Decryption failure is more likely after success

Nina Bindel and John M. Schanck

Institute for Quantum Computing, University of Waterloo, Canada

Abstract. The user of an imperfectly correct lattice-based public-key encryption scheme leaks information about their secret key with each decryption query that they answer—even if they answer all queries successfully. Through a refinement of the D'Anvers–Guo–Johansson–Nilsson–Vercauteren–Verbauwhede failure boosting attack, we show that an adversary can use this information to improve his odds of finding a decryption failure. We also propose a new definition of δ-correctness, and we reassess the correctness of several submissions to NIST's post-quantum standardization effort.

Keywords: public-key cryptography, lattice-based cryptography, decryption failure

1 Introduction

Imperfectly correct lattice-based encryption schemes carry risks that perfectly correct schemes do not. Namely, whenever the decryption procedure fails it indicates "some correlation between the secret key and the encryption randomness" that reveals "information about the secret key" [21]. This is widely acknowledged. And yet, if one notes that *successful* decryption indicates a *lack* of correlation in precisely the same way, the consequence is startling: the user of an imperfectly correct lattice-based encryption scheme leaks information about their secret key with each decryption query that they answer. In this work, we show that an adversary can use information from successful decryptions to improve his odds of causing a decryption failure.

First, let us head off some objections. One might object that "[non-failing ciphertexts] will contain negligible information about the secret" [9]. For many schemes, we agree. However, even if a single ciphertext provides negligible information, an adversary might submit *many* non-failing ciphertexts.

One might also object that the risk of imperfect correctness can be mitigated using existing analyses. Indeed, when the Fujisaki–Okamoto transformation [13] is applied to a δ-correct passively secure encryption scheme, the result is an actively secure scheme with a failure probability of no more than $C\delta$ relative to an adversary who generates C ciphertexts [18, Theorem 3.1]. If the designers of an encryption scheme account for this factor of C loss of correctness, they can argue that decryption failures are not a risk. However, when designers rely on a conservative analysis of correctness, they may choose sub-optimal parameters.

We have seen several attempts to plot lattice-based encryption schemes along axes of size and security. These plots mask differences in correctness, even when they accurately represent tradeoffs between size and security against known attacks (c.f. [3]). We believe that an *accurate* and *concrete* assessment of correctness will enable a more fair comparison of the candidates.

Contributions. Our main contributions are: (1) a refinement of the D'Anvers–Guo–Johansson–Nilsson–Vercauteren–Verbauwhede *failure boosting* attack [5]; and (2) a new definition of correctness that is tailored for de-randomized encryption schemes. We also provide software[1] to calculate the correctness of FrodoKEM [21], Saber [6], Kyber [25], and (some parameter sets of) Round5 [14]. We partially validate our calculations with experiments on FrodoKEM.

Our refinement of failure boosting. We focus on the Lindner–Peikert encryption scheme [19], as it underlies all of the imperfectly correct lattice-based public-key encryption schemes that have been submitted to NIST. The correctness condition of these schemes can be stated as

$$-t \leq \langle s, e \rangle \leq t \tag{1}$$

where s is a vector related to the secret key, e is a vector related to the ciphertext randomness, and t is a system parameter.

An instantiation of the Lindner–Peikert scheme is said to be δ-correct if the probability that Equation (1) is violated for a random honestly generated s and a random honestly generated e is at most δ. The condition that e is honestly generated is reasonable when the scheme is *de-randomized*, e.g. when the Fujisaki–Okamoto transformation is used. In this case, the adversary needs the help of a random oracle to generate a valid ciphertext. The random oracle severely limits the adversary's ability to cause a decryption failure: if the adversary generates C ciphertexts, then his probability of causing a decryption failure is no more than $C\delta$, by a union bound.

The adversary's success probability may be far lower than $C\delta$. A key observation is that if Equation (1) is satisfied for some e, then it is likely to be satisfied for all e' that are *close* to e. One can quantify the *overlap* between queries and, in doing so, show that a sequence of queries with small overlap are more likely to cause a decryption failure than a sequence of queries with large overlap. An adversary cannot hope to achieve a success probability of $C\delta$ (on average) unless he submits sequences of queries with no overlap. We depict the overlap of a sequence of queries in Figure 1 and give a precise definition in Section 4.

In a *failure boosting attack*, the adversary improves his odds of triggering a decryption failure by searching for values of e that are of large norm. More precisely, the failure boosting adversary generates ciphertexts $c^{(i)}$, $1 \leq i \leq C$, with the help of the random oracle, and selects $Q \leq C$ ciphertexts to query. Previous analyses of failure boosting [9, 16, 5] assume that the adversary decides

[1] `https://jmschanck.info/code/20200203-decfail.tar.gz`

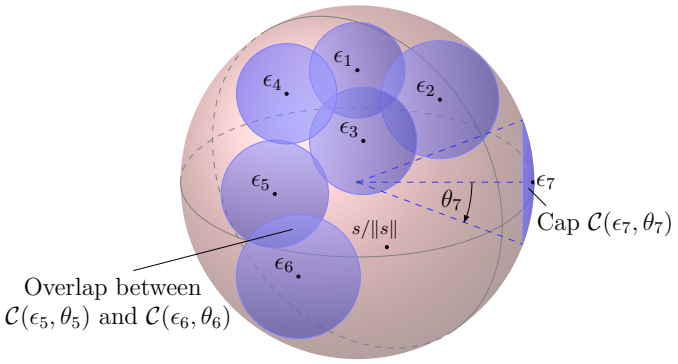

Fig. 1: A user who successfully decrypts ciphertexts $c^{(1)}, \ldots, c^{(7)}$ reveals that their secret, s, does not lie in the blue region. The ciphertext randomness determines the points $\epsilon_i := e^{(i)}/\|e^{(i)}\|_2$. The cap angle θ_i is determined by $\|e^{(i)}\|_2$ and $\|s\|_2$. The probability that a further query, $c^{(8)}$, causes a decryption failure depends on the extent to which the cap of angle θ_8 about ϵ_8 intersects the blue region.

whether to query $c^{(i)}$ by looking only at $c^{(i)}$. In effect, previous analyses ignore the overlap between queries. In contrast, we allow the adversary to minimize the overlap between his queries.

Our focus here is on finding one decryption failure. After observing a decryption failure, the adversary should switch to a different strategy such as the recently proposed *directional failure boosting* of D'Anvers, Rossi, and Virdia [7]. We will not discuss the process of estimating the secret from a collection of failures. For further background on failure boosting, and reaction attacks on lattice-based schemes more generally, see [8, 16, 5].

Correctness definition. We propose an alternative definition of δ-correctness to the one by Hofheinz–Hövelmanns–Kiltz [18]. The correctness experiment in [18] provides the adversary with the secret key. In contrast, our experiment provides the adversary only with the public key and a decryption oracle, and can therefore be run inside an IND-CCA experiment. More importantly, our definition allows a more fine-grained analysis of the impact of adaptive decryption queries on de-randomized encryption schemes. We give our formal definition in Section 3.

2 Preliminaries

Notation. For a finite set $\mathcal{X}$ we write $x \leftarrow_{\$} \mathcal{X}$ to say that x is sampled uniformly from $\mathcal{X}$. For a distribution χ on $\mathcal{X}$, we write $x \leftarrow \chi$ to say that x is sampled according to χ. We denote the joint distribution of $x \leftarrow \chi_1$ and $y \leftarrow \chi_2$ by $\chi_1 \times \chi_2$. If χ_1 and χ_2 are distributions on an abelian group, and $(x, y) \leftarrow \chi_1 \times \chi_2$, then we denote the distribution of $x + y$ by $\chi_1 * \chi_2$ where $(\chi_1 * \chi_2)(z) = \sum_{w \in \mathcal{X}} \chi_1(w)\chi_2(z - w)$.

2.1 Definition of PKEs and KEMs

A public-key encryption scheme $\mathsf{P} = (\mathsf{Keygen}, \mathsf{Encr}, \mathsf{Decr})$ is defined over a finite message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, a secret key space $\mathcal{SK}$ and a public key space $\mathcal{PK}$. In particular, Keygen is a randomized algorithm returning $\mathrm{sk} \in \mathcal{SK}$ and $\mathrm{pk} \in \mathcal{PK}$; Encr is a randomized, or de-randomized, algorithm that takes as input a public key pk and a message $\mathrm{msg} \in \mathcal{M}$ and outputs a ciphertext $c \in \mathcal{C}$; Decr is a deterministic algorithm that takes as input $\mathrm{sk} \in \mathcal{SK}$ and $c \in \mathcal{C}$ and returns either a message $\mathrm{msg} \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$ indicating failure.

