(One) failure is not an option
Bootstrapping the search for failures in lattice-based encryption schemes

Jan-Pieter D’Anvers\textsuperscript{1*} and Mélissa Rossi\textsuperscript{2,3,4,5†} and Fernando Virdia\textsuperscript{6‡}

\textsuperscript{1} imec-COSIC, KU Leuven, Belgium
janpieter.danvers@esat.kuleuven.be
\textsuperscript{2} ANSSI, Paris, France
\textsuperscript{3} ENS Paris, CNRS, PSL University, Paris, France
\textsuperscript{4} Thales, Gennevilliers, France
\textsuperscript{5} INRIA, Paris, France
melissa.rossi@ens.fr
\textsuperscript{6} Information Security Group, Royal Holloway, University of London, UK
fernando.virdia.2016@rhul.ac.uk

Abstract. Lattice-based encryption schemes are often subject to the possibility of decryption failures, in which valid encryptions are decrypted incorrectly. Such failures, in large number, leak information about the secret key, enabling an attack strategy alternative to pure lattice reduction. Extending the “failure boosting” technique of D’Anvers et al. in PKC 2019, we propose an approach that we call “directional failure boosting” that uses previously found “failing ciphertexts” to accelerate the search for new ones. We analyse in detail the case where the lattice is defined over polynomial ring modules quotiented by $\langle X^N + 1 \rangle$ and demonstrate it on a simple Mod-LWE-based scheme parametrized à la Kyber768/Saber. We show that using our technique, the cost of searching for additional failing ciphertexts after one or more have already been found can be sped up dramatically, thus demonstrating that these schemes should be designed so that it is hard to even obtain one decryption failure.

Keywords: cryptanalysis, lattice-based cryptography, reaction attacks, decryption errors

*The research of D’Anvers was supported the European Commission through the Horizon 2020 research and innovation programme Cathedral ERC Advanced Grant 695305 and the by Semiconductor Research Corporation (SRC), under task 2909.001.
†The research of Rossi was supported by the European Union’s H2020 Programme under PROMETHEUS project (grant 780701) and by ANRT under the programs CIFRE N 2016/1583. It was also supported by the French Programme d’Investissement d’Avenir under national project RISQ P14158.
‡The research of Virdia was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/P009301/1).
1 Introduction

Algebraic lattices are a powerful tool in cryptography, enabling the many sophisticated constructions such as digital signatures [40, 6], zero-knowledge proofs [42, 46], FHE [25], IBE [26], ABE [53], and others. Applications of main interest are public-key encryptions (PKE) [47, 41] and key encapsulation mechanisms (KEM).

The computational problems defined over lattices are believed to be hard to solve, even with access to large-scale quantum computers, and hence many of these constructions are considered to be quantum-safe. As industry starts to make steps forward into the concrete development of small quantum computers [37, 36, 31], the US National Institute of Standards and Technology (NIST) begun an open standardization effort, with the aim of selecting quantum-safe schemes for public-key encryption and digital signatures [44]. At the time of writing, the process is in its second round, and 9 out of 17 candidates for PKE or KEM base their security on problems related to lattices, with or without special structure.

One commonly occurring characteristic of lattice-based PKE or KEM schemes is that of lacking perfect correctness. This means that sometimes, ciphertexts generated honestly using a valid public key may lead to decryption failures under the corresponding private key. Throughout this paper we’ll refer to such ciphertexts as “failures”, “decryption failures”, or “failing ciphertexts”. While in practice, schemes are parametrised in such a way that decryption failures do not undermine overall performance, these can be leveraged as a vehicle for key recovery attacks against the key pair used to generate them. Such an attack was described by Jaulmes and Joux [32] against NTRU, after which is was extended in [30] and [24]. A similar attack on Ring-LWE based schemes was later presented by Fluhrer [21] and extended by Băetu et. al [5].

However, the aforementioned attacks all use specially crafted ciphertexts and can therefore be prevented with a transformation that achieves chosen ciphertext security. This can for example be obtained by means of an off-the-shelf compiler [22, 29] that stops the adversary from being able to freely malleate honestly generated ciphertexts.

The NIST Post-Quantum Standardization Process candidate Kyber [8] noted that it was possible to search for ciphertexts with higher failure probability than average. D’Anvers et al. [15] extended this idea to an attack called “failure boosting”, where ciphertexts with higher failure probability are generated to speedup the search for decryption failures, and provided an analysis of the effectiveness of the attack on several NIST candidates. At the same time, Guo et al. [28] described an adaptive attack against the IND-CCA secure ss-ntru-pke variant of NTRUEncrypt [10], which used an adaptive search for decryption failures exploiting information from previously collected ciphertexts.

Our contributions. In this paper, we present a novel attack technique called “directional failure boosting”, aimed at enhancing the search for decryption failures in public-key encryption schemes based on the protocol by Lyubashevsky et al. [41], in the single-target setting. Our technique is an improvement of the “failure boosting” technique of D’Anvers et al. [15].
Using a simple (but realistically parametrized) scheme based on the Mod-LWE problem, we show that for schemes with a low decryption error probability, the work of finding extra decryption failures can be sped up dramatically when one or more failures have already been found. In particular, the work and number of decryption queries needed to obtain multiple failing ciphertexts using our methodology is only marginally larger than those necessary to obtain the first decryption failure. For example, for our realistic parameters, we show that obtaining 30 decryption failures requires only 25% more quantum work and only 58% more queries than obtaining one decryption failure. As previously shown in [15] and [28], we recall that having many decryption failures enables more efficient lattice reduction which leads to key recovery attacks. As a result, we conclude that when protecting against decryption failure attacks, designers should make sure that an adversary can not feasibly obtain even a single decryption failure.

Our attack outperforms previously proposed attacks based on decryption failures. In particular, it improves over the multitarget attack of Guo et al. [28] on ss-ntru-pke, lowering the attack’s quantum complexity from $2^{139.5}$ to $2^{96.6}$.

Paper outline. In §2, we introduce some preliminaries about notation and structures. In §3, we describe the general idea of lattice-based encryption and how decryption failures are generated. In §4, we recall the original failure boosting technique from [12]. In §5, we describe our directional failure boosting technique. In §6, we show how this method impacts the total work and queries overhead. Finally in §7, we discuss the results by comparing them with the literature and conclude with possible future work.

2 Preliminaries

Let $\mathbb{Z}_q$ be the ring of integers modulo $q$. For $N$ a power of 2, we define $R_q$ the ring $\mathbb{Z}_q[X]/(X^N + 1)$, and $R_{q}^{l_1 \times l_2}$ the ring of $l_1 \times l_2$ matrices over $R_q$. Vectors and polynomials will be indicated with bold lowercase letters, eg. $\mathbf{v}$, while matrices will be written in bold uppercase letters, eg. $\mathbf{M}$. Denote with $\lfloor \cdot \rfloor$ flooring to the nearest lower integer, and with $\lceil \cdot \rceil$ rounding to the nearest integer. These operations are extended coefficient-wise for vectors and polynomials. Throughout, we abuse notation and identify elements in $\mathbb{Z}_q$ with their representatives in $[-q/2, q/2)$, and elements in $R_q$ with their representatives of degree $< N$, with index $i$ indicating the coefficient of $X^i$. This allows us to define the $\ell_2$-norm $\|\mathbf{x}\|_2$ of a polynomial $\mathbf{x} \in R_q$, so that $\|\mathbf{x}\|_2 = \sqrt{\sum x_i^2}$ where $x_i \in [-q/2, q/2)$, and extend this to vectors of polynomials $\mathbf{y} \in R_{q}^{l_1 \times 1}$ as $\|\mathbf{y}\|_2 = \sqrt{\sum_i \|y_i\|_2^2}$. Identically, we define and extend the $\ell_\infty$-norm.

Let $x \leftarrow X$ denote sampling $x$ according to the probability distribution $X$. We extend this notation for coefficient-wise sampling of a vector $\mathbf{x} \in R_{q}^{l_1 \times 1}$ as $\mathbf{x} \leftarrow X(R_{q}^{l_1 \times 1})$, and similarly for a matrix. We denote with $\mathbf{x} \leftarrow X(R_{q}^{l_1 \times 1}; r)$ sampling $\mathbf{x} \in R_{q}^{l_1 \times 1}$ pseudorandomly from the seed $r$ with each coefficient following

\footnote{The software is available at: \url{https://github.com/KULeuven-COSIC/PQCRYPTO-decryption-failures}.}
the distribution $X$. In algorithms, we also use $x \leftarrow \text{Alg}()$ to mean that the value $x$ is assigned to be the output of a probabilistic algorithm $\text{Alg}$.

Let $\mathcal{U}$ be the uniform distribution over $\mathbb{Z}_q$ and let $\mathcal{N}_{\mu,\sigma}$ be the normal distribution with mean $\mu$ and standard deviation $\sigma$, so that the probability density function of $x \leftarrow \mathcal{N}_{\mu,\sigma}$ is defined as:

$$f_{\mathcal{N}_{\mu,\sigma}}(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$  \hfill (1)

The discrete Gaussian distribution $\mathcal{D}_{\mu,\sigma}$ is a discrete restriction to $\mathbb{Z}_q$ of $\mathcal{N}_{\mu,\sigma}$, so that an integer $x$ is sampled with a probability proportional to $e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ and its remainder modulo $q$ in $[-q/2, q/2)$ is returned.

