbol{S_2}, \boldsymbol{S_4}, \boldsymbol{S_6}, \boldsymbol{S_8}, \boldsymbol{S_{10}}, \boldsymbol{S_{12}}\}$,otherwise $\boldsymbol{s_A}[0] \in \{\boldsymbol{S_1}, \boldsymbol{S_3}, \boldsymbol{S_5}, \boldsymbol{S_7}, S_9, \boldsymbol{S_{11}}\}$.In the second step, If $\mathcal{O}_s$ returns 0,the attacker set $h = h_1$, then set $h = h_2$,if $\mathcal{O}_s$ returns 0, we have $\boldsymbol{s_A}[0] = \boldsymbol{S_0}$. Otherwise $\boldsymbol{s_A} \in \{\boldsymbol{S_2}, \boldsymbol{S_4}, \boldsymbol{S_6}, \boldsymbol{S_8}, \boldsymbol{S_{10}}, \boldsymbol{S_{12}}\}$. In step 3, the adversary select parameter $h = h_2, ..., h_7$ until we know the certain value of$\boldsymbol{s_A}[0] \in \{\boldsymbol{S_2}, \boldsymbol{S_4}, \boldsymbol{S_6}, \boldsymbol{S_8}, \boldsymbol{S_{10}}, \boldsymbol{S_{12}}\}$. When $\boldsymbol{s_A}[0] \in \{\boldsymbol{S_1}, \boldsymbol{S_3}, \boldsymbol{S_5}, \boldsymbol{S_7}, \boldsymbol{S_9}, \boldsymbol{S_{11}}\}$,the attacker set $h = h_8 ... h_{12}$ and repeat step 2 and step 3 to determine $\boldsymbol{s_A}[0]$.

The theoretical lower bound for Frodo640, Frodo976 and Frodo1344 is 18227, 25796, 27973. The average number of queries needed for Frodo640,Frodo976 and Frodo1344 is 18329, 26000, 29353 . Besides, Qin et al. also did an experiment to verify their theory, the experiment result shows that the number of queires is 18360,26078,29378 separately.

Key Reuse Attack for other NIST Lattice-Based KEMs

There are also improved key mismatch attacks onNIST second round candidates: NewHope, LAC, Round5 and ThreeBears. The lower bound and detailed

attack showing how to choose parameters for each scheme are given in Appendix B[13]. The lower bound and number of queries in theory in listed in Table 6, 7 in [13].

5 Applying Lattice Techniques in Key Reuse Attack

The key mismatch attack described in section 4 relies on querying oracle repeatedly to narrow the range of s in half each time.If the adversary $\mathcal{A}$ wants to recover Alice's reused secret $\boldsymbol{s_A}$ with the help of oracle $\mathcal{O}_s$ only, $\mathcal{A}$ has to query at least $H(S)$ times to recover one coefficient block of $\boldsymbol{s_A}$ as described in theorem 2.

However, with the help of lattice reduction techniques, the adversary $\mathcal{A}$ can bypass the lower bound given in [13]. Specifically, each query to the oracle O_s gives part of the information about the secret key, such information can be integrated into the DBDD instance through embedding techniques [32]. Henceforth, each incorporation of query results will introduce new constraints on s and will ultimately decrease the security level. It cost 4 steps to combine key mismatch attack with lattice techiniques.

$$LWE \xrightarrow[Step1]{Section3.4} DBDD_{\Lambda_0,\boldsymbol{\Sigma_0},\boldsymbol{\mu_0}} \xrightarrow[Step2]{query\ O_s} DBDD_{\Lambda_1,\boldsymbol{\Sigma_1},\boldsymbol{\mu_1}} \xrightarrow[Step3]{Section3.5} uSVP_{\Lambda'} \xrightarrow[Step4]{Section3.6} lattice\ reduction\ algorithm$$

For the key mismatch attack on lattice-based KEMs, the adversary $\mathcal{A}'s$ goal is to recover each coefficient of Alice's secret key $\boldsymbol{s_A}$ by accessing the oracle O_s for multiple times . We take the procedure of recovering the first coefficient block as an example to explain how to apply standard lattice attacks in the above attack.

-Step 1: construct a DBDD instance from the LWE problem of given KEM through the embedding technique described in section 3.4.

-Step 2:Then the adversary queries oracle O_s for several times , recovering one coefficient block of $\boldsymbol{s_A}$. A perfect hint arises in the form of $< \boldsymbol{s}, \boldsymbol{v} >= z_1$ from such queries,where $\boldsymbol{v}$ is an all-zero vector except $s_{m+1} = 1$.Such hint can also be written as $< \bar{\boldsymbol{s}}, \bar{\boldsymbol{v}} >= 0$,where $\boldsymbol{s} = (\boldsymbol{e}, \boldsymbol{z}, 1), \bar{\boldsymbol{v}} := (\boldsymbol{v}; -\boldsymbol{z_1})$.