A key encapsulation meachanism (KEM) $\mathsf{K} = (\mathsf{Keygen}, \mathsf{Encaps}, \mathsf{Decaps})$ is defined over a ciphertext space $\mathcal{C}$, the secret key space $\mathcal{SK}$, a public key space $\mathcal{PK}$, and the key space $\mathcal{K}$. In particular, Keygen is a randomized algorithm that returns $\mathrm{pk} \in \mathcal{PK}$ and $\mathrm{sk} \in \mathcal{SK}$; Encaps is a randomized algorithm that takes as input $\mathrm{pk} \in \mathcal{PK}$ and outputs $c \in \mathcal{C}$ and $k \in K$; $\mathsf{Decaps}(\mathrm{sk}, c)$ is a deterministic algorithm that upon input $\mathrm{sk} \in \mathcal{SK}$ and $c \in \mathcal{C}$, returns $\kappa \in K$ or a special symbol $\perp \notin K$ indicating that c is not a valid ciphertext.

Fujisaki–Okamoto transform. The Fujisaki–Okamoto (FO) transform [12, 13, 10] can be used to construct an adaptively secure KEM from passively secure public-key encryption (PKE). Hofheinz, Hövelmanns, and Kiltz provide a decomposition of the FO transform into a sequence of simpler transformations [18]; Bernstein and Persichetti provide a complementary analysis [4]. These works emphasize that the FO transform performs three tasks:

- Derandomization: A probabilistic PKE is transformed into a deterministic PKE by fixing the coins used in encryption to a hash of the message.
- Reencryption: A deterministic PKE is transformed into a rigid[2] deterministic PKE that returns an error symbol, $\perp$, whenever the message obtained by decrypting c does not reencrypt to c.
- Hashing: A rigid deterministic PKE is transformed into an IND-CCA KEM that encrypts a random message and outputs a hash of this message as the session key.

Hofheinz, Hövelmanns, and Kiltz handle the derandomization and reencryption with a single transformation called T. Suppose that $\mathsf{P} = (\mathsf{Keygen}, \mathsf{Encr}, \mathsf{Decr})$ is a probabilistic PKE, that $G : \mathcal{M} \to \mathcal{R}$ and $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ are random oracles, and that $F : \mathcal{K}_F \times \mathcal{C} \to \mathcal{K}$ is a pseudorandom function family. Then $\mathsf{P}_1 = \mathsf{T}[\mathsf{P}, G] = (\mathsf{Keygen}, \mathsf{Encr}_1, \mathsf{Decr})$ is a derandomized PKE with $\mathsf{Encr}_1(\mathrm{pk}, \mathrm{msg}) := \mathsf{Encr}(\mathrm{pk}, \mathrm{msg};\ G(\mathrm{msg}))$.

Hofheinz, Hövelmanns, and Kiltz provide variants of the hashing step called $U^{\not\perp}$ and $U^{\not\perp}_{\mathrm{msg}}$. The $U^{\not\perp}$ transformation is defined in Figure 2. The $U^{\not\perp}_{\mathrm{msg}}$ transformation is defined similarly but with the encapsulation key equal to $H(\mathrm{msg})$ rather than $H(\mathrm{msg}, c)$.

[2] The term "rigid" is due to Bernstein and Persichetti. See [4, Section 6].

$\mathsf{Keygen}()$:	$\mathsf{Encaps}(\mathrm{pk})$:	$\mathsf{Decaps}\,((\mathrm{sk}_1, \mathrm{sk}_2), c)$:
1 $(\mathrm{pk}, \mathrm{sk}_1) \leftarrow \mathsf{Keygen}()$	1 $\mathrm{msg} \leftarrow_{\$} \mathcal{M}$	1 $\mathrm{msg} \leftarrow \mathsf{Decr}_1(\mathrm{sk}_1, c)$
2 $\mathrm{sk}_2 \leftarrow_{\$} \mathcal{K}_F$	2 $c \leftarrow \mathsf{Encr}_1(\mathrm{pk}, \mathrm{msg})$	2 if $\mathrm{msg} = \perp$:
3 $\mathrm{sk} \leftarrow (\mathrm{sk}_1, \mathrm{sk}_2)$	3 $K \leftarrow H(\mathrm{msg}, c)$	3 return $F(\mathrm{sk}_2, c)$
4 return $(\mathrm{pk}, \mathrm{sk})$	4 return (K, c)	4 return $H(\mathrm{msg}, c)$

Fig. 2: The algorithms of the $U^{\not\perp}[\mathsf{P}_1, H, F] = (\mathsf{Keygen}, \mathsf{Encaps}, \mathsf{Decaps})$ KEM.

δ-correctness. Hofheinz, Hövelmanns, and Kiltz [18, Section 2.1] define δ-correctness for a PKE as follows.

Definition 1 (δ-correctness for PKEs). *A public-key encryption scheme* $\mathsf{P} = (\mathsf{Keygen}, \mathsf{Encr}, \mathsf{Decr})$ *is δ-correct if*

$$\mathbf{E}\left[\max_{msg \in \mathcal{M}} \Pr[\mathsf{Decr}(sk, c) \neq msg \mid c \leftarrow \mathsf{Encr}(pk, msg)]\right] \leq \delta, \tag{2}$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Keygen}()$. Equivalently, δ-correctness means that for all (possibly unbounded) adversaries $\mathcal{A}$, $\Pr[\mathsf{COR}^{\mathcal{A}}_{\mathsf{P}}] \leq \delta$, where the correctness game COR is defined in Figure 3.

The definition is carefully crafted to obtain a security proof of the T transform—the derandomization step during the Fujisaki–Okamoto transformation [12, 13, 10] (cf. Appendix 2.1). Moreover, Theorem 3.1 of [18] states (in part) that if P is δ-correct, then $\mathsf{T}[\mathsf{P}, \mathsf{G}]$ is δ_1-correct where $\delta_1(q_{\mathsf{G}}) \leq q_{\mathsf{G}} \cdot \delta$ and q_{G} is the number of queries that the adversary makes to G.

2.2 Lindner–Peikert encryption scheme

The Lindner–Peikert scheme [19] is a passively secure public-key encryption scheme based on the learning with errors (LWE) problem [24]. It obtains smaller keys and ciphertexts than earlier LWE encryption schemes [24, 15] by using the LWE hardness assumption twice in its security reduction.

Parameters. The system parameters are $(R, q, k, \chi_s, \chi_e, \chi_{e'})$ where R is the *base ring*, q is the integer modulus, k is the R-module rank, χ_s and χ_e are probability distributions supported on R^k, and $\chi_{e'}$ is a probability distribution supported on R. The base ring must have the additive structure of $\mathbb{Z}^m$ for some positive integer m. The $\mathbb{Z}$-module rank, or dimension, of the system is $n = km$. We refer to χ_s as the *secret distribution*, and to χ_e and $\chi_{e'}$ as the *error distributions*. Another important derived parameter is the error threshold t, cf. Section 4.

Rings. Commonly used base rings are $R = \mathbb{Z}$ and $R = \mathbb{Z}[x]/(x^m + 1)$ with m a power of two. In the latter case, we view elements of R as vectors in $\mathbb{R}^m$

by expressing them over the power basis $\{1, x, x^2, \cdots, x^{m-1}\}$, i.e. we use the *coefficient embedding*. We identify the power basis with the standard basis of $\mathbb{R}^m$. For $a \in R$, we write $\|a\|_1 = \sum_i \left|\langle x^i, a\rangle\right|$, $\|a\|_2 = \sqrt{\langle a, a\rangle}$, and $\|a\|_\infty = \max_i \left|\langle x^i, a\rangle\right|$. For elements $a = (a_1, \ldots, a_k)$ and $b = (b_1, \ldots, b_k)$ of a rank k module over R we write $\langle a, b\rangle = \langle a_1, b_1\rangle + \cdots + \langle a_k, b_k\rangle$. We write $\bar{r}$ for the adjoint of the "multiplication by r" map, i.e. $\langle a, rb\rangle = \langle \bar{r}a, b\rangle$. With $R = \mathbb{Z}$ we have $r = \bar{r}$. With $R = \mathbb{Z}[x]/(x^m + 1)$ we have that $\bar{r}$ is the image of r under $x \mapsto -x^{m-1}$.