For an event $A$ we define $\Pr[A]$ as its probability. For an element which does not correspond to an event, a ciphertext $ct$ for example, we abusively write $\Pr[ct]$ to denote the probability of the event $ct' = ct$ where $ct'$ is drawn from a distribution which will be clear in the context. We will denote with $\mathbb{E}[A]$ the expected value of a variable drawn from a distribution $A$.

**Security definitions.** Let $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme, with message space $\mathcal{M}$, and let $K = (\text{KeyGen}, \text{Encaps}, \text{Decaps})$ be a key encapsulation mechanism (KEM). When a decapsulation or a decryption oracle is provided, we assume that the maximum number of ciphertexts that can be queried to it for each key pair is $2^K$; in practice, $K = 64$ is often considered [44, §4.A.2]. In this work, we keep the maximum number of queries as a parameter with no specific value, in order to provide a better granularity in the security assessment. Indeed, to mount an attack, the adversary trades off between number of queries and the work.

**Definition 1 (IND-CPA$_{A,\Pi}(k)$ game).** [35] Let $A$ be an adversary and $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme. The experiment IND-CPA$_{A,\Pi}(1^k)$ runs as follows:

1. $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  
2. $A$ is given $pk$. After evaluating $\text{Enc}(pk, \cdot)$ as desired, it outputs $(m_0, m_1) \in \mathcal{M} \times \mathcal{M}$.
3. A random bit $b \leftarrow \{0, 1\}$ is sampled, and $c \leftarrow \text{Enc}(pk, m_b)$ is passed to $A$.
4. $A$ keeps evaluating $\text{Enc}(pk, \cdot)$ as desired, until it returns a bit $b'$.
5. The experiment outputs 1 if $b = b'$ and 0 otherwise.

**Definition 2 (IND-CCA$_{A,K}(k)$ game).** [35] Let $A$ be an adversary and $K = (\text{KeyGen}, \text{Encaps}, \text{Decaps})$ be a key encapsulation mechanism. The experiment IND-CCA$_{A,K}(1^k)$ runs as follows:

1. $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  
2. $(c, k) \leftarrow \text{Encaps}(pk)$
3. $b \leftarrow \{0, 1\}$. If $b = 0$, set $k = k$, else let $\tilde{k} \leftarrow \{0, 1\}^n$.
4. $A$ is given $(pk, c, k)$, and access to a decapsulation oracle $\text{Decaps}(sk, \cdot)$. After evaluating $\text{Encaps}(pk, \cdot)$ and querying $\text{Decaps}(sk, \cdot)$ as desired (except for decapsulation queries on $c$), it returns $b' \in \{0, 1\}$.
5. The experiment outputs 1 if \( b = b' \) and 0 otherwise.

**Definition 3 (PKE and KEM security).** [23] A public-key encryption scheme \( \Pi \) (resp. a key encapsulation mechanism \( K \)) is \((t, \epsilon)\)-GAME secure if for every \( t \)-time adversary \( A \), we have that

\[
\left| \Pr[\text{GAME}_{A,\Pi}(k) = 1] - \frac{1}{2} \right| \leq \epsilon \quad \text{(resp. } \left| \Pr[\text{GAME}_{A,K}(k) = 1] - \frac{1}{2} \right| \leq \epsilon \text{)}
\]

For a security parameter \( 1^k \), we usually mean \( t \approx \text{poly}(k) \) and \( \epsilon \leq \text{negl}(k) \).

If GAME is IND-CPA (resp. IND-CCA) we say that \( \Pi \) (resp. \( K \)) is \((t, \epsilon)\)-secure against chosen-plaintext attacks (resp. \((t, \epsilon)\)-secure against adaptive chosen-ciphertext attacks).

### 3 Lattice-based encryption

The Module-LWE (or Mod-LWE) problem [38] is a mathematical problem that can be used to build cryptographic primitives such as encryption [7, 13], key exchange [13] and signatures [19]. It is a generalization of both the Learning With Errors (or LWE) problem [47], and the Ring-LWE problem [51, 41].

**Definition 4 (Mod-LWE [38]).** Let \( n, q, k \) be positive integers, \( \chi \) be a probability distribution on \( \mathbb{Z} \), and \( s \) be a secret module element in \( \mathbb{R}^k \). We denote by \( \mathcal{L} \) the probability distribution on \( \mathbb{R}^k \times \mathbb{R}^q \) obtained by choosing \( a \in \mathbb{R}^k \) uniformly at random, choosing \( e \in \mathbb{R}^q \) by sampling each of its coefficients according to \( \chi \) and considering it in \( \mathbb{R}^q \), and returning \((a, c) = (a, \langle a, s \rangle + e) \in \mathbb{R}^k \times \mathbb{R}^q\). Decision-Mod-LWE is the problem of deciding whether pairs \((a, c) \in \mathbb{R}^k \times \mathbb{R}^q \) are sampled according to \( \mathcal{L} \) or the uniform distribution on \( \mathbb{R}^k \times \mathbb{R}^q \). Search-Mod-LWE is the problem of recovering \( s \) from \((a, c) = (a, \langle a, s \rangle + e) \in \mathbb{R}^k \times \mathbb{R}^q \) sampled according to \( \mathcal{L} \).

#### 3.1 Passively and actively secure encryption

Lyubashevsky et al. [41] introduced a simple protocol to build passively secure encryption from the Ring-LWE problem, inspired by Diffie-Hellman key exchange [18] and ElGamal public-key encryption [20]. Naturally, the protocol can also be adapted to work based on plain and Module LWE assumptions. A general extension of the protocol for all aforementioned assumptions is described in Algorithms 1, 2, and 3, where \( r \in \mathcal{R} = \{0, 1\}^{256} \), and where the message space is defined as \( \mathcal{M} = \{\text{polynomials in } \mathbb{R}_q \text{ with coefficients in } \{0, 1\}\} \).

In order to obtain active security, designers usually use an off-the-shelf CCA compiler, usually a (post-quantum) variant [17, 29, 52, 48, 33] of the Fujisaki-Okamoto transform [22] (FO). These come with proofs of security in the (quantum) random oracle model, with explicit bounds about the loss of security caused by the transformation. In Appendix E, we show such transformed KEM Decapsulation and Encapsulation algorithms.
In the case of FO for lattice-based schemes, the randomness used during the encryption is generated by submitting the message (and sometimes also the public key) to a random oracle. As this procedure is repeatable with knowledge of the message, one can check the validity of ciphertexts during decapsulation. Hence, an adversary wanting to generate custom ephemeral secrets $s', e', e''$ in order to fabricate weak ciphertexts, would need to know a preimage of the appropriate random coins for the random oracle. Therefore, their only option is to mount a (Grover’s) search by randomly generating ciphertexts corresponding to different messages $m$, and testing if their predicted failure probability is above a certain threshold.

\begin{itemize}
  \item Algorithm 1: PKE.KeyGen()
    \begin{enumerate}
      \item $A \leftarrow \mathcal{U}(R_q^{l \times l})$
      \item $s, e \leftarrow D_{0,\sigma_s}(R_q^{l \times 1}) \times D_{0,\sigma_e}(R_q^{l \times 1})$
      \item $b := As + e$
      \item return $(pk = (b, A), sk = s)$
    \end{enumerate}
  \item Algorithm 2: PKE.Enc($pk = (b, A), m \in \mathcal{M}; r$)
    \begin{enumerate}
      \item $s', e' \leftarrow D_{0,\sigma_s}(R_q^{l \times 1}; r) \times D_{0,\sigma_e}(R_q^{l \times 1}; r)$
      \item $e'' \leftarrow D_{0,\sigma_e}(R_q; r)$
      \item $b' := A^T s' + e'$
      \item $v' := b'^T s' + e'' + \lfloor q/2 \rfloor \cdot m$
      \item return $ct = (v', b')$
    \end{enumerate}
  \item Algorithm 3: PKE.Dec($sk = s, ct = (v', b')$)
    \begin{enumerate}
      \item $m' := \lfloor \lfloor 2/q \rfloor (v' - b'^T s) \rfloor$
      \item return $m'$
    \end{enumerate}
\end{itemize}

Remark 1. Several lattice-based candidates submitted to the NIST Post-Quantum Cryptography Standardization Process use a variant of the protocol by Lyubashevsky et al. [41]. Deviating from the original design, most candidates perform an additional rounding of the ciphertext $v'$, in order to reduce bandwidth. New Hope [3] and LAC [39] choose to work directly over rings (or equivalently, they choose a module of rank $l = 1$) and add error correction on the encapsulated message, while Kyber [7] and Saber [13] choose a module of rank $l > 1$ and perform an additional rounding of $b'$ (and $b$ in case of Saber). For ease of the explanation, we focus on the basic version given in Algorithms 1 to 3.

We selected the parameters of the studied encryption scheme to ensure a similar failure probability and security to Kyber and Saber. These parameters can be found in Table 1. The security estimates are generated using the CoreSVP methodology [3] and the LWE estimator [2], while the failure probability
of Kyber and Saber is given as reported in their respective the NIST round 2 documentations [50, 14]. The failure probability of our chosen parameters is determined by calculating the variance of the error term and assuming the distribution to be Gaussian.