-Step 3:the adversary includes the perfect hint $\boldsymbol{v}$ into a DBDD instance by modifying $DBDD_{\Lambda,\boldsymbol{\mu},\boldsymbol{\Sigma}}$ to $DBDD_{\Lambda',\boldsymbol{\mu'},\boldsymbol{\Sigma'}}$ as described in equation (8).

-Step 4:we transform $DBDD_{\Lambda',\boldsymbol{\mu'},\boldsymbol{\Sigma}}$ into an uSVP instance through steps described in Section 3.5.

-Step 5:we solve the uSVP instance through lattice reduction algorithm [34] to give the concrete hardness of the LWE problem after certain times of oracle queries.

For predicting attack costs of the new lattice, one only needs the dimension of the lattice Λ and its volume. According to the statement of [31]. The dimension of the lattice decreases by 1, and the volume of the lattice increases by a factor

$\|\boldsymbol{v}\|$. For the concrete relationship between query times and the cost of lattice attack, see Theorem 3.

Our work is based on a distorted version of the Bounded Distance Decoding problem(DBDD) given in [13]. According to Section 3, at most $H(\boldsymbol{S})+1$ queries to the PCA oracle O_s helps the adversary to recover a coefficient block of $\boldsymbol{s}$. Those queries help make hints in the form of $\langle \boldsymbol{s}, \boldsymbol{v} \rangle = z_i$ where $\boldsymbol{v}$ is an all-0 vector except the $m+1$-th coefficient is 1. We intersect the lattice with a hyperplane by integrating hint $\boldsymbol{v}$ into the lattice. Such a hint affects the mean and/or the covariance parameter of the DBDD instance, making the problem easier.

Let $\Lambda \subseteq \mathbb{R}^d$ be a lattice, $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ be a symmetric matrix and such that

$$Span(\boldsymbol{\Sigma}) \subsetneq Span(\boldsymbol{\Sigma} + \boldsymbol{\mu}^{\boldsymbol{T}}\boldsymbol{\mu}) = Span(\Lambda) \tag{10}$$

Given a lattice Λ of dimension $dim(\Lambda)$. Suppose the initial $BKZ-\beta$ can solve an $uSVP_\Lambda$ instance with secret $\boldsymbol{s}$ with block size β s.t. $\sqrt{\beta} \leq \delta_\beta^{2\beta-\dim(\Lambda)-1} \cdot \mathrm{Vol}(\Lambda)^{1/\dim(\Lambda)}$.

Let oracle O_s be an oracle returning whether the two shared keys $\boldsymbol{K_A}$ and $\boldsymbol{K_B}$ matches or not. Set $H(\boldsymbol{S})$ the Shannon entropy for $\boldsymbol{S}$, according to Theorem 1 in [13], we have $H(\boldsymbol{S}) \leq min\ E(\boldsymbol{S}) < H(\boldsymbol{S}) + 1$. The following theorem describes new hardness of the underlying LWE problem after each $min\ E(\boldsymbol{S})$ queries(which is described by block size β and volume $Vol(\Lambda)$).

Theorem 3. *Suppose the adversary $\mathcal{A}$ queries the oracle O_s for $min\ E(\boldsymbol{S})$ times and recovers the $i-th$ coefficient block of secret z. The adversary $\mathcal{A}$ can make hint $\boldsymbol{v}$ in the form of $\boldsymbol{v} = (\underbrace{0, \cdots, 0}_{m+i-1}, 1, 0, \cdots, 0)$. Including hint $\boldsymbol{v}$ modifies $DBDD_{\Lambda,\boldsymbol{\mu},\boldsymbol{\Sigma}}$ to $DBDD_{\Lambda',\boldsymbol{\mu'},\boldsymbol{\Sigma'}}$,where:*

$$\begin{aligned} \dim(\Lambda') &= \dim(\Lambda) - 1 \\ Vol(\Lambda') &= \mathrm{Vol}(\Lambda) \cdot \sqrt{1 + \boldsymbol{z_i^2}} \cdot det(\boldsymbol{\Pi}_\Lambda) \end{aligned} \tag{11}$$

When $\bar{\boldsymbol{v}}$ is a primality vector, we have

$$\mathrm{Vol}(\Lambda') = \mathrm{Vol}(\Lambda)\sqrt{1 + \boldsymbol{z_i^2}} \tag{12}$$

Proof. Suppose the adversary queries the oracle $\mathcal{O}_s$ to recover the $i-th$ coefficient block of secret $\boldsymbol{z}$. The hint made in such process is in the form of $\boldsymbol{v} = (\underbrace{0, \cdots, 0}_{m+i-1}, 1, 0, \cdots, 0)$ since we have $\langle \boldsymbol{s}, \boldsymbol{v} \rangle = \boldsymbol{z_i}$.