Message encoding. The message space is a subset of R that is defined by maps encode and decode. These maps must satisfy $\mathsf{decode}(\mathsf{encode}(\text{msg})) = \text{msg}$ for all bit strings msg in the domain of encode. A typical choice for a plain LWE system is $\mathsf{encode} : \{0,1\} \to \mathbb{Z}$ and $\mathsf{decode} : \mathbb{Z} \to \{0,1\}$ with $\mathsf{encode}(\text{msg}) = \text{msg} \cdot \lfloor q/2 \rfloor$ and $\mathsf{decode}(\text{msg}) = \{0 \text{ if } |\text{msg} \bmod q| \in [0, q/4);\ \ 1 \text{ otherwise}\}$. We call this the *standard encoding*. Observe that $\mathsf{decode}(\mathsf{encode}(\text{msg}) + \delta) = \text{msg}$ if $|\delta| < q/4$, so we say that the standard encoding has an error threshold of $t = q/4$. The *standard b-bit encoding* is defined similarly: it divides $[0, q/2)$ into 2^b intervals and has an error threshold of $q/2^{b+1}$. Elements of $\{0,1\}^{b \cdot \text{msg}}$ can be encoded into elements of R by extending the standard b-bit encoding component wise on the power basis.

Algorithms. The key generation, encryption and decryption routines of the passively secure encryption scheme are as follows.

- $\mathsf{Keygen}()$: Sample a $k \times k$ matrix A with each coefficient chosen independently from the uniform distribution on R/q. Sample $k \times 1$ vectors s_1 and s_2 independently from χ_s. Compute $b = (s_1 - As_2) \bmod q$. The public key is (A, b). The secret key is (s_1, s_2).
- $\mathsf{Encr}\,(\text{msg}, (A, b))$: Sample $1 \times k$ vectors e_1 and e_2 independently from χ_e. Sample e_3 from $\chi_{e'}$. Compute the ciphertext (c_1, c_2) with

$$c_1 = (e_1 A + e_2) \bmod q, \quad c_2 = (e_1 b + e_3 + \mathsf{encode}(\text{msg})) \bmod q.$$

- $\mathsf{Decr}\,((c_1, c_2), (s_1, s_2))$: To decrypt (c_1, c_2) using the secret key (s_1, s_2), let $v = (c_1 s_2 + c_2) \bmod q$ and output $\mathsf{decode}(v)$.

3 Correctness in an adaptive setting

The Hofheinz–Hövelmanns–Kiltz (HHK) definition of δ-correctness (Definition 1 in Section 2.1) involves an expectation over keys and ciphertexts. Care must be taken when the key is fixed (as in an IND-CCA setting) or when the ciphertext is determined by the message (as in a derandomized encryption scheme). For derandomized schemes that use a random oracle G during encryption, HHK define a notion of $\delta(q_G)$-correctness which is stated in terms of the number of queries q_G that the adversary makes to G. They prove that a δ-correct scheme

that is derandomized using their T transformation has a correctness error of $\delta(q_G) \leq q_G \cdot \delta$ [18, Theorem 3.1].

The loss of correctness caused by derandomization is often ignored in practice. For example, the authors of the FrodoKEM NIST submission correctly calculate the *one-shot* correctness (the probability of decryption failure for a random key and random ciphertext) of their IND-CPA PKE [21, Section 2.2.7]. They note that the one-shot correctness is equal to the δ-correctness [21, Equation 2]. They then apply the T transformation and claim that the correctness of the resulting IND-CCA PKE is equal to the one-shot correctness of the underlying IND-CPA PKE [21, Section 2.2.10]. This claim is not justified.

And yet, a full factor q_G loss of correctness does not seem realistic. To address this, we propose the following alternative to the $\delta(q_G)$-correctness. This definition restricts the adversary's time, t, and number of decryption queries, q_d.

Definition 2 ($\delta(q_d, t)$-correctness for PKEs). *Let P be a derandomized PKE against a (classical or quantum) adversary $\mathcal{A}$ making at most q_d (classical) queries to its decryption oracle D and running in time t. We say, P is $\delta(q_d, t)$-correct if*

$$\Pr[\mathsf{COR\text{-}ad}^{\mathsf{A}}_{\mathsf{PKE}} \to 1] \leq \delta(q_d, t),$$

where the correctness game $\mathsf{COR\text{-}ad}$ is defined in Figure 3.

In contrast to the HHK correctness experiment (COR in Figure 3), our correctness experiment ($\mathsf{COR\text{-}ad}$ in Figure 3) does not provide the adversary with the user's secret key, and can be run as part of the IND-CCA security experiment[3] In this case we call it $\mathsf{COR\text{-}ad\text{-}CCA}$.

It is important to note that running $\mathsf{COR\text{-}ad\text{-}CCA}$ inside the IND-CCA experiment does not change the power of the IND-CCA adversary; in particular, the number of decryption queries q'_d in $\mathsf{COR\text{-}ad\text{-}CCA}$ is no more than the number of decryption queries q_d in $\mathsf{IND\text{-}CCA}$. As such, one can obtain an upper bound on the IND-CCA security of a scheme given the $\delta(q_d, t)$-correctness of a scheme and an attack that violates IND-CCA security using decryption failures.

4 Correctness of the Lindner–Peikert scheme

Suppose that (c_1, c_2) is an honest encryption of msg to a user with public key (A, b) and secret key $s = (s_1, s_2)$. Let (e_1, e_2, e_3) be the noise that was used to generate (c_1, c_2), and let $e = (e_1, e_2)$. Decryption will be successful, i.e., the decrypting party will recover msg exactly, as long as

$$\|e_1 s_1 + e_2 s_2 + e_3\|_\infty < t, \tag{3}$$

where t is the error threshold. The exact one-shot probability of failure can be calculated from Equation 3 (our software does this). However, we will use a

[3] A slight modification is necessary, as the IND-CCA decryption oracle gives special treatment to the challenge ciphertext.

$\underline{\text{Expt}^{\mathsf{COR}}_{\mathsf{P}}(\mathcal{A})}$:

1 $(\text{pk}, \text{sk}) \leftarrow \mathsf{Keygen}()$
2 $\text{msg} \leftarrow \mathsf{A}(sk, pk)$
3 $c \leftarrow \mathsf{Encr}(\text{pk}, \text{msg})$
4 return $[\mathsf{Dec}(sk, c) \neq \text{msg}]$

$\underline{\text{Expt}^{\mathsf{COR\text{-}ad\text{-}CCA}}_{\mathsf{P}}(\mathcal{A}, c^*, q_d, L_d, H)}$:

1 $(\text{pk}, \text{sk}) \leftarrow \mathsf{Keygen}()$
2 $\text{msg} \leftarrow \mathcal{A}^{H,D}(\text{pk}, c^*)$
3 $c \leftarrow \mathsf{Encr}(\text{pk}, \text{msg})$
4 return $[\mathsf{Decr}(sk, c) \neq \text{msg}]$

Decryption oracle $D(c)$:

1 $q_d \leftarrow q_d + 1$
2 if $(c = c^*)$: $r \leftarrow \perp$, $L_d = L_d \cup (c, r)$
3 else: $r \leftarrow \mathsf{Decr}(\text{sk}, c)$, $L_d = L_d \cup (c, r)$
4 return r

$\underline{\text{Expt}^{\mathsf{IND\text{-}CCA}}_{\mathsf{P}}((\mathcal{A}_1, \mathcal{A}_2))}$:

1 $H \xleftarrow{\$} \mathcal{H}$
2 $q_d \leftarrow 0$
3 $L_d = \{\}$
4 $(\text{pk}, \text{sk}) \leftarrow \mathsf{Keygen}()$
5 $\text{msg}_0, \text{msg}_1 \leftarrow \mathcal{A}_1^H(\text{pk})$
6 $b \xleftarrow{\$} \{0, 1\}$
7 $c^* \leftarrow \mathsf{Encr}(\text{pk}, \text{msg}_b^*)$
8 $b' \leftarrow \mathcal{A}_2^{H,D}(\text{pk}, c^*)$
9 return $[b = b']$

Fig. 3: COR and IND-CCA experiment for any PKE P;COR-ad-CCA experiment for a (derandomized) PKE P.