Remark 2. LAC employs a strong error correcting code, to reduce the probability of decryption failures. Hence, our results do not immediately carry over to LAC. We also do not consider the case of “plain” LWE based schemes like FrodoKEM [43] or Round5 [4]. Nonetheless, we believe that the attack methodology would easily translate to the LWE setting as the failure condition and the failure probabilities are similar to the investigated case.

<table>
<thead>
<tr>
<th>Chosen parameters</th>
<th>l</th>
<th>N</th>
<th>q</th>
<th>σₚ</th>
<th>σₑ</th>
<th>P[F]</th>
<th>Classical</th>
<th>Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saber</td>
<td>3</td>
<td>256</td>
<td>8192</td>
<td>2.00</td>
<td>2.00</td>
<td>2⁻¹¹⁹</td>
<td>2¹⁹⁵</td>
<td>2¹⁷⁷</td>
</tr>
<tr>
<td>Kyber 768</td>
<td>3</td>
<td>256</td>
<td>3329</td>
<td>1.00</td>
<td>1.00</td>
<td>2⁻¹⁶⁴</td>
<td>2¹⁸¹</td>
<td>2¹⁶⁴</td>
</tr>
</tbody>
</table>

† Standard deviation of the error term in the public key and ciphertext respectively

Table 1. Comparison between our target scheme and Saber and Kyber 768, as parametrised in Round 2 of the NIST PQC standardization process. The classical (resp. quantum) security is evaluated using the Core-SVP [3] methodology, assuming the cost of BKZ with block size β to be $2^{70.292β}$ (resp. $2^{70.265β}$).

3.2 Decryption failures

Following the execution of the protocol, both messages $m'$ and $m$ are the same if the coefficients of the error term $e'^T s' - s'^T e' + e''$ are small enough; more exactly if $\|e'^T s' - s'^T e' + e''\|_\infty \leq q/4$. This expression can be simplified by defining the vector $S$ as the vertical concatenation of $-s$ and $e$, the vector $C$ as the concatenation of $e'$ and $s'$, and by replacing $e''$ with $G$, as shown below:

$$\begin{align*}
S &= \begin{bmatrix} -s \\ e \end{bmatrix} \\
C &= \begin{bmatrix} e' \\ s' \end{bmatrix} \\
G &= e''.
\end{align*}$$

Here, $S$ contains the secret elements of the secret key, and $C$ and $G$ consist of elements used to construct the ciphertexts\(^8\). Using these vectors, the error expression can be rewritten: a failure occurs when $\|S^T C + G\|_\infty > q/4$.

The standard deviation of the terms in the polynomial $S^T C$ equals $\sqrt{2N\sigma_p\sigma_e}$, versus a standard deviation of $\sigma_e$ for the terms of $G$. Therefore, the influence of $G$ on the failure rate is limited, i.e. $\|S^T C + G\|_\infty \approx \|S^T C\|_\infty$. Let $q_t := q/4$ denote the failure threshold, we will use

$$\|S^T C\|_\infty > q_t$$

\(^8\)When talking about ciphertexts throughout the paper, we will sometimes refer to their underlying elements $C$ and $G$.
Correspondingly, one can rewrite a more accurate Equation 3 as
\[ \| S^T C \|_\infty > q_t - \| G \|_\infty, \]
and instead of considering \( q_t \) to be fixed, taking the distribution of \( q_t - \| G \|_\infty \) as shown in [15]. For the ease of the implementation and due to the low influence of \( G \) on the failure rate, we prefer to stick with Equation 3. We now introduce a more handy way of writing (3) by only using vectors in \( \mathbb{Z}_q \).

**Definition 5 (Coefficient vector).** For \( S \in R_q^{1 \times 1} \), we denote by \( \mathbf{S} \in \mathbb{Z}_q^{N \times 1} \), the representation of \( S \) where each polynomial is decomposed as a list of its coefficients in \( \mathbb{Z}_q \).

**Definition 6 (Rotations).** For \( r \in \mathbb{Z} \) and \( C \in R_q^{l \times 1} \), we denote by \( C^{(r)} \in R_q^{l \times 1} \), the following vector of polynomials
\[
C^{(r)} := X^r \cdot C(X^{-1}) \mod X^N + 1.
\]

Correspondingly, \( \overline{C^{(r)}} \in \mathbb{Z}_q^{l \times 1} \) denotes its coefficient vector.

It is easy to show that \( \overline{C^{(r)}} \) is constructed as to ensure that for \( r \in \{0, \ldots, N-1\} \), the \( r \)th coordinate of \( S^T C \) is given by the scalar product \( \mathbf{S}^T \overline{C^{(r)}} \). In other words, one is now able to decompose \( S^T C \) as a sum of scalar products:
\[
S^T C = \sum_{r \in \{0, N-1\}} S^T \overline{C^{(r)}} \cdot X^r. \quad (4)
\]

One can observe that this construction is only valid for the modulo \( X^N + 1 \) ring structure, but it could be adapted for other ring structures. Note that for any \( r \in \mathbb{Z} \), \( C^{(r+N)} = -C^{(r)} \) and \( C^{(r+2N)} = C^{(r)} \).

**Remark 3.** Note that for any \( r \in \mathbb{Z} \), \( \| \overline{C^{(r)}} \|_2 = \| C \|_2 = \| C \|_2 \) and \( \| \overline{C^{(r)}} \|_\infty = \| C \|_\infty \).

The decomposition in Equation 4 will allow a geometric interpretation of the failures as it will be shown in the rest of the paper. First, let us introduce a brief example to illustrate Definitions 5 and 6.

**Example 1.** For a secret \( S \) and a ciphertext \( C \) in \( \mathbb{Z}_q^{2 \times 1}[X]/(X^3 + 1) \):
\[
S = \begin{bmatrix} s_{0,0} + s_{0,1}X + s_{0,2}X^2 \\ s_{1,0} + s_{1,1}X + s_{1,2}X^2 \end{bmatrix}, \quad C = \begin{bmatrix} c_{0,0} + c_{0,1}X + c_{0,2}X^2 \\ c_{1,0} + c_{1,1}X + c_{1,2}X^2 \end{bmatrix}
\]
we get the following vectors:
\[
S = \begin{bmatrix} s_{0,0} \\ s_{0,1} \\ s_{0,2} \\ s_{1,0} \\ s_{1,1} \\ s_{1,2} \end{bmatrix}, \quad \overline{C^{(0)}} = \begin{bmatrix} c_{0,0} \\ -c_{0,0} \\ -c_{0,1} \\ c_{1,0} \\ -c_{1,2} \\ -c_{1,1} \end{bmatrix}, \quad \overline{C^{(1)}} = \begin{bmatrix} c_{0,1} \\ -c_{0,2} \\ c_{0,0} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \end{bmatrix}, \quad \overline{C^{(2)}} = \begin{bmatrix} c_{0,2} \\ c_{0,1} \\ c_{0,0} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \end{bmatrix}, \quad \overline{C^{(3)}} = \begin{bmatrix} -c_{0,0} \\ c_{0,2} \\ c_{0,1} \\ c_{1,2} \\ c_{1,1} \\ c_{1,0} \end{bmatrix}, \ldots
\]
In case of a failure event, $S^T C$ satisfies Equation 3. Therefore, at least one element among all the coefficients

$$S^T \cdot C^{(0)}, \ldots, S^T \cdot C^{(2N-1)}$$

is larger than $q_t$.

**Definition 7 (Failure event).** A failure event will be denoted with $F$, while we use $S$ to indicate a successful decryption. More precisely, for $r \in [0, 2N - 1]$, we denote by $F_r$ the failure event where

$$S^T \cdot C^{(r)} > q_t.$$  

The event $F_r$ provides the location of the failure in the $S^T C$ polynomial and it also provides the sign of the coefficient that caused the failure.

**An assumption on the failing ciphertexts.** In the rest of the paper, in order to predict the results of our attack, we will make the following orthogonality assumption.

**Assumption 1** Let $n \ll 2Nl$, and $C_0, \ldots, C_n$ be ciphertexts that lead to failure events $F_{r_0}, \ldots, F_{r_n}$. The vectors $C^{(t_0)}, \ldots, C^{(t_n)}$ are considered orthogonal when projected on the hyperplane orthogonal to $S$.

This assumption is an approximation that is supported by the fact that vectors in high dimensional space have a strong tendency towards orthogonality, as can be seen in Figure 2.

## 4 Failure Boosting attack technique

Failure boosting is a technique introduced in [15] to increase the failure rate of (Ring/Mod)-LWE/LWR based schemes by honestly generating ciphertexts and only querying weak ones, i.e. those that have a failure probability above a certain threshold $f_t > 0$. This technique is especially useful in combination with Grover’s algorithm [27], in which case the search for weak ciphertexts can be sped up quadratically. Failure boosting consists of two phases: a precomputation phase, and a phase where the decryption oracle is queried.

**Precomputation phase.** The adversary does an offline search for weak ciphertexts with the following procedure:

1. Generate a key encapsulation $ct = (C, G)$.
2. If $P[F \mid ct] \geq f_t$, keep $ct$ in a weak ciphertext list, otherwise go to Step 1.