Hint $\boldsymbol{v}$ can also be written as $\langle \bar{\boldsymbol{s}}, \bar{\boldsymbol{v}} \rangle = 0$,where $\bar{\boldsymbol{s}} = (\boldsymbol{e}, \boldsymbol{z}, 1)$,$\bar{\boldsymbol{v}} = (\boldsymbol{v}; -\boldsymbol{z_1})$.Then the adversary $\mathcal{A}$ can integrate $\bar{\boldsymbol{v}}$ into the lattice as described in equation (8).

When $\bar{\boldsymbol{v}}$ is a primitive vector(see Definition 3),the volume of the lattice after integrating hint $\bar{\boldsymbol{v}}$ is $Vol(\Lambda) = \|\bar{\boldsymbol{v}}\| \cdot \mathrm{Vol}(\Lambda) = \mathrm{Vol}(\Lambda) \cdot \sqrt{1 + \boldsymbol{z_i^2}}$(see Lemma 3).

When $\bar{\boldsymbol{v}}$ is not in the span of Λ, we can also apply orthogonal projection $\bar{\boldsymbol{v}}' = \bar{\boldsymbol{v}} \cdot \boldsymbol{\Pi}_\Lambda$ of $\bar{\boldsymbol{v}}$ onto Λ.Replacing $\bar{\boldsymbol{v}}$ by $\bar{\boldsymbol{v}}'$ is still valid. According to equation (10), the orthogonal projection matrix is $\boldsymbol{\Pi_\Lambda} = \boldsymbol{\Pi_{\Sigma'}} = \sqrt{\boldsymbol{\Sigma'}}^{\sim} \cdot \boldsymbol{\Sigma'} \cdot \sqrt{\boldsymbol{\Sigma'}}^{\sim T}$, where

$\boldsymbol{\Sigma'} = \boldsymbol{\Sigma} + \boldsymbol{\mu}^T \cdot \boldsymbol{\mu}$ is the covariance matrix after homogenization as described in section 3.5, $\widetilde{\sqrt{\boldsymbol{\Sigma'}}}$ is the restricted inverse of $\sqrt{\boldsymbol{\Sigma'}}$ defined in definition 7.

Thus we have $\mathrm{Vol}(\Lambda') = \mathrm{Vol}(\Lambda) \cdot \sqrt{1 + \boldsymbol{z_i}^2} \cdot det(\boldsymbol{\Pi}_\Lambda))$,where $\boldsymbol{\Pi}_\Lambda$ is the orthogonal projection onto Λ . □

It is predicted that the $BKZ - \beta'$ can solve a $uSVP_{\Lambda'}$ after $min\ E(\boldsymbol{S})$ queries s.t. $\sqrt{\beta'} \leq \delta_{\beta'}^{2\beta' - dim(\Lambda') - 1} \cdot Vol(\Lambda')^{1/dim(\Lambda')}$, where $dim(\Lambda'), Vol(\Lambda')$ are as described in equation (11,12),and $min\ E(\boldsymbol{S})$ calculated by the optimal BRT is the lower bound for the minimum average number of queries as described in section 4.

Remark. *We can obtain another type of side information from the design of the schemes. Specifically, the so-called q-vectors (the vectors* $(q, 0, 0, \ldots, 0)$ *and its permutations) and their generalizations in the form of* $(q, -q, 0, \ldots, 0)$ *and all its permutations. According to [31], such information is called short vector hint.*

Definition 9. *[31, Definition 28] A short vector hint on the lattice is the knowledge of a short vector* $\bar{v}$ *such that* $\bar{\boldsymbol{v}} \in \Lambda$.

Such short vectors can also be integrated into the lattice. When the secret vector is short enough to be a solution after applying projection $\boldsymbol{\Pi}_{\bar{\boldsymbol{v}}}^{\perp}$ on $DBDD_{\Lambda, \boldsymbol{\Sigma}, \boldsymbol{\mu}}$,we have

$$\begin{aligned}
\Lambda' &= \Lambda \cdot \boldsymbol{\Pi}_{\bar{\boldsymbol{v}}}^{\perp} \\
\boldsymbol{\Sigma'} &= \left(\boldsymbol{\Pi}_{\bar{\boldsymbol{v}}}^{\perp}\right)^T \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{\Pi}_{\bar{\boldsymbol{v}}}^{\perp} \\
\boldsymbol{\mu'} &= \boldsymbol{\mu} \cdot \boldsymbol{\Pi}_{\bar{\boldsymbol{v}}}^{\perp}
\end{aligned}$$

The basis of the new lattice can be computed by applying the projection to all the vectors of its current basis, then use LLL to eliminate all linear dependencies in the resulting basis.