slightly weaker condition to analyze the the probability of failure in an adaptive setting. First, an application of the triangle inequality gives

$$\|e_1 s_1 + e_2 s_2\|_\infty < t - \|e_3\|_\infty. \tag{4}$$

Then, by fixing some $\gamma \geq \|e_3\|_\infty$ and using properties of the max-norm and inner product that we discussed in Section 2.2, we have

$$\|e_1 s_1 + e_2 s_2\|_\infty = \max_{0 \leq i < m} \left| \langle \bar{s}, x^i e \rangle \right| < t - \gamma. \tag{5}$$

A geometric interpretation. Let $\mathcal{S}$ be the unit sphere in $\mathbb{R}^d$. We denote the *angular distance* between points u and v in $\mathbb{R}^d$ by

$$\theta(u, v) = \arccos\left(\frac{\langle u, v \rangle}{\|u\|_2 \cdot \|v\|_2}\right), \tag{6}$$

where $\arccos(x) \in [0, \pi]$. The *spherical cap* of angle θ about u is

$$\mathcal{C}(u, \theta) = \{v \in \mathcal{S} : \theta(u, v) \leq \theta\}. \tag{7}$$

Equation (5) tells us that each *successful* decryption reveals some geometric information about s, as explained next. By restating the condition $\langle \bar{s}, e \rangle < t - \gamma$ (without the absolute value bars that appear in Equation (5)) in terms of the angular distance,

$$\theta(\bar{s}, e) = \arccos\left(\frac{\langle \bar{s}, e \rangle}{\|s\|_2 \cdot \|e\|_2}\right) > \arccos\left(\frac{t - \gamma}{\|s\|_2 \cdot \|e\|_2}\right) = \theta^*, \tag{8}$$

we see that $\langle \bar{s}, e \rangle < t - \gamma$ implies that $\bar{s}/\|s\|_2$ *does not* lie in the cap of angle θ^* about $e/\|e\|_2$. The full condition, $|\langle \bar{s}, e \rangle| < t - \gamma$, also says that $\bar{s}/\|s\|_2$ does not lie in the cap of angle θ^* about $-e/\|e\|_2$. An adversary can use this information to improve his odds of triggering a decryption failure.

A heuristic assumption. We measure the volume of subsets of $\mathcal{S} \subset \mathbb{R}^d$ using the $(d-1)$-dimensional spherical probability measure, σ. This measure is normalized such that $\sigma(\mathcal{S}) = 1$. If u is a point on $\mathcal{S}$ and v is drawn uniformly from $\mathcal{S}$, then the probability that $\theta(u, v) \leq \theta$ is $C(\theta) = \sigma(\mathcal{C}(u, \theta))$. It is important to note that $C(\theta)$ does not depend on u. We assume the following heuristic in our analysis.

Heuristic 1 (Spherical symmetry) *For fixed $\bar{s}$ and $e \leftarrow \chi_e \times \chi_e$, the probability that $\theta(\bar{s}, e) \leq \varphi$, for any $0 < \varphi < \pi/2$, is $C(\varphi)$. Equivalently, $e/\|e\|_2$ "looks like" a uniformly random point on $\mathcal{S}$.*

If Heuristic 1 holds true, the probability that e causes a decryption failure is at least $2C(\theta^*)$. It may even be as large as $2mC(\theta^*)$, due to the maximization over i in Equation (5).

Remark 1. Previous analyses of failure boosting [5] have modeled the distribution of $\chi_e \times \chi_e$ with a spherically symmetric Gaussian distribution. In contrast, our software uses the exact distribution of $\chi_e \times \chi_e$. Our experiments in Section 6 indicate that the spherical symmetry assumption is reasonable for Frodo640. Further experiments are needed for other schemes.

4.1 The efficacy of a query set

Recall θ^* of the previous section. We write $\theta^*_\alpha(\beta; z) = \arccos(z/\alpha\beta)$ with $0 \leq \theta^*_\alpha(\beta; z) \leq \frac{\pi}{2}$. We are primarily interested in the case $\alpha = \|s\|_2$ and $\beta = \|e\|_2$. In later sections we will take α to be an approximation to $\|s\|_2$. We write $\theta^*_\alpha(e; z)$ in place of the cumbersome notation $\theta^*_\alpha(\|e\|_2; z)$, and we suppress the dependence on z when it is clear.

We refer to $e = (e_1, e_2)$ as the "query", rather than (c_1, c_2). We also ignore both the absolute value bars and the maximization over i in Equation (5). This way queries are one-to-one with spherical caps, and each query can be thought of as "exploring" some cap; by querying e the adversary learns whether or not $\bar{s}$ lies in $C(\theta^*_\alpha(e))$.

We define the *efficacy* of a set E of queries to be the fraction of the sphere that the corresponding caps cover:

$$\mathrm{Eff}_\alpha(E) = \sigma\left(\bigcup_{e \in E} \mathcal{C}\left(e, \theta^*_\alpha(e)\right)\right). \tag{9}$$

Under the spherical symmetry heuristic, the probability that an adversary causes a decryption failure is proportional to the efficacy of his queries. An intelligent adversary will maximize the efficacy of his queries while minimizing the number of queries that he makes. Adversaries are constrained both by their computational power and by the need to collaborate with a random oracle.

In the notation of Definition 2, an instantiation of the Lindner–Peikert scheme is $\delta(q_d, t)$-correct if

$$\delta(q_d, t) \geq 2m\,\mathrm{Eff}_\alpha(E) \tag{10}$$

for all E of size $|E| \leq q_d$ that an adversary can produce in time t. It is important to note that some instantiations exchange more than one element of R; for instance, FrodoKEM exchanges 64 elements of $\mathbb{Z}$. For such instantiations the right hand side of Equation (10) should be $2\ell m \operatorname{Eff}_\alpha(E)$ where ℓ is the number of coefficients exchanged. Assuming spherical symmetry, the actual correctness error can be anywhere between $2\operatorname{Eff}_\alpha(E)$ and $2\ell m \operatorname{Eff}_\alpha(E)$, as the failure events may not be independent.

4.2 Approximating the efficacy

The efficacy of a query set may be difficult to compute exactly. Using the principle of inclusion-exclusion, we can write a k-th order approximation to $\operatorname{Eff}_\alpha(E)$ as

$$\operatorname{Eff}_\alpha^{(\mathrm{k})}(E) = \sum_{\substack{F \subset E \\ 0 < |F| \leq k}} (-1)^{|F|+1} \cdot \sigma\left(\bigcap_{e \in F} \mathcal{C}\left(e, \theta_\alpha^*(e)\right)\right). \tag{11}$$

Maximizing the second-order approximation,

$$\operatorname{Eff}_\alpha^{(2)}(E) = \sum_{e \in E} C(\theta_\alpha^*(e)) - \sum_{\{e,e'\} \subset E} \sigma\left(\mathcal{C}(e, \theta_\alpha^*(e)) \cap \mathcal{C}(e', \theta_\alpha^*(e'))\right), \tag{12}$$

already presents quite a challenge. We do not consider algorithms for approximating the efficacy here, but we note that techniques from the near-neighbor search literature, e.g. [2], may be useful for producing high-efficacy query sets.

4.3 The efficacy of a random query set

A first-order approximation to the efficacy of a random query set, normalized by the query set size N, is

$$\mathrm{Q}_\alpha(\chi_1, \chi_2) = \lim_{N \to \infty} \frac{1}{N} \mathbb{E}\left[\operatorname{Eff}_\alpha^{(1)}(V)\right] \tag{13}$$

where the expectation is taken over sets V of N elements drawn independently from $\chi_1 \times \chi_2$. Equation (13) can also be written as the expected size of a cap with respect to the 2-norm distribution of v drawn from $\chi_1 \times \chi_2$,

$$\mathrm{Q}_\alpha(\chi_1, \chi_2) = \sum_{j > 0} \Pr\left[\|v\|_2 = j\right] \cdot C\left(\theta_\alpha^*(j)\right). \tag{14}$$

5 Heuristic analysis of NIST candidates

In this section we calculate *first-order approximations* to the efficacy of *random query sets* drawn from distributions that come from concrete instantiations of the Lindner–Peikert encryption scheme. It is important to note that a first-order

approximation to the efficacy ignores the overlap between queries; it thereby overestimates efficacy and underestimate correctness. Since we are ignoring the overlap, we expect our results to closely mirror those of D'Anvers, Guo, Johansson, Nilsson, Vercauteren, and Verbauwhede from [5]. The calculations that we perform are quite different and serve as an independent check on their results.