In Step 2, $P[F \mid ct]$ is defined as the failure probability given a certain ciphertext $ct$. It is computed as follows.

$$P[F \mid ct] := \sum_S P[\|S^T C + G\|_\infty > q_t \mid S] \cdot P[S]$$  

(6)
Given the probability of generating ciphertexts $P[ct] = P[C, G]$, the probability of finding such a weak ciphertext can be expressed as follows:

$$\alpha_{f_t} = \sum_{\forall ct: P[F|ct] > f_t} P[ct]. \quad (7)$$

An adversary thus needs to perform on average $\alpha_{f_t}^{-1}$ work to obtain one weak ciphertext, or $\sqrt{\alpha_{f_t}^{-1}}$ assuming Grover’s search achieves a full speed-up.

**Decryption oracle query phase.** After the precomputation phase, an adversary has a probability $\beta_{f_t}$ that a weak ciphertext results in a failure, where $\beta_{f_t}$ can be calculated as a weighted average of the failure probabilities of weak ciphertexts:

$$\beta_{f_t} = \frac{\sum_{\forall ct: P[F|ct] > f_t} P[ct] \cdot P[F|ct]}{\sum_{\forall ct: P[F|ct] > f_t} P[ct]}. \quad (8)$$

Thus to obtain one decryption failure with probability $1 - e^{-1}$, an adversary needs to perform approximately $\beta_{f_t}^{-1}$ queries and therefore $\alpha_{f_t}^{-1} \beta_{f_t}^{-1}$ work (or $\sqrt{\alpha_{f_t}^{-1} \beta_{f_t}^{-1}}$ using a quantum computer).

The better an adversary can predict $P[F|ct]$, the more efficient failure boosting will be. Having no information about the secret except its distribution, an adversary is bound to standard failure boosting, where the failure probability is estimated based on $\|C\|_2$ and $\|G\|_2$. For a graphical intuition, a two dimensional toy example is depicted in Figure 1a below, where the red arrow represents the secret vector $S$. Ciphertexts with $C$ that lie in the dashed area will provoke a failure as the inner product with $S$ will exceed the threshold $q_t$. The blue circle is a circle of ciphertexts that have a certain failure probability $f_t$ as estimated by an adversary who does not know the secret. During the failure boosting procedure, we will generate random ciphertexts, and only select the ciphertexts with a higher failure probability than $f_t$, i.e. that are outside the blue circle. One can graphically see in Figure 1a that these ciphertexts will have a higher failure probability and a higher norm. We refer to [15] for a full description of the failure boosting technique. Note that Figure 1a is an oversimplified 2-dimension example that does not take into account the polynomial structure and the high dimensionality of the space.

## 5 Directional Failure Boosting

Once $n \geq 1$ decryption failures $C_0, \ldots, C_{n-1}$ are found, additional information about the secret key $S$ becomes available, and can be used to refine the failure estimation for new ciphertexts and thus speed up failure boosting. We now introduce an iterative two-step method to perform directional failure boosting.

**Step 1** An estimate, denoted $\hat{E}$, of the ‘direction’ of the secret $S$ in $Z_q^{\text{dim}}$ is obtained from $C_0, \ldots, C_{n-1}$. 
Without directional information, as in [15], the weak ciphertexts (in blue) are defined as the ciphertexts with a probability higher than $f_t$. 

With directional information, the weak ciphertexts (in blue) are found according to a refined acceptance criterion, here represented as an ellipse.

**Fig. 1.** Simplified diagram trying to provide an intuition on the effect of directional failure boosting. The red arrow represents the secret vector $\mathbf{S}$. Ciphertexts with $\mathbf{C}$ that lie in the dashed area will provoke a failure as the inner product with $\mathbf{S}$ will exceed the threshold $q_t$. Ciphertexts outside the blue circle are considered weak.

**Step 2** The estimate $\mathbf{E}$ is used to inform the search for weak ciphertexts and improve the failure probability prediction for a new ciphertext $\mathbf{C}_n$. One is able to refine the criterion $P[F|ct] \geq f_t$ with computing $P[F|ct, \mathbf{E}] \geq f_t$ instead.

Once new failing ciphertexts are found in step 2, one can go back to step 1 and improve the estimate $\mathbf{E}$ and thus bootstrap the search for new failures.

To give an intuition, a two dimensional toy representation can be found in Figure 1b. Like in the classical failure boosting technique, the red arrow depicts the secret $\mathbf{S}$, while the additional blue arrow marks estimate $\mathbf{E}$ (as calculated in step 1, see §5.2). Using this estimate, we can refine the acceptance criterion to the depicted ellipse to better reflect our knowledge about the secret (step 2, see §5.3). Ciphertexts outside this ellipse will be flagged as weak ciphertexts, and while the probability of finding such a ciphertext is the same, the failure probability of weak ciphertexts is now higher. As in, more of the blue zone lies in the dashed area.

**5.1 Distributions**

We now introduce some probability distributions that will be useful in following sections.

*Scaled $\chi$-distribution.* The scaled $\chi$-distribution $\chi_{n, \sigma}$ is the distribution of the $\ell_2$-norm of a vector with $n$ coefficients, each following the normal distribution $\mathcal{N}_{0, \sigma}$. Denoting with $\Gamma$ the gamma function, the probability density function of $\chi_{n, \sigma}$ is given by:
\[ f_{\chi_n,\sigma}(x) = \left(\frac{\sigma}{2}\right)^{n-1} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\Gamma\left(\frac{n}{2}\right)} \text{ for } x \geq 0, \quad (9) \]

which has mean \( \mathbb{E}[x] = \sqrt{2\Gamma(n+1)/\Gamma(n/2)} \sigma \approx \sqrt{n}\sigma. \)

We will approximate the probability distribution of \( \|x\|_2 \) where \( x \leftarrow D_{0,\sigma}(R_l \times 1_q) \) with a discretized version of the \( \chi_{(l,N),\sigma} \)-distribution, which will be denoted with \( \chi_{D_{l,N},\sigma} \). Using this distribution, the probability density function of \( \|x\|_2 \) is calculated as:

\[ P[\|x\|_2 = x] = C \left(\frac{x}{\sigma}\right)^{l-N-1} e^{-\frac{x^2}{2\sigma^2}} \text{ for } x \in \left\{0, \ldots, \left\lfloor \frac{q}{2} \sqrt{lN} \right\rfloor \right\}, \quad (10) \]

with \( C \) a normalization constant.

**Angle distribution.** The distribution of angles between \( n \)-dimensional vectors in \( \mathbb{R}^n \) with coefficients drawn from a normal distribution \( N_{0,\sigma} \) can be modelled using the following probability density function [9]:

\[ f_{\Theta_n}(\theta) = \sin^{n-2}(\theta) / \int_0^\pi \sin^{n-2}(t) dt, \quad \text{for } \theta \in [0, \pi]. \]  

Due to the high dimensionality of the vector space used in this paper, vectors will have a very strong tendency towards orthogonality, i.e. \( \theta \) is close to \( \pi/2 \), as can be seen in Figure 2.

![Fig. 2](image-url)  

**Fig. 2.** Probability density function (pdf) of the angle between two random vectors in 1536-dimensional space. As the dimension increases, the pdf tends to the Dirac delta function centered at \( \frac{\pi}{2} \).

For computational reasons, we will use a discretized version \( \Theta_n^D \) of this distribution to model the distribution of the angles between discrete vectors, if no extra directional information is present. Given a uniformly spaced list of angles between 0 and \( \pi \), we assign to each angle a probability

\[ P[\theta] = C \sin^{n-2}(\theta) \]  

with \( C \) a normalization constant. The higher the number of angles in this list, the better this distribution approximates the continuous distribution \( \Theta_n \).
Order statistics. The maximal order statistic of a distribution $X$ in $n$ dimensions, is the distribution of the maximum of $n$ samples drawn from this distribution. We will denote this distribution with $M(X, n)$. For a discrete distribution $X$, the probability mass function of $M(X, n)$ can be computed as:

$$f_{M(X,n)}(x) = P[x \geq y | y \sim X]^n - P[x > y | y \sim X]^n$$

$$\approx n \cdot P[x = y | y \sim X] \cdot P[x > y | y \sim X]^{n-1},$$

where the latter approximation gets better for smaller probabilities.

5.2 Step 1: Estimating the direction $\mathbf{E}$

Informally, $\mathbf{E}$ should be a vector that has approximately the same direction as $\mathbf{S}$. Denoting the angle between $\mathbf{E}$ and $\mathbf{S}$ as $\theta_{ES}$, the bigger $|\cos(\theta_{ES})|$, the closer our estimate is to $\pm \mathbf{S}$ and the better our estimate of failure probability will be. Since we focus on estimating the direction of $\mathbf{S}$, $\mathbf{E}$ will always be normalized.

In this section, we derive an estimate $\mathbf{E}$ of the direction of the secret $\mathbf{S}$ given $n \geq 1$ ciphertexts $C_0, \ldots, C_{n-1}$. Our goal is to find $\mathbf{E}$ such that $|\cos(\theta_{ES})|$ is as big as possible. We will first discuss the case where the adversary has one ciphertext, then the case where she has two, followed by the more general case where she has $n$ ciphertexts.