Each integration of the short vector hint decreases the dimension of the lattice by 1 and the volume of the lattice. The decrease of the determinant of Λ can be predicted via $rdet(\Lambda') = rdet(\boldsymbol{\Lambda}) \cdot \frac{\|\bar{\boldsymbol{v}}\|^2}{\bar{\boldsymbol{v}} \boldsymbol{\Sigma} \bar{\boldsymbol{v}}^{\boldsymbol{T}}}$.

Short vectors are not always worthy to integrate since we have to balance the dimension and the volume. Given a set $\boldsymbol{W}$ of short vectors of Λ, determining which subset leads to the easiest DBDD instance is a potentially hard problem as soon as Λ has been altered or if the set $\boldsymbol{W}$ is arbitrary.

The hardness of the new problem after integrating $\boldsymbol{V}$ into the lattice grows with $\frac{\mathrm{rdet}(\boldsymbol{\Sigma'})}{\mathrm{Vol}(\Lambda')^2} = \frac{\mathrm{rdet}(\boldsymbol{\Sigma})}{\mathrm{Vol}(\Lambda)^2} \cdot \frac{\det(\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}})^2}{\det(\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}})}$. For an artbitrary set $\boldsymbol{W}$, the problem of determining the optimal subset $\boldsymbol{V} \subset \boldsymbol{W}$ is NP-hard and is still NP-hard up to exponential approximation factors. However, we can get an approximate solution in polynomial time with the help of making greedy choices. Making greedy choices needs to compute $|\boldsymbol{V}| \cdot |\boldsymbol{W}|$ many matrix-vector products over rationals. Since $\boldsymbol{W}$ consists of q-vectors, such computation is somewhat practical.

6 Experiment Results

6.1 Kyber

Figure 2 gives the relationship between query times and security.

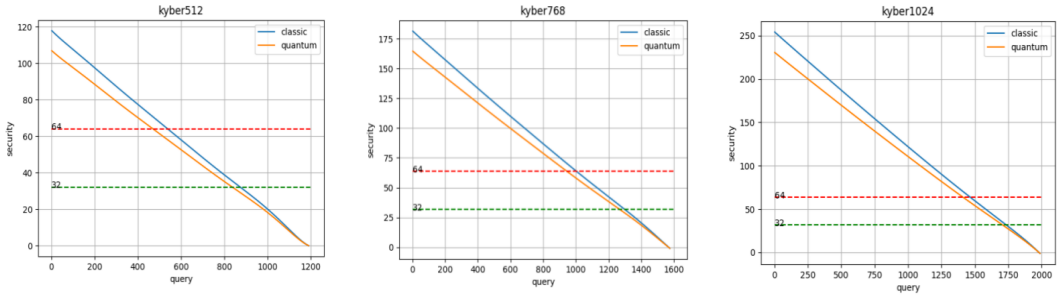

Figure 2: Relationship between query and security under primal attack

For our experiment, we make use of the LWE estimator from [31]. The analysis of the BKZ success condition is based on geometric-series assumption. It also integrates short vector into the lattice as described in Section 2.4.

Estimating the hardness needs the dimension of the lattice Λ and its volume only. According to 4.1.1, for Kyber512/Kyber768/Kyber1024, every 2.77/2.31/2.31 queries reduces the dimension of the lattice by 1. After integrating short vectors into the lattice, we get the concrete dimension of the lattice Λ and its volume, which tells us the security of current LWE problem after certain times of queries.

We list the number of queries needed for Kyber512 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 2 under primal/dual attack.

Table 2: Classical&Quantum Query-Security For Kyber 512

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	533	699	867	953	1033
primal(dual)	(657)	(761)	(865)	(922)	(981)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	464	646	831	925	1016
primal(dual)	(624))	(733)	(838)	(906)	(962)

We list the number of queries needed for Kyber768 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 3 under primal/dual attack.

Table 3: Classical&Quantum Query-Security For Kyber768

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	998	1144	1292	1366	1437
primal(dual)	(1112)	(1206)	(1320)	(1396)	(1481)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	938	1098	1259	1343	1423
primal(dual)	(1096)	(1176)	(1302)	(1361)	(1476)

We list the number of queries needed for Kyber1024 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 4 under primal/dual attack.