We analyze Saber [6], the R5ND_PKE_0d and R5N1_PKE_0d parameter sets of Round5 [14], Frodo [21], and Kyber [25]; all of which are second round candidates in NIST's post-quantum standardization effort.

5.1 Overview

We caution the reader that the following sketch of our analysis is only accurate for Frodo. The treatment of the other schemes is described in Appendix A.

Let χ be a distribution on R. We write $\|\chi\|_2$ and $|\langle 1, \chi\rangle|$, respectively, for the distribution of $\|r\|_2$ and $|\langle 1, r\rangle|$ when $r \leftarrow \chi$. The top u-th quantile of $\|\chi\|_2$ is the largest $\beta \in \mathbb{Z}_+$ for which $\Pr_{r\leftarrow\chi}[\|r\|_2 \geq \beta] \geq 1/u$. We write $\chi(u)$ for the distribution of $r \leftarrow \chi$ conditioned on the event that $\|r\|_2 \geq \beta$. It is important to note that $\chi(1) = \chi$.

We assume that the user has drawn a secret key s from $\chi_s(v) \times \chi_s(v)$, for some $v \geq 1$. A random user does so with probability $1/v^2$. Unless otherwise stated we take $v = 2$, i.e., we assume that the user has a key of above-median length in both components. We evaluate correctness using $\mathrm{Q}_\alpha(\cdot, \cdot)$ which depends on the γ of Equation (5) through θ^*_α. We take α equal to the expected norm of s, and we take γ equal to the top 100-th quantile[4] of $|\langle 1, \chi_{e'}\rangle|$. We account for the absolute value bars in Equation (5) but ignore the maximization over $0 \leq i < m$. By doing so, we are estimating the *per-coordinate* failure rate: the probability of a failure in the first coordinate of the coefficient embedding.

To first order, an adversary who samples (e_1, e_2) from $\chi_e(u) \times \chi_e(u)$ and who discards all ciphertexts with $|\langle 1, e_3\rangle| < \gamma$ can expect a query set of size $1/(2\,\mathrm{Q}_\alpha(\chi_e(u), \chi_e(u)))$ to include a query that causes a decryption failure (cf. Equation (10)). A classical adversary expects to make approximately $100u^2$ queries to the random oracle per sample. A quantum adversary, using Grover's algorithm, expects to make approximately $10u$ superposition queries to the random oracle per sample.

5.2 Comparison with one-shot failure rate

Before presenting the results of our analysis, we recall that the *one-shot* failure probability is the probability that Equation (3) is violated for $(s_1, s_2) \leftarrow \chi_s \times \chi_s$, $(e_1, e_2) \leftarrow \chi_e \times \chi_e$, and $e_3 \leftarrow \chi_{e'}$. Theorem 3.1 of [18] states that a de-randomized scheme with a one-shot failure rate of δ is $\delta_1 \leq q_G \cdot \delta$ correct against an adversary who generates q_G ciphertexts. Table 1 lists the one-shot failure probabilities

[4] The constant 100 is arbitrary. Our software can produce an optimized value if needed.

for Kyber512, R5ND1PKE0d, Frodo640, R5N11PKE0d, and LightSaber[5]. Each parameter set is advertised as meeting NIST's level 1 security category, so it is reasonable to assume that generating, say, $q_G = 2^{128}$ ciphertexts has lower cost than breaking the scheme. The corresponding values of δ_1 are all larger than 2^{-60}. We find this concerning, as Section 3.3 (resp. Section 4.4 against quantum adversaries) of [18] states potentially large integer multiple of δ_1 in the upper bound on the adversary's success probability in the IND-CCA game.

5.3 Comparison of NIST candidates

The results of our analyses of Kyber512, R5ND1PKE0d, Frodo640, R5N11PKE0d, and LightSaber are shown in Figure 4. There are subtleties to each analysis, but one can largely imagine that the lines on the left and right of Figure 4 plot $u \mapsto 1/(2\,\mathrm{Q}_\alpha(\chi_e(u), \chi_e(u)))$ and $u \mapsto 10u/(2\,\mathrm{Q}_\alpha(\chi_e(u), \chi_e(u)))$ respectively. We give more details in Appendix A.

An adversary who is not constrained in the number of queries that he can submit will minimize cost. As can be seen from Figure 4 and Table 1, after minimizing the cost of the attack, the number of queries in an effective query set ranges from $2^{106.7}$ for LightSaber to $2^{152.1}$ for Kyber512. The attacks differ in cost per query. Of course, an honest user will not answer so many queries.

NIST suggests that "[f]or the purpose of estimating security strengths, it may be assumed that the attacker has access to the decryptions of no more than 2^{64} chosen ciphertexts" [22]. An adversary with this constraint will spend more time per query to improve the efficacy of a smaller query set.

An attacker who can perform a total of 2^{128} quantum operations will perform roughly 2^{64} operations per query and submit 2^{64} queries. Let us briefly assume that our first-order approximation to the efficacy is accurate. Our experiment in the following section provides some indication that the overlap between random queries may be negligible, and supports this assumption. The attaker may then be thought of as randomly sampling from a query set of size $1/(2\,\mathrm{Q}_\alpha(\chi_e(2^{64}), \chi_e(2^{64})))$, which is the right-most point in Figure 4. Let us also briefly assume that the elements of the adversary's query set are equally likely to cause a decryption failure. Under these assumptions, the $\delta(2^{64}, 2^{128})$-correctness of LightSaber is $2^{64}/2^{84.7} = 2^{-20.7}$. This should be compared with the δ_1 correctness of $2^{-0.4}$ that we alluded to in Section 5.2. The $\delta(2^{64}, 2^{128})$-correctness of the other schemes, under the same assumptions, is given in Table 1.

6 Experiments

Both the spherical symmetry heuristic and the accuracy of the first-order approximation to the efficacy need to be examined further. As a first step, we have

[5] Note that our analysis should roughly coincide with the one-shot failure probability when $u = v = 1$. We expect some discrepancy due to our treatment of e_3 and the fact that we fix an estimate, α, for the norm of the secret. In contrast, the one-shot failure probabilities are averaged over all keys.