One ciphertext. Assume that a unique failing ciphertext $C$ is given. For a failure event $F_r$, $\mathbf{E} = \mathbf{C}(r)/\|\mathbf{C}(r)\|_2$ is a reasonable choice as $\cos(\theta_{ES})$ is bigger than average. This can be seen as follows:

$$|\cos(\theta_{ES})| = \frac{\|\mathbf{S}^T \mathbf{E}\|_2}{\|\mathbf{S}\|_2 \|\mathbf{E}\|_2} = \frac{\|\mathbf{S}^T \mathbf{C}(r)\|_2}{\|\mathbf{S}\|_2 \|\mathbf{C}(r)\|_2} > \frac{q_0}{\|\mathbf{S}\|_2 \|\mathbf{C}(r)\|_2}.$$  

Keep in mind that the cosine of angles between random vectors strongly tend to zero in high dimensional space, so that even a relatively small value of $|\cos(\theta_{ES})|$ might be advantageous.

One can argue that it is not possible to compute $\mathbf{C}(r)$ without knowledge of $r$; whereas in the general case, the failure location is unknown. However, $\mathbf{E} = \mathbf{C}(0)/\|\mathbf{C}(0)\|_2$ is an equally good estimate regardless of the value of $r$. Indeed, $\mathbf{C}(0)$ approximates a rotation of the secret $\mathbf{S}' := X^{-r} \cdot \mathbf{S}$ instead of $\mathbf{S}$, which can be seen using the equality $\overline{\mathbf{A}^T \cdot \mathbf{B}} = X^{-r} \cdot \mathbf{S}^T \cdot \overline{\mathbf{C}(0)}$:

$$\mathbf{S}^T \cdot \mathbf{C}(r) = X^{-r} \cdot \mathbf{S}^T \cdot X^{-r} X^r \mathbf{C}(0)$$

$$= X^{-r} \cdot \mathbf{S}^T \cdot \mathbf{C}(0).$$

Furthermore, multiplicating a polynomial in $R_q$ with a power of $X$ does not change its infinity norm, as the multiplication only results in the rotation or negation of coefficients. Thus, using an estimate of the direction of $X^{-r} \cdot \mathbf{S}$ is as good as an estimate of the direction of $\mathbf{S}$ when predicting the failure probability of ciphertexts, and we can use $\mathbf{E} = \mathbf{C}(0)/\|\mathbf{C}(0)\|_2$.  

13
Two ciphertexts. Now, assume that two linearly independent failing ciphertexts \( C_0 \) and \( C_1 \), resulting from failure events \( F_0 \) and \( F_1 \) respectively, are given. Taking \( \bar{E} \) as the normalized version of an average \( C_{\text{av}} = (C_0(0) + C_1(0))/2 \) may not necessarily result in a good estimate. For example, if \( C_0 \) comes from a failure event \( F_0 \) and \( C_1 \) from a failure event \( F_N \), the two directions cancel each other out as the ciphertexts \( C_0(0) \) and \( C_1(0) \) are in opposite directions.

Keeping the convention that \( C_0(0) \) approximates a rotation of the secret \( S' = X^{-r_0} \cdot S \), we will compute the relative error position \( \delta_{1,0} = r_1 - r_0 \) and show that is enough to build a correct estimate \( \bar{E} \) as \( \bar{E} = \frac{C_{\text{av}}}{\|C_{\text{av}}\|_2} \) where:

\[
\bar{C}_{\text{av}} := \left( \frac{C_0(0) + C_1(\delta_{1,0})}{2} \right).
\]

The reason why such \( \bar{E} \) is a good estimator of \( S' \) can be seen as follows:

\[
\cos(\theta_{ES'}) = \frac{1}{2\|C_{\text{av}}\|_2 \|S'\|_2} \cdot \left( X^{-r_0} \cdot S^T \cdot C_0(0) + X^{-r_0} \cdot S^T \cdot X^{r_1-r_0} C_1(0) \right) = \frac{1}{2\|C_{\text{av}}\|_2 \|S'\|_2} \cdot \left( S^{T} \cdot C_0(\delta_{r_0}) + S^{T} \cdot C_1(\delta_{r_1}) \right) > \frac{q_1}{\|C_{\text{av}}\|_2 \|S'\|_2}.
\]

Remark 4. In practice ciphertexts with smaller norm will on average be better aligned with the secret, as \( \cos(\theta_{CS'}) > q_1/(\|C\|_2 \|S'\|_2) \). Therefore they carry more information than ciphertexts with larger norm. To compensate for this effect we will calculate \( \bar{C}_{\text{av}} \) as \( \bar{C}_{\text{av}} := \left( C_0(0)/\|C_0(0)\|_2 + C_1(\delta_{1,0})/\|C_1(\delta_{1,0})\|_2 \right)/2 \). While it is possible to further refine the calculation of \( \bar{E} \) using extra directional information, this heuristic is good enough for our purposes.

Computation of the relative position \( \delta_{1,0} \). One can use the fact that both \( C_0(0) \) and \( C_1(\delta_{1,0}) \) are expected to be directionally close to \( S' \). Thus, the cosine of the angle between \( C_0(0) \) and \( C_1(\delta_{1,0}) \) should be larger than usual. Therefore, \( \delta_{1,0} \) can be estimated with the following distinguisher:

\[
\delta'_{1,0} := \arg\max_{r \in [0,2N-1]} C(r) \quad \text{where} \quad C(r) := \frac{C_0(0)^T \cdot C_1(r)}{\|C_0(0)\|_2 \|C_1(\delta_{1,0})\|_2}.
\]

The next paragraph estimates the efficiency of using Equation 18 as a distinguisher for deriving \( \delta_{1,0} \). We will show that, for Table 1 parameters, we expect

\[
P[\delta'_{1,0} = \delta_{1,0}] \approx 89\%.
\]

Experiments run by simulating the sampling \( 10^4 \) failing ciphertexts (refer to §A for the generation technique), and using Equation 18 for finding \( \delta_{1,0} \) between pairs of them, return \( P_{\text{Exp}}[\delta'_{1,0} = \delta_{1,0}] \approx 84.8\% \), in sufficiently good agreement.
To obtain the value (19), the idea is to estimate the distribution of a correct guess $C(\delta_{1,0})$ and an incorrect guess $\max_{r \neq \delta_{1,0}} C(r)$ and quantify the discrepancy. First, we decompose the ciphertexts in a component parallel to $\mathbf{S'}$, denoted with $\|$, and a component orthogonal, denoted with $\perp$, we rewrite $C(r)$ as follows:

$$C(r) = \frac{C_{0,\|}^{(0)} \cdot C_{1,\|}^{(r)} + C_{0,\perp}^{(0)} \cdot C_{1,\perp}^{(r)}}{\|C_{0,\|}^{(0)}\|_2 \|C_{1,\perp}^{(r)}\|_2}$$

(20)

In the first term, the scalar product of two parallel elements equals the product of their norms (up to their sign). For the second term, we apply the scalar product definition and introduce $t$ as the angle between $C_{0,\perp}^{(0)}$ and $C_{1,\perp}^{(r)}$.

$$C(r) = \pm \frac{C_{0,\|}^{(0)} \cdot C_{1,\|}^{(r)} + C_{0,\perp}^{(0)} \cdot C_{1,\perp}^{(r)}}{\|C_{0,\|}^{(0)}\|_2 \|C_{1,\perp}^{(r)}\|_2} \cdot \cos(t)$$

(21)

$$= \cos \left( \theta_{S'C_0^{(0)}} \right) \cos \left( \theta_{S'C_1^{(r)}} \right) + \sin \left( \theta_{S'C_0^{(0)}} \right) \sin \left( \theta_{S'C_1^{(r)}} \right) \cos(t)$$

(22)

The vectors $\overrightarrow{C_{0,\perp}}$ and $\overrightarrow{C_{1,\perp}}$ are orthogonal to $\mathbf{S'}$. This means that they live in the $2Nl - 1$ dimensional space orthogonal to $\mathbf{S'}$. The high dimension of the space will strongly drive the vectors towards orthogonality as can be seen in Figure 2. Using Assumption 1, the angle $t$ between $\overrightarrow{C_{0,\perp}^{(0)}}$ and $\overrightarrow{C_{1,\perp}^{(r)}}$ is then assumed to follow the distribution of random angles between vectors in a $2Nl - 1$ dimensional space (See Equation 11).

Now, let us study the distribution of $C(r)$ depending of the value $r \in [0, 2N - 1]$. One can refer to Figure 3 for a graphical interpretation based on the parameters of Table 1.

- If $r = \delta_{1,0}$, the expected value of $C(r)$ will be higher than average. Indeed, by definition of $F_{\delta_1}$ and $F_{\delta_0}$ the cosines forming the first term are positive. The distribution of $C(r)$ can then be estimated using Equation 22 (orange curve).
- If $r = \delta_{1,0} + N \mod 2N$, the distribution of $C(r)$ is equal to the distribution of $-C(\delta_{1,0})$ and will be closer to $-1$ (green curve).
- If $r \neq \delta_{1,0} \mod N$, $C(r)$ can be assumed to follow the distribution of random angles in a $2Nl$ dimensional space $\Theta_{2Nl}$, as given in Equation 11 (blue curve).
- The pdf of $\max_{r \neq \delta_{1,0}} C(r)$ is then calculated as $M(\Theta_{2Nl}, 2N - 1)$ by definition of the maximal order statistic (red curve).