Table 4: Classical&Quantum Query-Security For Kyber 1024

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1459	1593	1728	1796	1862
primal(dual)	(1732)	(1805)	(1915)	(1973)	(2095)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1404	1550	1699	1775	1849
primal(dual)	(1673)	(1773)	(1893)	(1959)	(2029)

6.2 Saber

Figure 3 gives the relationship between query times and security.

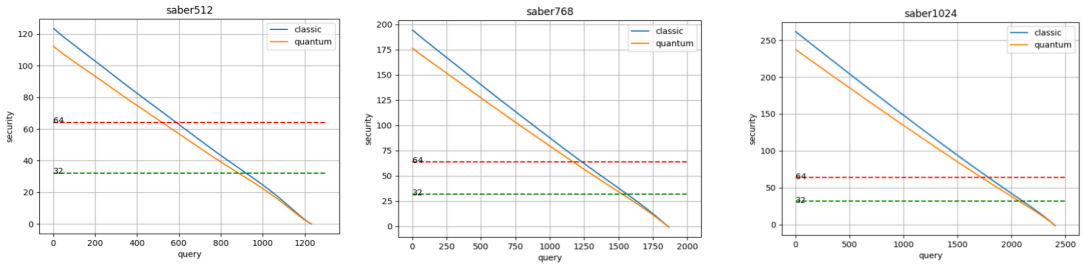

Figure 3: Relationship between query and security under primal attack

According to 4.2.1, for Saber512/Saber768/Saber1024, every 2.85/2.72/2.58 queries reduces the dimension of the lattice by 1. After integrating short vectors into the lattice, we get the concrete dimension of the lattice Λ and its volume, which tells us the security of current LWE problem after certain times of queries.

We list the number of queries needed for Saber512 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 5 under primal/dual attack.

Table 5: Classical&Quantum Query-Security For Saber 512

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	631	839	1075	1204	1340
primal(dual)	(844)	(933)	(1029)	(1096)	(1155)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	553	772	1023	1164	1311
primal(dual)	(799)	(902)	(1003)	(1066)	(1138)

We list the number of queries needed for Saber768 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 6 under primal/dual attack.

Table 6: Classical&Quantum Query-Security For Saber 768

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1230	1390	1554	1638	1717
primal(dual)	(1446)	(1550)	(1656)	(1714)	(1774)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1162	1339	1521	1611	1701
primal(dual)	(1415)	(1517)	(1625)	(1699)	(1756)

We list the number of queries needed for Saber1024 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 7 under primal/dual attack.

Table 7: Classical&Quantum Query-Security For Saber 1024

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1782	1941	2100	2179	2258
primal(dual)	(1963)	(2067)	(2179)	(2257)	(2326)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1718	1890	2066	2153	2243
primal(dual)	(1929)	(2029)	(2165)	(2223)	(2289)

6.3 Frodo

Figure 4 gives the relationship between query times and security.

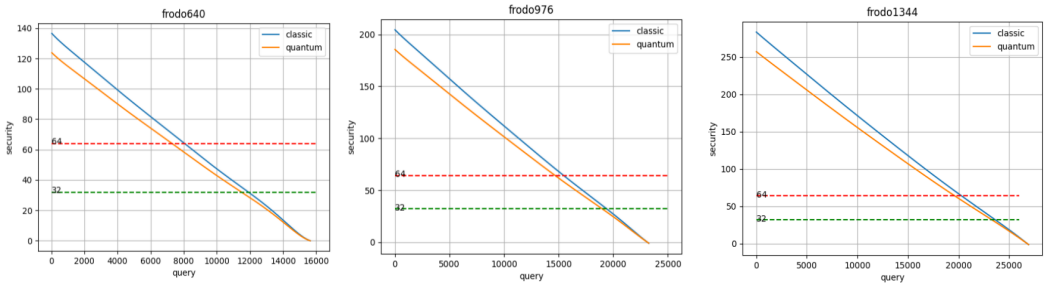

Figure 4: Relationship between query and security under primal attack

For our experiment we make use of the LWE estimator from [31]. The analysis of the BKZ success condition is based on geometric-series assumption. It also integrates short vector into the lattice as described in Section 2.4.

Estimating the hardness needs the dimension of the lattice Λ and its volume only. According to 4.1.1, for Frodo640/Frodo976/Frodo1344, every 3.58/3.33/2.73 queries reduces the dimension of the lattice by 1. After integrating short vectors into the lattice, we get the concrete dimension of the lattice Λ and its volume, which tells us the security of current LWE problem after certain times of queries.

We list the number of queries needed for Frodo640 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 8 under primal/dual attack.

Table 8: Classical&Quantum Query-Security For Frodo640

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	8005	9899	11821	12796	13772
primal(dual)	(10508)	(12001)	(13572)	(14577)	(15475)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	7230	9296	11419	12509	13571
primal(dual)	(9961)	(11474)	(13230)	(14235)	(15162)

We list the number of queries needed for Frodo976 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 9 under primal/dual attack.