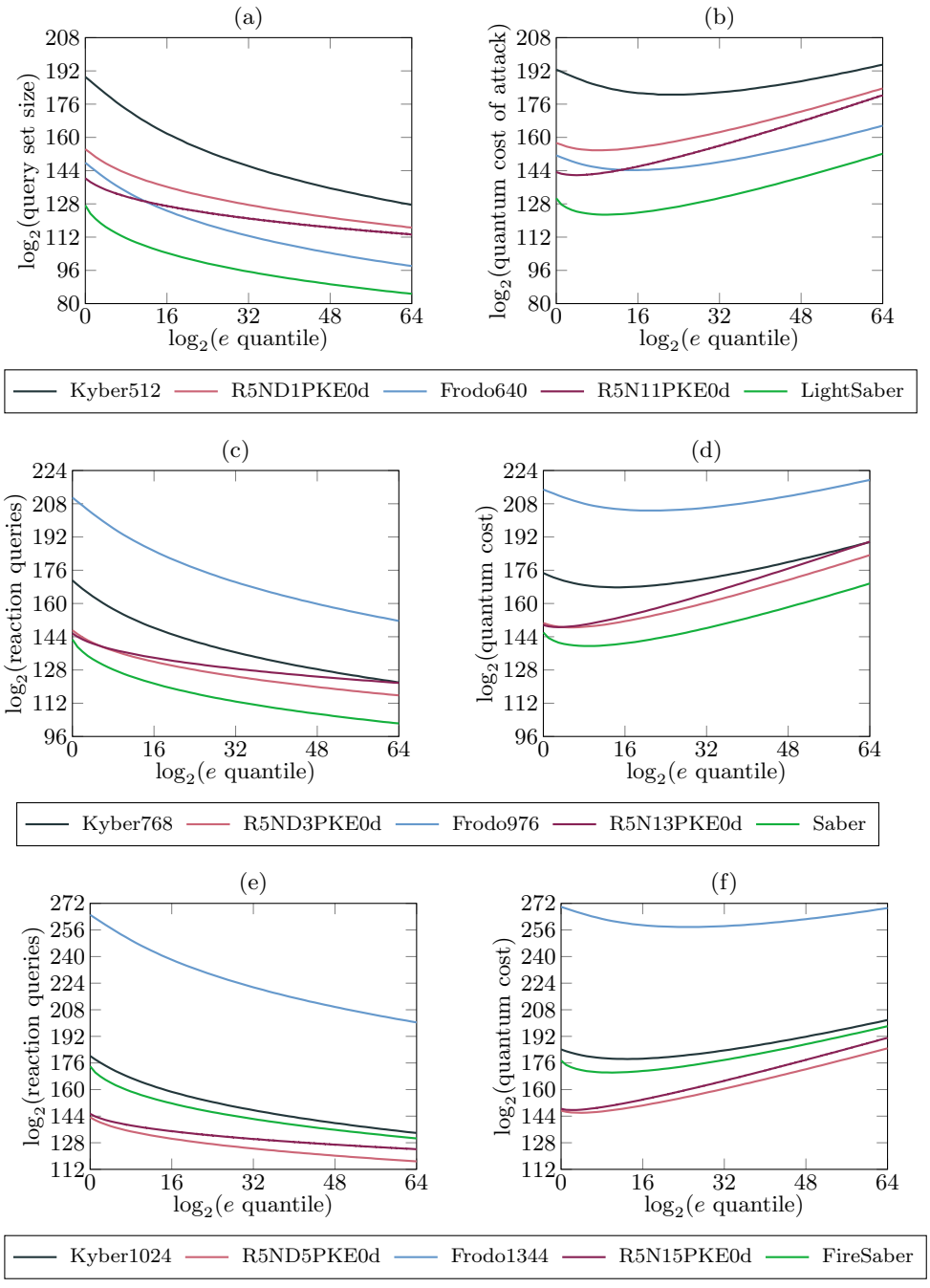

Fig. 4: The predicted size of a query set of unit efficacy (a, c, e) and the quantum cost of producing such a query set (b, d, f). "Quantum cost" is based on Grover's algorithm and has units of "superposition queries to a random oracle". Plots (a) and (b) are NIST level 1 schemes. Plots (c) and (d) are NIST level 3 schemes. Plots (e) and (f) are NIST level 5 schemes. Data embedded in PDF: Kyber512, R5ND1PKE0d, Frodo640, R5N11PKE0d, LightSaber, Kyber768, R5ND3PKE0d, Frodo976, R5N13PKE0d, Saber, Kyber1024, R5ND5PKE0d, Frodo1344, R5N15PKE0d, FireSaber.

	kyber512	frodo640	r5nd1pke	r5n11pke	lightsaber
$\mathcal{A}$	$2^{186.9}$	$2^{144.8}$	$2^{155.1}$	$2^{126.9}$	$2^{128.4}$
$\mathcal{B}$	$2^{187.1}$	$2^{145.8}$	$2^{152.5}$	$2^{138.5}$	$2^{123.3}$
$\mathcal{C}$	$2^{152.1}$	$2^{124.7}$	$2^{142.8}$	$2^{133.9}$	$2^{106.7}$
$\mathcal{D}$	$2^{-63.5}$	$2^{-34.1}$	$2^{-52.6}$	$2^{-49.4}$	$2^{-20.7}$
	kyber768	frodo976	r5nd3pke	r5n13pke	saber
$\mathcal{A}$	$2^{173.2}$	$2^{205.6}$	$2^{131.0}$	$2^{143.9}$	$2^{144.2}$
$\mathcal{B}$	$2^{169.0}$	$2^{209.0}$	$2^{145.3}$	$2^{144.0}$	$2^{139.1}$
$\mathcal{C}$	$2^{141.0}$	$2^{185.3}$	$2^{137.3}$	$2^{139.9}$	$2^{123.7}$
$\mathcal{D}$	2^{-58}	$2^{-87.6}$	$2^{-51.8}$	$2^{-57.8}$	$2^{-38.3}$
	kyber1024	frodo1344	r5nd5pke	r5n15pke	firesaber
$\mathcal{A}$	$2^{183.2}$	$2^{258.7}$	$2^{144.5}$	$2^{127.3}$	$2^{173.4}$
$\mathcal{B}$	$2^{178.1}$	$2^{263.1}$	$2^{141.6}$	$2^{143.8}$	$2^{170.3}$
$\mathcal{C}$	$2^{151.9}$	$2^{238.1}$	$2^{134.8}$	$2^{140.2}$	$2^{154.0}$
$\mathcal{D}$	$2^{-69.9}$	$2^{-136.4}$	$2^{-52.8}$	$2^{-60.2}$	$2^{-66.6}$

Table 1: Adversary $\mathcal{A}$ sends random queries to random users. Adversaries $\mathcal{B}$ and $\mathcal{C}$ target a fixed user that has a random, above-median norm, key. Adversary $\mathcal{B}$ sends queries of above-median norm to the user. Adversary $\mathcal{C}$ sends queries with norm in the top u-quantile for the value of u that minimizes his total quantum cost, i.e. he chooses u based on the local minima in Figure 4 (Plot b, d and f). Adversary $\mathcal{D}$ is restricted to 2^{64} queries and 2^{128} quantum operations. Rows $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ give the expected number of queries that the adversary submits before causing a decryption failure. Row $\mathcal{A}$ is the reciprocal of the one-shot failure probability for a single coordinate. Rows $\mathcal{B}$ and $\mathcal{C}$ are values of $1/(2\,\mathrm{Q}_\alpha(\cdot,\cdot))$. Row $\mathcal{D}$ gives the value of $\delta(2^{64}, 2^{128})$ under the assumptions of Section 5.3. The impact of m and ℓ are suppressed throughout.

performed experiments with a variant of Frodo640. Since the decryption failure rate of Frodo640 is too small for us to observe experimentally, we have used $q = 2^{13}$ rather than $q = 2^{15}$. We have kept the rest of the parameters the same. This variant has a one-shot failure rate of $2^{-11.7}$.

In the notation of Section 5.1, we take α to be the expected value of $\|s\|_2$ when s is drawn from $\chi_s(v) \times \chi_s(v)$. The "Predicted" row in Table 2 gives $1/(2\,\mathrm{Q}_\alpha(\chi_1(u), \chi_1(u)))$. The "Observed" row gives $1/f$ where f is the fraction of failures that we observed.

Frodo640 replaces the $k \times 1$ vectors s_1, s_2, e_1 and e_2 by $k \times 8$ matrices. It replaces the scalar e_3 by an 8×8 matrix. The session key is split across 64 approximately agreed upon scalars. In one run of the experiment, we generate 512 keys and 64 key encapsulations per key. For each encapsulation, we draw 16 samples from $\chi_s(v)$, 16 samples from $\chi_e(u)$, and 64 samples from $\chi_{e'}(100)$. We count

$\|\chi_s\|$ quantile		2^0			2^{10}			2^{20}	
$\|\chi_e\|$ quantile	2^0	2^{10}	2^{20}	2^0	2^{10}	2^{20}	2^0	2^{10}	2^{20}
Predicted	$2^{11.4}$	$2^{9.8}$	$2^{9.1}$	$2^{9.8}$	$2^{8.4}$	$2^{7.8}$	$2^{9.1}$	$2^{7.8}$	$2^{7.3}$
Observed	$2^{11.1}$	$2^{9.4}$	$2^{8.7}$	$2^{9.4}$	$2^{8.0}$	$2^{7.4}$	$2^{8.8}$	$2^{7.4}$	$2^{6.9}$

Table 2: Results of the experiment of Section 6. We did not run the experiment to completion for the columns with $\|\chi_e\| = 2^{20}$. The values reported in those columns are averages over $\approx 2^{18}$, rather than 2^{21}, coordinates.

the total number of coordinates with errors, not the number of encapsulations that fail. In other words f is the fraction of errors observed in $512 \cdot 64 \cdot 64 = 2^{21}$ coordinates.

If $1/(2\,\mathrm{Q}_\alpha(\chi_1(u), \chi_1(u)))$ is a good approximation to the size of an effective query set, and each element of an effective query set is equally likely to cause a failure, then we expect $1/f$ to tend to $1/(2\,\mathrm{Q}_\alpha(\chi_1(u), \chi_1(u)))$ as we average over many keys and encapsulations. As can be seen in Table 2, we observed a fraction of failures such that $f/(2\,\mathrm{Q}_\alpha(\chi_1(u), \chi_1(u))) \approx 2^{-0.4}$ in each case. This provides some indication that our heuristics are reasonable for Frodo640. Further experiments are needed for the other schemes.