Figure 3 assembles the probability density functions of the above distributions in a plot. The probability of selecting the correct $\delta_{1,0}$ using $\arg\max_{r \in [0, 2N - 1]} C(r)$, can then be computed as:

$$P[\delta_{1,0} = \delta_{1,0}] = P[\max_{r \neq \delta_{1,0}} C(r) < C(\delta_{1,0})].$$

For our toy scheme’s parameters, this results in Equation 19.
Multiple ciphertexts. In this section, we assume that \( n \) linearly independent failing ciphertexts \( C_0, \ldots, C_{n-1} \), resulting from failure events \( F_{r_0}, \ldots, F_{r_{n-1}} \) respectively, are given. We introduce a generalized method to recover the relative positions \( \delta_{1,0}, \ldots, \delta_{n-1,0} \), based on “loopy belief propagation” [45]. Once these relative positions are found, they can be combined in an estimate \( E \) with

\[
E = \frac{C_{\text{rav}}}{\|C_{\text{rav}}\|_2}
\]

where

\[
C_{\text{rav}} := \left( \frac{C^{(0)}_0}{\|C^{(0)}_0\|_2} + \sum_{i \in [1, n-1]} \frac{C^{(\delta_{i,0})}_i}{\|C^{(\delta_{i,0})}_i\|_2} \right) / n. \tag{23}
\]

To find the correct rotations, we construct a weighted graph that models the probability of different relative rotations, and we will use loopy belief propagation to obtain the most probable set of these rotations:

- The nodes represent the obtained failing ciphertexts: \( (C_i)_{i \in [0, n-1]} \). In total, there are \( n \) nodes.
- Each node \( C_i \) where \( i \neq 0 \) is associated a list with \( 2N \) probability values called beliefs and denoted \( (b_i(0), \ldots, b_i(2N-1)) \) that together define a probability distribution over \( [0, 2N-1] \). The \( r \)th item in the list represents our belief that the correct relative position \( \delta_{0,1} \) equals \( r \). The correct rotation of the \( 0 \)th node will be fixed to 0 (i.e. \( b_0(0) = 1 \) and \( b_0(i) = 0 \) for all other \( i \)) as only the relative rotations of the ciphertexts is important. These node weights are initialized as follows:

\[
b_i(r) := P[\delta_{i,0} = r] \quad (= P[F_r \text{ for } C_i | F_0 \text{ for } C_0])
\]

- For two nodes \( C_i \) and \( C_j \), the value of the vertex called message, models the influence of the beliefs in the rotations \( s \) of node \( j \) towards the beliefs in rotation \( r \) of node \( i \), which can be formalized as follows:

\[
m_{i,j}(r, s) := P[\delta_{i,j} = r - s] \quad (= P[F_{r-s} \text{ for } C_i | F_0 \text{ for } C_j])
\]
Loopy belief propagation tries to find the node values $r$ for each node, so that the probabilities over the whole graph are maximized. This is done in an iterative fashion by updating the node beliefs according to the messages coming from all other nodes. Our goal is eventually to find $r = \delta_{i,0}$ for each node $i$.

**Example 2.** For example, with $N = 3$ and $n = 3$, the graph contains the nodes $\mathbf{C}_0$, $\mathbf{C}_1$, and $\mathbf{C}_2$. In Figure 4, we represent how such a graph could look like where we arbitrarily instantiate the messages and beliefs. We can see that if one chooses the $r_i = \arg\max_s b_i(r)$ for each node, one would have chosen $r_1 = 1$ and $r_2 = 3$. Nevertheless, we notice that the influence of the other probabilities allows for a better choice (underlined in blue in the figure): $r_1 = 2, r_2 = 3$.

**Fig. 4.** Example of the graph for finding the relative rotations where $N = 3$ and $n = 3$. The beliefs are in the rectangles, the circles represent the nodes and some messages are represented between the nodes.

**Vertex probabilities.** As discussed, the edge between two nodes $\mathbf{C}_i$ with rotation $r$ and $\mathbf{C}_j$ with rotation $s$ is weighted with $P[\delta_{i,j} = r - s]$. This probability can be computed using a generalization of the distinguisher technique used for two ciphertexts as detailed in §D.

**Loopy belief propagation.** This technique looks for the best set $(r_1, \ldots, r_{n-1})$ by iteratively correcting the beliefs using messages from other nodes. This procedure is detailed in Algorithm 4, where normalize$(f)$ normalizes the list $b()$ so that $\sum_{x \in \text{supp}(b)} b(x) = 1$. In each iteration, the belief of each node $\mathbf{C}_i$ is updated according to the messages of the other nodes $\mathbf{C}_j$. For each $i$ the belief is updated as follows:

$$b_i(r) = \prod_{j=0, j \neq i}^{n} \text{infl}_{ji}(r)$$  \hspace{1cm} (24)

where $\text{infl}_{ji}(r)$ captures the influence of the value of the node $\mathbf{C}_j$ to node $\mathbf{C}_i$. This influence can be calculated as $\text{infl}_{ji}(r) \leftarrow C \sum_x m_{i,j}(r, x) \cdot b_j(x)$, with $C$ as normalizing constant.
Algorithm 4: GetRotation()

1 initialization
2 for $i \in [1, n - 1]$ do
3     foreach $r$ do
4         $b_i(r) := P[\delta_{0,i} = r]
5 update phase
6 for # of iterations do
7     for $i \in [1, n)$ do
8         for $j \in [1, n)$ if $i \neq j$ do
9             foreach $r$ do
10                $\text{infl}_{ji}(r) := \sum_s m_{i,j}(r,s) \cdot b_j(s)$ // influence of node $j$ on node $i$
11                normalize($\text{infl}_{ji}$)
12         foreach $r$ do
13             $b_i(r) := \prod_{j=0, j \neq i}^n \text{infl}_{ji}(r)$ // calculate new belief
14         normalize($b_i$)
15 finally
16 for $i \in [1, n)$ do
17     $r_i := \text{argmax}_{r \in [0, 2N-1]} b_i(r)$ // pick the $r_i$ with highest belief
18 return $(r_i)_{i \in [1, n-1]})$

Experimental verification. With Table 1 parameters, we obtained the correct values $r_i = \delta_{i,0}$ for all $i \in [1, n - 1]$ after 3 iterations with the probabilities as reported in Table 2, by generating 1000 times each number of ciphertexts and trying to find the correct values of the $r_i$.

<table>
<thead>
<tr>
<th></th>
<th>2 ciphertexts</th>
<th>3 ciphertexts</th>
<th>4 ciphertexts</th>
<th>5 ciphertexts</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P[r_i = \delta_{i,0} \forall i \in [1, n-1]]$</td>
<td>84.0%</td>
<td>95.6%</td>
<td>&gt; 99.0%</td>
<td>&gt; 99.0%</td>
</tr>
</tbody>
</table>

Table 2. Probability of finding the correct relative rotations and thus building the correct estimate $E$ with the knowledge of 2, 3, 4 and 5 failing ciphertexts.

Remark 5 (Consistency with the previous section). Note that this procedure also incorporates the setting where one has only 2 failing ciphertexts, which would yield exactly the same results as in the previous paragraph.

Finally, once all the rotations are found, recall that the estimate is obtained by $E = \overline{C}_{\text{rav}}/\|\overline{C}_{\text{rav}}\|_2$ where

$$\overline{C}_{\text{rav}} = \left(\frac{C^{(0)}_0}{\|C^{(0)}_0\|_2} + \sum_{i \in [1,n-1]} \frac{C^{(r_i)}_i}{\|C^{(r_i)}_i\|_2}\right)/n.$$ (25)
5.3 Step 2: Finding weak ciphertexts

In this section, we are given an estimate $E$ and we refine the acceptance criterion. Instead of accepting if $P[F|ct] \geq f_t$, our condition is slightly changed.

1. Generate a key encapsulation $ct = (C, G)$ with derived key $K$.
2. If $P[F|E, ct] \geq f_t$, keep $ct$ in a weak ciphertext list, otherwise go to to Step 1.

In Step 2, $P[F|E, ct]$ is defined as the failure probability, given a certain ciphertext $ct$ and a certain estimate $\overline{E}$. In the following, we explain a way to compute it.

First, for $r \in [0, 2N - 1]$, we will estimate the probability that a ciphertext leads to an error in the $r$th location. Decomposing the vectors $S$ and $C$ in a component orthogonal to $E$, denoted with subscript $\perp$, and a component parallel to $E$, denoted with subscript $\parallel$, we obtain the failure expression:

$$P[F_r | E, C] = P[S^T \cdot C^{(r)} > q_t | E, C] = P[S^T \cdot C^{(r)} + S_{\perp}^T \cdot C^{(r)} > q_t | E, C]$$

$$= P \left[ \frac{\cos(t) - \|S\|_2 \|C^{(r)}\|_2}{\|S\|_2 \|C^{(r)}\|_2} \frac{\cos(\theta_{SE}) \cos(\theta_{C^{(r)}E})}{\sin(\theta_{SE}) \sin(\theta_{C^{(r)}E})} \right] > q_t | E, C$$

where $\theta_{SE}$ and $\theta_{C^{(r)}E}$ are the angles of $S$ and $C^{(r)}$ with the estimate $E$ respectively, and where $t$ is the angle between $S_{\perp}$ and $C_{\perp}^{(r)}$. We assume no other knowledge about the direction of the secret apart from the directional estimate $E$. In this case, using Assumption 1, $t$ can be estimated as a uniform angle in a $2N - 1$ dimensional space. Then $t$ is assumed to follow the probability distribution $\Theta_{2N - 1}$ (defined in Equation 11).