Table 9: Classical&Quantum Query-Security For Frodo976

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	15445	17368	19346	20334	21296
primal(dual)	(19863)	(21041)	(22368)	(23154)	(23609)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	14670	16754	18918	20014	21109
primal(dual)	(19384)	(20792)	(21994)	(22773)	(23527)

We list the number of queries needed for Frodo1344 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 10 under primal/dual attack.

Table 10: Classical&Quantum Query-Security For Frodo1344

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	20234	21872	23577	24429	25259
primal(dual)	(23284)	(24257)	(25217)	(25841)	(26304)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	19556	21370	23205	24145	25084
primal(dual)	(23009)	(23812)	(25124)	(25581)	(26174)

6.4 NewHope

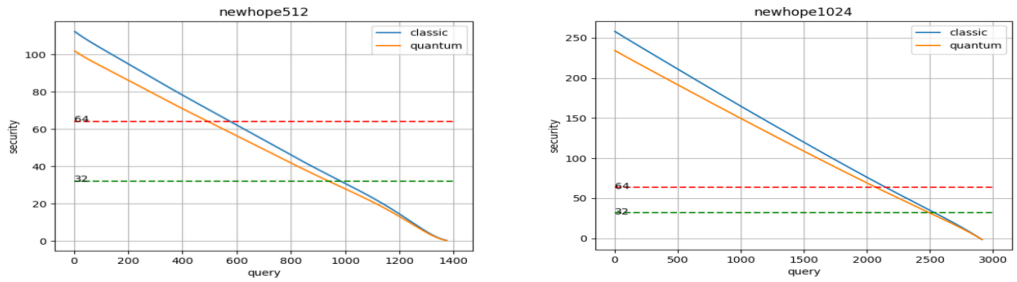

Figure 5: Relationship between query and security under primal attack

According to 4.2.1, for NewHope512/NewHope1024, every 3.24/3.11 queries reduces the dimension of the lattice by 1. After integrating short vectors into the lattice, we get the concrete dimension of the lattice Λ and its volume, which tells us the security of current LWE problem after certain times of queries.

We list the number of queries needed for NewHope512 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 11 under primal/dual attack.

Table 11: Classical&Quantum Query-Security For NewHope512

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	571	772	979	1083	1183
primal(dual)	(915)	(1054)	(1173)	(1231)	(1325)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	490	710	934	1050	1164
primal(dual)	(866)	(1011)	(1148)	(1224)	(1309)

We list the number of queries needed for NewHope1024 when the bit security of the underlying LWE reaches 64, 48, 32, 24, 16 in Table 12 under primal/dual attack.

Table 12: Classical&Quantum Query-Security For NewHope1024

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	2140	2333	2532	2632	2728
primal(dual)	(2475)	(2594)	(2714)	(2771)	(2864)
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	2062	2274	2488	2600	2709
primal(dual)	(2438)	(2555)	(2678)	(2738)	(2837)

6.5 Other NIST KEM Candidates

We also listed the lower bounds and the exception of query numbers for key mismatch attacks against other NIST KEM candidates: LAC, Round5,Three Bears in 13 ˜table?. We also estimates the relationship between query numbers and bit security.

Table 13: Classical&Quantum Query-Security For BabyBear

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	257	307	358	384	409
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	237	291	348	375	404

Table 14: Classical&Quantum Query-Security For MamaBear

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	472	512	553	573	593
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	456	500	544	567	589

Table 15: Classical&Quantum Query-Security For PapaBear

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	579	610	641	657	672
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	567	600	634	652	668

Table 16: Classical&Quantum Query-Security For LAC128

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	277	332	387	415	441
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	255	315	375	407	436

Table 17: Classical&Quantum Query-Security For LAC192

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	830	885	940	968	994
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	808	868	928	960	989

Table 18: Classical&Quantum Query-Security For LAC256

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1045	1114	1184	1219	1252
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	1018	1093	1169	1208	1245

Table 19: Classical Quantum Query-Security For Round5$_R$5ND1

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	351	420	491	527	562
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	323	398	427	516	555

Table 20: Classical Quantum Query-Security For Round5$_R$5ND3

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	613	717	821	875	927
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	571	684	799	859	917

Table 21: Classical Quantum Query-Security For Round5$_R$5ND5

Classical Bit Security	**64**	**48**	**32**	**24**	**16**
Query	914	1013	1112	1163	1213
Quantum Bit Security	**64**	**48**	**32**	**24**	**16**
Query	874	982	1091	1148	1203

7 Conclusions & Discussions

In this paper, we combine lattice techniques with a unified method of key mismatch attack as described in section 4(key mismatch attack). Our main technique is to transform the side-channel information / query results into hints.Integrating such hints into the lattice reduces its dimension and improves its volume, thus reduce the hardness of solving $uSVP$ problem. With the help of lattice technique, we can further reduce the number of queries needed to recover the whole reused secrets.