7 Conclusion and future work

We have presented a decryption failure attack on the Lindner–Peikert scheme that exploits dependencies between failure events. In contrast with previous attacks, our attack leverages information from adaptive queries. The adversary improves his odds of causing a decryption failure by choosing his next query as a function of his past queries—even those queries that were answered successfully.

Our results do not necessarily call for a re-parametrization of the schemes that we have analyzed. However, like previous analyses of failure boosting, they show that the one-shot failure probability is not a reliable indicator of the difficulty of causing decryption failures. We hope that our work stimulates discussion on what an acceptable $\delta(q_d, t)$-correctness is for various security levels.

Future work. Both the spherical symmetry heuristic and the accuracy of the first-order approximation need further confirmation, either experimentally or theoretically. Beyond this, it is an interesting question to extend our approach to schemes that use error-correction such as ThreeBears [17], NewHope [23], LAC [20], and other parameter sets of Round5 [14]. In a more speculative direction, we wonder whether the information revealed by successful decryptions might be useful in other attacks. Perhaps the knowledge that the secret key does not lie in a particular direction can help an adversary prune an enumeration tree.

The general message that successful queries can leak information about the secret key may be applicable to other constructions as well. Drucker–Gueron–Kostic [11] have already pointed out the risk of ignoring the factor q_{G} loss of tightness in de-randomizing the code-based scheme BIKE [1].

8 Acknowledgements

Special thanks to Kathrin Hövelmanns for insights on the correctness definition for PKEs, Jan-Pieter D'Anvers for helpful discussions and for providing us with a copy of [7], and Steve Weiss for computer systems support. NB is supported by NSERC Discovery Accelerator Supplement grant RGPIN-2016-05146. This work was supported by IQC. IQC is supported in part by the Government of Canada and the Province of Ontario

References

1. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Gueron, S., Guneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P., Zémor, G., Vasseur, V.: BIKE. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
2. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Krauthgamer, R. (ed.) 27th SODA. pp. 10–24. ACM-SIAM (Jan 2016)
3. Bernstein, D.J.: Visualizing size-security tradeoffs for lattice-based encryption. Cryptology ePrint Archive, Report 2019/655 (2019), https://eprint.iacr.org/2019/655
4. Bernstein, D.J., Persichetti, E.: Towards KEM unification. Cryptology ePrint Archive, Report 2018/526 (2018), https://eprint.iacr.org/2018/526
5. D'Anvers, J.P., Guo, Q., Johansson, T., Nilsson, A., Vercauteren, F., Verbauwhede, I.: Decryption failure attacks on IND-CCA secure lattice-based schemes. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 565–598. Springer, Heidelberg (Apr 2019)
6. D'Anvers, J.P., Karmakar, A., Roy, S.S., Vercauteren, F.: SABER. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
7. D'Anvers, J.P., Rossi, M., Virdia, F.: (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. Cryptology ePrint Archive, Report 2019/1399 (2019), https://eprint.iacr.org/2019/1399
8. D'Anvers, J.P., Vercauteren, F., Verbauwhede, I.: On the impact of decryption failures on the security of LWE/LWR based schemes. Cryptology ePrint Archive, Report 2018/1089 (2018), https://eprint.iacr.org/2018/1089
9. D'Anvers, J.P., Vercauteren, F., Verbauwhede, I.: The impact of error dependencies on ring/mod-LWE/LWR based schemes. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 103–115. Springer, Heidelberg (2019)
10. Dent, A.W.: A designer's guide to KEMs. Cryptology ePrint Archive, Report 2002/174 (2002), http://eprint.iacr.org/2002/174

11. Drucker, N., Gueron, S., Kostic, D.: On constant-time QC-MDPC decoding with negligible failure rate. Cryptology ePrint Archive, Report 2019/1289 (2019), https://eprint.iacr.org/2019/1289
12. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (Aug 1999)
13. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)
14. Garcia-Morchon, O., Zhang, Z., Bhattacharya, S., Rietman, R., Tolhuizen, L., Torre-Arce, J.L., Baan, H., Saarinen, M.J.O., Fluhrer, S., Laarhoven, T., Player, R.: Round5. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
15. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008)
16. Guo, Q., Johansson, T., Nilsson, A.: A generic attack on lattice-based schemes using decryption errors with application to ss-ntru-pke. Cryptology ePrint Archive, Report 2019/043 (2019), https://eprint.iacr.org/2019/043
17. Hamburg, M.: Three Bears. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
18. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 341–371. Springer, Heidelberg (Nov 2017)
19. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (Feb 2011)
20. Lu, X., Liu, Y., Jia, D., Xue, H., He, J., Zhang, Z., Liu, Z., Yang, H., Li, B., Wang, K.: LAC. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
21. Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
22. National Institute of Standards and Technology (NIST): Submission requirements and evaluation criteria or the post-quantum cryptography standardization process (2017), available at https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf
23. Poppelmann, T., Alkim, E., Avanzi, R., Bos, J., Ducas, L., de la Piedra, A., Schwabe, P., Stebila, D., Albrecht, M.R., Orsini, E., Osheter, V., Paterson, K.G., Peer, G., Smart, N.P.: NewHope. Tech. rep., National Institute of Standards and Technology (2019), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005)

25. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehlé, D.: CRYSTALS-KYBER. Tech. rep., National Institute of Standards and Technology (2019), available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`

A Details of our analysis for each scheme

A.1 Secret and error distributions

Definition 3 (Modulus switching function). *The modulus switching function is defined by* $[\![x]\!]_q^r = \lfloor x\frac{r}{q} \rceil \bmod r$ *with* $\lfloor x\frac{r}{q} \rceil$ *computed over* $\mathbb{R}$*. It is also extended component-wise to vectors and matrices.*

Definition 4 (Compression artifact distribution). *The compression artifact distribution with parameters* r *and* q *is the distribution of* $y - [\![z]\!]_r^q$ *when* y *is drawn uniformly from* $\mathbb{Z}/q$ *and* $z = [\![y]\!]_q^r$*.*

Definition 5 (Centered binomial distribution). *The centered binomial distribution of parameter* w *assigns probability* $\frac{1}{2^{2w}}\binom{2w}{x+w}$ *to* $x \in \mathbb{Z}$*.*

Definition 6 (Fixed weight distribution). *The fixed weight trinary distribution of parameter* w *in dimension* d *is the uniform distribution on all* $2^w\binom{d}{w}$ *vectors in* $\mathbb{Z}^d$ *that have exactly* $\lceil w/2 \rceil$ *coefficients equal to* $+1$*, exactly* $\lfloor w/2 \rfloor$ *coefficients equal to* -1*, and the remaining* $d - w$ *coefficients equal to* 0*.*

A.2 Compression and learning with rounding

Some variants of the Lindner–Peikert scheme have additional *rounding parameters* r_0, r_1, and r_2. They compress the public key to $(A, [\![b]\!]_q^{r_0})$ and the ciphertext to $([\![c_1]\!]_q^{r_1}, [\![c_2]\!]_q^{r_2})$. Note that if $r_i = q$ then no compression occurs in the corresponding component. If $b' = [\![b]\!]_q^{r_0}$ then there is some $v_1 \in \mathbb{Z}/q$ such that $[\![b']\!]_{r_0}^q = (v_1 - As_2) \bmod q$. Likewise, if $c_1' = [\![c_1]\!]_q^{r_1}$ then there is some $v_2 \in \mathbb{Z}/q$ such that $[\![c_1']\!]_{r_1}^q = (e_1A + v_2) \bmod q$, and if $c_2' = [\![c_2]\!]_q^{r_1}$ then there is some $v_3 \in \mathbb{Z}/q$ such that $[\![c_2']\!]_{r_2}^q = (e_1A + v_3 + \mathsf{encode}(\mathrm{msg})) \bmod q$. Variants that use well chosen rounding parameters can omit the s_1, e_2, and e_3 terms in key generation and encryption; the *compression artifacts* v_1, v_2, and v_3 take their place. Such schemes are said to be based on the Learning With Rounding problem (LWR). The difference between LWE and LWR is immaterial for our purposes; we simply incorporate the compression artifact noise into the distributions of s_1, e_2, and e_3.