The values $\overline{E}$, $\|C\|_2$ and $\cos(\theta_{C^{(r)}E})$ are known, meanwhile the values $\|S\|$ and $\theta_{SE}$ can be modelled using their probability distribution. Thus, we can approximate $P[F_r | \overline{E}, C]$ with $P[F_r | \overline{E}, \|C\|_2, \cos(\theta_{C^{(r)}E})]$.

**Assumption 2** We assume that failures at different locations are independent.

Assumption 2 is a valid assumption for schemes without error correcting codes, as discussed in [16]. We can then calculate the failure probability of a certain ciphertext as:

$$P[F | E, C] = 1 - \prod_{r=0}^{2N} \left( 1 - P[F_r | E_r, \|C\|_2, \cos(\theta_{C^{(r)}E})] \right)$$ (26)

As this expression gives us a better prediction of the failure probability of ciphertexts by using the information embedded in $\overline{E}$, we can more accurately
(Grover) search for weak ciphertexts and thus reduce the work to find the next decryption failure. Moreover, the better \( \mathbf{E} \) approximates the direction of \( \overline{\mathbf{S}} \), the easier it becomes to find a new decryption failure.

5.4 Finalizing the attack with lattice reduction

Once multiple failures are found, the secret key can be recovered with lattice reduction techniques as presented in [16, §4] and in [28, Step 3 of the attack]. The following Section simply outlines how their technique transposes to our framework. As shown in §5, an estimate \( \mathbf{E} \) of the direction of a rotated version of \( \overline{\mathbf{S}} = \mathbf{X}^r \mathbf{S} \) with an unknown value \( r \) is provided. Therefore, similarly to [28], an attacker can obtain an estimation of \( \overline{\mathbf{S}} \) (and not only its direction) by rescaling

\[
\mathbf{E}' := \mathbf{E} \cdot nq_t \cdot \left( \left\| \mathbf{C}_0^{(0)} + \sum_{i \in [1,n-1]} \mathbf{C}_i^{(r_i)} \right\|_2^{-1} \right),
\]

using the approximation \( \mathbf{E}'^T \cdot \frac{1}{n} \left( \mathbf{C}_0^{(0)} + \sum_{i \in [1,n-1]} \mathbf{C}_i^{(r_i)} \right) \approx q_t \).

Then, for each possible \( r \in [0,2N-1] \), an attacker can perform lattice reduction and recover candidates for \( \mathbf{s}, \mathbf{e} \) that are accepted if they verify \( \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \). One caveat is that an attacker may have to run a lattice reduction up to \( 2N \) times. Since \( \mathbf{E}' - \overline{\mathbf{S}} \) is small, the attacker can construct an appropriate lattice basis encoding \( \mathbf{E}' - \overline{\mathbf{S}} \) as a unique shortest target vector, and solves the corresponding Unique-SVP problem with the BKZ algorithm [49, 11, 3, 1]. The block size of BKZ will depend on the accuracy of the estimate \( \mathbf{E} \). Indeed, the standard deviation of \( \mathbf{E}'_i - \overline{\mathbf{S}}_i \) is of the order of \( \sigma_{\mathbf{s}} \cdot \sin(\theta_{\mathbf{S}'E}) \) (assuming that \( \theta_{\mathbf{S}'E} \) is small and \( \|\overline{\mathbf{S}}\|_2 \approx \|\mathbf{E}'\|_2 \)). Thus, when many decryption failures are available, \( \sin(\theta_{\mathbf{S}'E}) \) gets very small and the complexity of this step is dominated by the work required for constructing \( \mathbf{E} \). For example, in the case of our toy scheme, if \( \cos(\theta_{\mathbf{S}'E}) > 0.985 \), using [2], the BKZ block size becomes lower than 363 which leads to less than \( 2^{100} \) quantum work (in the Core-SVP [3] \( 0.265\beta \) model). As we will see in §6.3, this is less than the work required to find the first failure.

Remark 6. One can think that the failures obtained by directional failure boosting will not be totally independent. It is true that the failing ciphertexts are roughly following the same direction. But applying our Assumption 1, in high dimensions, for a reasonable number \( n \) of failures (\( n \ll 2lN \)), the hypercone in which the failures belong is large enough that linear dependency will happen with very low probability.

6 Efficiency of Directional Failure Boosting

In this section, we experimentally verify the efficiency of the directional failure boosting technique. We first quantify the accuracy of the estimate \( \mathbf{E} \) computed according to §5.2. We then derive the necessary work required to run the directional failure boosting technique and the optimal number of queries. For the rest
of the section, we focus on minimizing the total work for finding failures and we will assume there is no upper limit to the number of decryption queries.

Our key takeaway is that, for Table 1 parameters, the more failing ciphertexts have been found, the easier it becomes to obtain the next one, and that most of the effort is concentrated in finding the first failure. The final work and query overheads are stored in Table 4.

### 6.1 Accuracy of the estimate

Let $C_0, \ldots, C_{n-1}$ be $n$ previously found failing ciphertexts and we take the estimate defined according to Equation 25. Similarly to §5.2, we define $S = X^{-r_0} \cdot \mathbf{S}$ as the secret vector for an unknown $F_{r_0}$. To estimate the accuracy of $E$, we compute $\cos(\theta_{S'E}) = \frac{\mathbf{S}'^T \mathbf{E}}{\|\mathbf{S}'\|_2 \|\mathbf{C}_{\text{rav}}\|_2}$ as

\[
\cos(\theta_{S'E}) = \frac{\mathbf{S}'^T \left( \frac{C_{(0)}}{\|C_{(0)}\|_2} + \sum_{i=1}^{n-1} \frac{C_{(r_i)}}{\|C_{(r_i)}\|_2} \right)}{n \|\mathbf{S}'\|_2 \|\mathbf{C}_{\text{rav}}\|_2} \quad (27)
\]

\[
= \cos(\theta_{C_{(0)}S'}) + \sum_{i=1}^{n-1} \cos(\theta_{C_{(r_i)}S'})
\]

\[
= \frac{\frac{C_{(0)}}{\|C_{(0)}\|_2} + \sum_{i=1}^{n-1} \frac{C_{(r_i)}}{\|C_{(r_i)}\|_2}}{\sqrt{n^2 \cos(\theta_{CS'})^2 + n \sin(\theta_{CS'})^2}} \quad (28)
\]

First, we make the following approximation.

**Approximation 1** We approximate the cosine with the secret $\mathbf{S}'$ by its expected value denoted $\cos(\theta_{CS'}) := \mathbb{E}\left[\cos(\theta_{C_{(r_i)}S'})\right]$. In other words, for all $i \in [1, n-1]$ we assume $\cos(\theta_{CS'}) = \cos(\theta_{C_{(r_i)}S'}) = \cos(\theta_{C_{(0)}S'})$.

To estimate the denominator of Equation 28, we split the ciphertexts in a component parallel to the secret $C_{(r_i)}$ and a component orthogonal $C_{(r_i)}$ to the secret. Following Assumption 1, we will assume orthogonality between the various $C_{(r_i)}$. As the norm of the sum of parallel vectors is the sum of their norm, and the norm of the sum of orthogonal vectors can be calculated using Pythagoras’ theorem, we can approximate $\cos(\theta_{S'E})$ as follows:

\[
\cos(\theta_{S'E}) \approx \frac{n \cos(\theta_{CS'})}{\sqrt{n^2 \cos(\theta_{CS'})^2 + n \sin(\theta_{CS'})^2}} = \frac{\cos(\theta_{CS'})}{\sqrt{\cos(\theta_{CS'})^2 + \sin(\theta_{CS'})^2}} \quad (29)
\]

One can see from this equation that $\cos(\theta_{S'E})$ gets closer to 1 when $n$ increases.
Experimental verification. The first line of Table 3 gives the expected values of \( \cos(\theta_{S'E}) \) for various \( n \), according to Equation 29, with \( \cos(\theta_{CS'}) \) set to \( q_t / \|S\|\|C(0)\| \), which is a good approximation of \( \cos(\theta_{CS'}) \) as \( \cos(\theta_{CS'}) > q_t / \|S\|\|C(0)\| \) and because angles tend to orthogonality in high dimensional space.

Then, to verify the theory, we implemented a method to simulate the distribution of random failing ciphertexts. This technique is described in §A. Once the simulated failing ciphertexts are found, we combine them to build \( \overline{E} \) using their correct rotations, and we compute \( \cos(\theta_{S'E}) \). The latter experiment was repeated 100 times and the average values are reported in line two of Table 3.

**Table 3.** Accuracy of the estimate derived from several failures. Expected value of \( \cos(\theta_{S'E}) \) according to Equation 29. The closer to 1, the more accurate \( \overline{E} \) is.