Our improved key mismatch attack is still a unified attack for CPA-secure lattice-based KEMs designed as Figure 1.We build a relationship in theory between LWE security of lattice-based KEMs and query times to oracle O_s in Theorem 3 (section 5). We also give relationship between concrete bit security and query times to $\mathcal{O}_s$ for all lattice-based CPA secure NIST second round candidate KEMs in section 6. Such improved key mismatch attack can also be applied to CCA-secure lattice-based KEMs when the adversary $\mathcal{A}$ can bypass **FO** transformation with the help of side channel information as described in[15, 37].

The lower bound of query numbers needed in key mismatch attackin [13]

already performs better than previous works. Using improved key mismatch attack with lattice techniques, we further decrease the number of queries needed for all KEMs listed in section 6. Compared to existing results in [13], our improved attack against Kyber512, Kyber768, and Kyber1024 with a reduced number of queries with 33.9%, 27.3% and 27% when the bit security decreases to 32 bits.

There are two strategies to recover Alice's reused secret $\boldsymbol{s_A}$. One strategy is to recover one coefficient block at a time, and a perfect hint matches such requirements. Our analysis in section 4 and section 5 follows the first strategy.

Another strategy to recover Alice's reused secret is to recover part of the information of one coefficient block at a time and recover part of the information of another unknown at a time. (insert definition here)

integrating such modular hints into the lattice reduces the volume of the lattice by k. When the range of secret and error is large and it requires more number of queries (e.g. Frodo640 requires 18227 times of queries to $\mathcal{O}_s$ to recover secret $\boldsymbol{s_A}$), the second strategy performs may perform better than the strategy used in this work.

However, It is difficult to create hints in the form of $\langle s, v\rangle = l \ mod \ k$. Take Kyber512 as an example, the secret $s_A \in [-3, 3]$ should be separated into $\{-2, 0, 2\}$ and $\{-3, -1, 1, 3\}$ if the adversary wants to make modular hints in the form of $\langle \boldsymbol{s}, \boldsymbol{v}\rangle = lmod2$. However, there exists no such an h to create such a modular hint. The situations are similar in other schemes. We leave this as an open problem: can we utilize modular hints in key mismatch attack to further decrease the number of queries needed to recover the reused secret?

References

[1] D. NIST, "Preparing for post-quantum cryptography:informatic." [Online]. Available: https://www.dhs.gov/sites/default/files/publications/post-quantum_cryptography_infographic_october_2021_508.pdf

[2] R. Avanzi *et al.*, "Crystals-kyber :algorithm specifications and supporting documentation," https://pq-crystals.org/kyber/index.shtml.

[3] NIST, "Selected algorithms 2022." [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[4] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Annual international cryptology conference*. Springer, 1999, pp. 537–554.

[5] J. Ding, S. Fluhrer, and S. Rv, "Complete attack on rlwe key exchange with reused keys, without signal leakage," in *Information Security and Privacy*, W. Susilo and G. Yang, Eds. Cham: Springer International Publishing, 2018, pp. 467–486.

[6] A. Bauer, H. Gilbert, G. Renault, and M. Rossi, "Assessment of the key-reuse resilience of newhope," in *Cryptographers' track at the RSA conference*. Springer, 2019, pp. 272–292.

[7] Y. Qin, C. Cheng, and J. Ding, "A complete and optimized key mismatch attack on nist candidate newhope," in *Computer Security – ESORICS 2019*, K. Sako, S. Schneider, and P. Y. A. Ryan, Eds. Cham: Springer International Publishing, 2019, pp. 504–520.

[8] S. Okada, Y. Wang, and T. Takagi, "Improving key mismatch attack on newhope with fewer queries," in *Information Security and Privacy*, J. K. Liu and H. Cui, Eds. Cham: Springer International Publishing, 2020, pp. 505–524.