A.3 Frodo

Frodo is an instantiation of the Lindner–Peikert scheme with $R = \mathbb{Z}$. The FrodoKEM NIST submission [21] defines three parameter sets `frodo640` ($n = 670$, $q = 2^{15}$, $t = 2^{12}$), `frodo976` ($n = 976$, $q = 2^{16}$, $t = 2^{12}$), and `frodo1344` ($n = 1344$,

$q = 2^{16}$, $t = 2^{11}$). All three use the standard b-bit encoding, and therefore have an error threshold of $t = q/2^{b+1}$. Each parameter set takes $\chi_s = \chi_e = \chi^{\times n}$ where χ is an approximation to a discrete Gaussian distribution on $\mathbb{Z}$. We refer to [21, Table 2] for the exact definition of χ. Our analysis is as described in Section 5.1.

A.4 Kyber (second round)

Kyber is an instantiation of the Lindner–Peikert scheme over $R = \mathbb{Z}[x]/(x^{256}+1)$. The second round NIST submission [25] includes three parameter sets `kyber512` ($m = 256$, $k = 2$, $n = 512$, $q = 3329$, $r_0 = q$, $r_1 = 2^{10}$, $r_2 = 2^3$), `kyber768` ($m = 256$, $k = 3$, $n = 768$, $q = 3329$, $r_0 = q$, $r_1 = 2^{10}$, $r_2 = 2^4$), and `kyber1024` ($m = 256$, $k = 4$, $n = 1024$, $q = 3329$, $r_0 = q$, $r_1 = 2^{11}$, $r_2 = 2^5$). All three use the standard 1-bit encoding. All three parameter sets sample s_1, s_2, e_1, and e_2 from $\eta_2^{\times n}$, where η_2 is the centered binomial distribution of parameter 2.

We write ρ_r for the compression artifact distribution with parameters r and q. We model e_1 as being drawn from $\eta_2^{\times n}$; we model e_2 as being drawn from $(\eta_2 * \rho_{r_1})^{\times n}$; and we model e_3 as being drawn from $(\eta_2 * \rho_{r_2})^{\times m}$. Due to the difference in size between the coefficients of e_1 and e_2, it seems unlikely that the spherical symmetry heuristic is reasonable. We adapt our analysis as follows.

Let $\chi_1 \times \chi_2$ be the distribution from which the adversary draws $e = (e_1, e_2)$. We will assume that χ_1 and χ_2 (viewed as distributions on the coefficient embedding of R^k) are invariant under permutations of the standard basis. Let z_1 and z_2 be the expected values of $\|e_1\|_2$ and $\|e_2\|_2$ respectively. Let $w = \sqrt{z_2/z_1}$, $e^* = (e_1 \cdot w, e_2/w)$, $s^* = (s_1/w, s_2 \cdot w)$, and observe that $\langle \bar{s^*}, e^* \rangle = \langle \bar{s}, e \rangle$. We apply the analysis of Section 5.1, but we take α to be the expected value of $\|s^*\|_2$ and we compute Q_α with respect to the scaled distributions $\chi_1 \cdot w$ and χ_2/w. The expected values of $\|e_1 \cdot w\|_2$ and $\|e_2/w\|_2$ are both $\sqrt{z_1 z_2}$. By assumption on χ_1 and χ_2, this implies that all $2n$ coefficients of e^* have the same expected size. While this does not imply that the distributions are spherically symmetric, it does make the assumption of spherical symmetry more plausible.

A.5 Saber

Saber is a learning with rounding variant of the Lindner–Peikert scheme that uses the base ring $R = \mathbb{Z}[x]/(x^{256}+1)$. The submission proposes three parameter sets `lightsaber` ($m = 256$, $k = 2$, $q = 2^{13}$, $r_0 = 2^{10}$, $r_1 = 2^{10}$, $r_2 = 2^3$, $w = 10$), `saber` ($m = 256$, $k = 3$, $q = 2^{13}$, $r_0 = 2^{10}$, $r_1 = 2^{10}$, $r_2 = 2^4$, $w = 8$), and `firesaber` ($m = 256$, $k = 4$, $q = 2^{13}$, $r_0 = 2^{10}$, $r_1 = 2^{10}$, $r_2 = 2^6$). All three parameter sets sample s_2 and e_1 from the centered binomial distribution of parameter μ, $\eta_\mu^{\times n}$, for the μ listed in [6, Table 1]. Recall that $s_1 = e_2 = e_3 = 0$ for learning with rounding variants.

We write ρ_r for the compression artifact distribution with parameters q and r. The correctness condition can be rewritten as an inner product between $(\bar{v_1}, \bar{s_2})$ and (e_1, v_2), where v_1 is drawn from ρ_{r_0} and v_2 is drawn from ρ_{r_1}. The distributions of v_1 and s_2 are invariant under taking adjoints. Note that $r_0 = r_1$ for

all of the proposed parameter sets. The coefficients of (e_1, v_2) are not identically distributed, so the spherical symmetry assumption is suspect. However, the inner product is unchanged if we write $\bar{s} = (\bar{v_1}, \bar{v_2})$ and $e = (e_1, s_2)$. Moreover, unlike the original vectors, the coefficients of s and e are identically distributed. There is still a slight complication: the adversary has control over one component of s and one component of e. If the adversary chooses particularly large values of e_1 and v_2, then the spherical symmetry assumption will again be violated. We compensate for this by applying the same re-scaling trick from our analysis of Kyber.

A.6 Round5 (R5N1∗PKE_0d)

Round5 is a collection of learning with rounding instantiations of the Lindner–Peikert scheme. The R5N1_∗_PKE_0d parameter sets of Round5 take $R = \mathbb{Z}$. The second round NIST submission includes three parameter sets [14, Table 13] `r5n11pke0d` ($n = 636$, $q = 2^{12}$, $b = 2$, $r_0 = 2^9$, $r_1 = 2^9$, $r_3 = 2^6$, $w = 114$), `r5n13pke0d` ($n = 876$, $q = 2^{15}$, $b = 3$, $r_0 = 2^{11}$, $r_1 = 2^{11}$, $r_3 = 2^7$, $w = 446$), and `r5n15pke0d` ($n = 1217$, $q = 2^{15}$, $b = 4$, $r_0 = 2^{12}$, $r_1 = 2^{12}$, $r_3 = 2^9$, $w = 462$). All three use fixed weight w vectors for e_1 and s_2. Since there are no large values of e_1, the adversary will invest all of his effort in finding large values of v_2, As with Saber, we swap components between vectors and apply the re-scaling trick from our analysis of Kyber. The only difference is that we compute Q_α with respect the honest distribution of e_1 and the u^2-th quantile of $\|v_2\|$.

A.7 Round5 (R5ND∗0d)

The R5ND_∗_0d parameter sets of Round5 take $R = \mathbb{Z}[x]/(1 + x + \cdots + x^m)$. The specification includes three parameter sets [14, Table 11] `r5nd1pke0d` ($m = 586$, $q = 2^{13}$, $b = 1$, $r_0 = 2^9$, $r_1 = 2^9$, $r_3 = 2^4$, $w = 182$), `r5nd3pke0d` ($m = 852$, $q = 2^{12}$, $b = 1$, $r_0 = 2^9$, $r_1 = 2^9$, $r_3 = 2^5$, $w = 212$), and `r5nd5pke0d` ($m = 1170$, $q = 2^{13}$, $b = 1$, $r_0 = 2^9$, $r_1 = 2^9$, $r_3 = 2^5$, $w = 222$). We apply essentially the same analysis as for R5N1_∗_0d. However, the choice of ring presents a slight obstacle as the adjoint does not preserve spherical symmetry.

Multiplication by a fixed element of R, say $a = a_0 + a_1 x + a_2 x^2 + \cdots + a_{m-1} x^{m-1}$, is a linear operation on the coefficient embedding. Specifically, it corresponds to left multiplication by the $m \times m$ matrix $[a]_{i,j} = a_{i-j} - a_{-(j+1)}$ where the index arithmetic is modulo $m+1$ and $a_m = 0$. It follows that the adjoint of multiplication by a is multiplication by $\bar{a}$ where $\bar{a} = a_0 + (a_m - a_{m-1})x + (a_{m-1} - a_{m-2})x^2 + \cdots + (a_1 - a_0)x^{m-1}$. Note that the x^0 and x^1 coefficients are expected to be smaller than the rest. Since only two out of m coefficients are affected, we simply ignore the issue. We re-write the correctness condition as an inner product between $(v_1, \bar{v_2})$ and $(\bar{e_1}, s_2)$. Since e_1 and v_2 have i.i.d. coefficients, we can easily compute the distributions of $\bar{e_1}$ and $\bar{v_2}$.