[9] Y. Qin, C. Cheng, and J. Ding, "An efficient key mismatch attack on the nist second round candidate kyber," Cryptology ePrint Archive, Paper 2019/1343, 2019, https://eprint.iacr.org/2019/1343. [Online]. Available: https://eprint.iacr.org/2019/1343

[10] A. Greuet, S. Montoya, and G. Renault, "Attack on lac key exchange in misuse situation," Cryptology ePrint Archive, Paper 2020/063, 2020, https://eprint.iacr.org/2020/063. [Online]. Available: https://eprint.iacr.org/2020/063

[11] X. Zhang, C. Cheng, and R. Ding, "Small leaks sink a great ship: An evaluation of key reuse resilience of pqc third round finalist ntru-hrss," in *Information and Communications Security: 23rd International Conference, ICICS 2021, Chongqing, China, November 19-21, 2021, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 283–300. [Online]. Available: https://doi.org/10.1007/978-3-030-88052-1_17

[12] L. Huguenin-Dumittan and S. Vaudenay, "Classical misuse attacks on nist round 2 pqc," in *Applied Cryptography and Network Security*, M. Conti, J. Zhou, E. Casalicchio, and A. Spognardi, Eds. Cham: Springer International Publishing, 2020, pp. 208–227.

[13] Y. Qin, C. Cheng, X. Zhang, Y. Pan, L. Hu, and J. Ding, "A systematic approach and analysis of key mismatch attacks on lattice-based nist candidate kems," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2021, pp. 92–121.

[14] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[15] P. Ravi, S. S. Roy, A. Chattopadhyay, and S. Bhasin, "Generic side-channel attacks on cca-secure lattice-based pke and kems." *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 3, pp. 307–335, 2020.

[16] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A testing methodology for side channel resistance," vol. 7, 2011, pp. 115–136.

[17] M. Naehrig, E. Alkim *et al.*, "Frodokem learning with errors key encapsulation: algorithm specification and supporting documentation. submission to the nist post-quantum project (2019)," https://frodokem.org/.

[18] J.-P. D'Anvers, A. Karmakar, S. S. Roy, F. Vercauteren *et al.*, "Saber: Mod-lwr based kem algorithm specification and supporting documentation. submission to the nist post-quantum project (2019)," https://www.esat.kuleuven.be/cosic/pqcrypto/saber/.

[19] T. Poppelmann, E. Alkim *et al.*, "Newhope: algorithm specification and supporting documentation-version 1.03(2019)," https://newhopecrypto.org/.

[20] X. Lu *et al.*, "Lac: lattice-based cryptosystems algorithm specification and supporting documentation. submission to the nist post-quantum project (2019)," https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions.

[21] M. Hamburg, "Three bears nist round2 submission," https://sourceforge.net/projects/threebears/.

[22] H. Baan *et al.*, "Round5: merge of round2 and hila5 algorithm specification and supporting documentation. submission to the nist post-quantum project (2019)," https://round5.org/SupportingDocumentation/Round5Submission.pdf.

[23] Q. Guo and E. Mårtensson, "Do not bound to a single position: Near-optimal multi-positional mismatch attacks against kyber and saber," Cryptology ePrint Archive, Paper 2022/983, 2022, https://eprint.iacr.org/2022/983. [Online]. Available: https://eprint.iacr.org/2022/983

[24] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," Cryptology ePrint Archive, Paper 2015/046, 2015, https://eprint.iacr.org/2015/046. [Online]. Available: https://eprint.iacr.org/2015/046

[25] C. Chen *et al.*, "Ntru algorithm specifications and supporting documentation.submission to the nist post-quantum project (2019)," https://ntru.org/.

[26] D. J. Bernstein, Chitchanok, Chuengsatiansup, T. Lange, and van Vredendaal, "Ntru prime: round 2. submission to the nist post-quantum project (2019)," https://ntruprime.cr.yp.to/.

[27] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, pp. 34:1–34:40, 2009.

[28] 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*. Springer, 2010, pp. 1–23.

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

[30] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *International algorithmic number theory symposium*. Springer, 1998, pp. 267–288.

[31] D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "Lwe with side information: attacks and concrete security estimation," in *Annual International Cryptology Conference*. Springer, 2020, pp. 329–358.

[32] R. Kannan, "Minkowski's convex body theorem and integer programming," *Math. Oper. Res.*, vol. 12, pp. 415–440, 1987.

[33] M. R. Albrecht, R. Fitzpatrick, and F. Göpfert, "On the efficacy of solving lwe by reduction to unique-svp," in *International Conference on Information Security and Cryptology*. Springer, 2013, pp. 293–310.

[34] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, "Revisiting the expected cost of solving usvp and applications to lwe," in *Advances in Cryptology – ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds. Cham: Springer International Publishing, 2017, pp. 297–322.

[35] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key {Exchange—A} new hope," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 327–343.

[36] K. Mehlhorn, "Nearly optimal binary search trees," *Acta Informatica*, vol. 5, pp. 287–295, 2004.

[37] K. Xagawa, A. Ito, R. Ueno, J. Takahashi, and N. Homma, "Fault-injection attacks against nist's post-quantum cryptography round 3 kem candidates," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2021, pp. 33–61.