<table>
<thead>
<tr>
<th>( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>theoretical</td>
<td>0.328</td>
<td>0.441</td>
<td>0.516</td>
<td>0.613</td>
<td>0.739</td>
<td>0.841</td>
<td>0.885</td>
<td>0.926</td>
<td>0.963</td>
</tr>
<tr>
<td>experimental</td>
<td>0.318</td>
<td>0.429</td>
<td>0.502</td>
<td>0.600</td>
<td>0.727</td>
<td>0.832</td>
<td>0.878</td>
<td>0.921</td>
<td>0.958</td>
</tr>
</tbody>
</table>

6.2 Estimating \( \alpha_{i,f_t} \) and \( \beta_{i,f_t} \)

To estimate the effectiveness of directional failure boosting given a certain number \( i \) of previously collected failing ciphertexts, we need to find the optimal weak ciphertext threshold \( f_t \) for each \( i \). This corresponds to consider how much time to spend for one precalculation \( \sqrt{\alpha_{i,f_t}^{-1}} \) and the average failure probability of weak ciphertexts \( \beta_{i,f_t} \) after the precalculation. Let us recall the definition of \( \alpha_{n,f} \) and \( \beta_{n,f} \), derived from Equations 7 and 8, where \( C_0, ..., C_{n-1} \) are the \( n \) previously found failing ciphertexts.

\[
\alpha_{i,f_t} = \sum_{\forall ct: P[F|ct, C_0, ..., C_{n-1}] > f_t} P[ct] \tag{30}
\]

\[
\beta_{i,f_t} = \frac{\sum_{\forall ct: P[F|ct, C_0, ..., C_{n-1}] > f_t} P[ct] \cdot P[F|ct, C_0, ..., C_{n-1}] > f_t]}{\sum_{\forall ct: P[F|ct, C_0, ..., C_{n-1}] > f_t} P[ct]} \tag{31}
\]

To find the optimal values, we need to calculate Equation 30 and 31 as functions of \( f_t \). This requires us to list the probability of all ciphertexts \( ct := (C,G) \), and their failure probability \( P[F|ct, C_0, ..., C_{i-1}] \). As discussed in [15], exhaustively computing both values is not practically feasible, and therefore we will make some assumptions to get an estimate.

A first simplification is to cluster ciphertexts that have similar \( \|C\|_2 \) and \( \|\theta_{C(0)E}\| \cdots \|\theta_{C(N-1)E}\| \) and thus a similar failure probability. To further reduce the list size, we only take into account the largest value of \( \cos(\theta_{C(i)E}) \) denoted

\[
\max \cos(\theta_{CE}) := \max_i (\cos(\theta_{C(i)E}))
\]

which results in a slight underestimation of the effectiveness of the attack. In other words,
We combined these results into Figure 5b. These experimental results confirm

\[ P[ct] \text{ becomes } P[\|C\|_2, \max \cos(\theta_{CE})], \]

\[ P[F|ct, C_0, ..., C_{n-1}] \text{ becomes } P[F|\|C\|_2, \max \cos(\theta_{CE})]. \]

Assuming independence between the norm of \( C \) and its angle with \( \mathbf{E} \), \( P[\|C\|_2, \max \cos(\theta_{CE})] \) can be estimated using the distributions defined with Equations 10 and 13 as follows:

\[
P[\|C\|_2, \max \cos(\theta_{CE})] = P[\|C\|_2] \cdot P[\max \cos(\theta_{CE})].
\]  

(32)

Denoting with \( r \) the position of the maximum angle, we can rewrite \( P[F|\|C\|_2, \max \cos(\theta_{CE})] \) as follows:

\[
P[F|\|C\|_2, \max \cos(\theta_{CE})] = 1 - \left( 1 - \prod_{i=1}^{n} \left( 1 - P[F_i|\|C\|_2, \cos(\theta_{C}^{(i)})] \right) \right).
\]  

(33)

\[
= 1 - \left( 1 - P[F_r|\|C\|_2, \cos(\theta_{C}^{(r)})] \right) \prod_{i \neq r} \left( 1 - P[F_i|\|C\|_2, \cos(\theta_{C}^{(i)})] \right).
\]  

(34)

where \( 1 - P[F_r|\|C\|_2, \cos(\theta_{C}^{(r)})] \) can be estimated using Equation 26, and \( P[F_i|\|C\|_2, \cos(\theta_{C}^{(i)})] \leq \cos(\theta_{C}^{(i)}) \) using an integral over Equation 26. The estimated failure probability of ciphertexts given \( \|C\|_2 \) and \( \cos(\theta_{CE}) \) for the parameters listed in Table 1 is depicted in figure 5a.

Verification experiment. We verified these results experimentally by generating \( 5 \cdot 10^6 \) failing ciphertexts and \( 5 \cdot 10^6 \) successful ciphertexts, and calculating their norm and angle with 1000 estimates, or in this case other ciphertexts. The failing ciphertexts were produced using the methodology of §A. Once they are generated, we estimate their failure probability with a procedure detailed in §A. We combined these results into Figure 5b. These experimental results confirm our theoretical estimates given in Figure 5a.

With the estimation of \( P[F|\|C\|_2, \max \cos(\theta_{CE})] \) and \( P[\|C\|_2, \max \cos(\theta_{CE})], \alpha_{i,f_i} \) and \( \beta_{i,f_i} \) can be estimated as functions of \( i \) and \( f_i \). Let us now define the optimal threshold \( f_i \) as a function of \( i \) as:

\[
f_i := \arg\min_{f_i} \left( \sqrt{\alpha_{i,f_i} \cdot \beta_{i,f_i}} \right)^{-1}.
\]

6.3 Total amount of work and queries

In this section, we will derive the optimal work and queries for an adversary to perform, in order to obtain \( n \) ciphertexts with probability \( 1 - e^{-1} \). We introduce the following notation: to find the \( (i+1)^{th} \) ciphertext, the adversary performs \( Q_i \) queries. Using a Poisson distribution, the success probability of finding the \( (i+1)^{th} \) ciphertext in \( Q_i \) queries is \( 1 - e^{-Q_i \beta_{i,f_i}} \). The probability of obtaining
Fig. 5. Failure probability of ciphertexts as a function of $\|C\|_2$ and $\cos(\theta_C)$. A zoomed version of Figure 5a for easier comparison can be found in § B.

$n$ failures can then be calculated as the product of the success probabilities of finding ciphertexts 0 to $n - 1$:

$$P_n = \prod_{i=0}^{n-1} (1 - e^{-Q_i \beta_i, f_i}).$$

(35)

This is a slight underestimation of the success probability of the attack, because if an adversary finds a failing ciphertexts in less than $Q_i$ samples, she can query more ciphertexts in the next stages $i+1, \ldots, n$. However, this effect is small due to the large value of $Q_i$.

The total amount of precomputation quantum work, and the total amount of queries to obtain the $n$ failing ciphertexts by performing $Q_i$ tries for each ciphertext, can be expressed as

$$W_{n}^{\text{tot}} := \sum_{i=0}^{n-1} \frac{Q_i}{\sqrt{\alpha_i, f_i}}; \quad Q_{n}^{\text{tot}} := \sum_{i=0}^{n-1} Q_i.$$

(36)

Recall that for now we assume there is no upper limit to the number of decryption queries that can be made, and we focus on minimizing the amount of work. The values of $Q_i$ that minimizes the total quantum work $W_{n}^{\text{tot}}$ can be found using the following Lagrange multiplier, minimizing the total amount of work to find $n$ failures with probability $1 - e^{-1}$ using the above probability model:

$$L(Q_0, \ldots, Q_{n-1}, \lambda) = \sum_{i=0}^{n-1} \frac{Q_i}{\alpha_i, f_i} + \lambda \left( (1 - e^{-1}) - \prod_{i=0}^{n-1} (1 - e^{-Q_i \beta_i, f_i}) \right)$$

(37)

By equating the partial derivative of $L$ in $Q_0, \ldots, Q_{n-1}$ and $\lambda$ to zero and solving the resulting system of equations, we obtain the optimal values of $Q_0, \ldots, Q_{n-1}$ to mount our attack.
The resulting total work and queries of obtaining \( n \) ciphertext using directional failure boosting are given in Table 4 and Figure 6. One can see that the majority of the work lies in obtaining the first ciphertext, and that obtaining more than one ciphertext can be done in less than double the work and queries, or less than one extra bit of complexity. For schemes with a lower failure probability, failing ciphertexts will be more correlated to the secret, so that the directional information is higher and directional failure boosting will be more effective.

In conclusion, the security of a scheme with low failure probability under a single target decryption failure attack can be approximated by the amount of work and queries that an adversary needs to do in order to obtain the first decryption failure. We emphasize the fact that obtaining many failures for a low overhead threatens the security of the scheme (See §5.4).

<table>
<thead>
<tr>
<th>ciphertexts ( n )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log_2(W_n^{\text{tot}}) )</td>
<td>112.45</td>
<td>112.77</td>
<td>112.78</td>
<td>112.78</td>
<td>112.78</td>
<td>112.78</td>
<td>112.78</td>
</tr>
<tr>
<td>( \log_2(W_n^{\text{tot}}/W_1^{\text{tot}}) )</td>
<td>—</td>
<td>0.32</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
</tr>
<tr>
<td>( \log_2(Q_n^{\text{tot}}) )</td>
<td>102.21</td>
<td>102.86</td>
<td>102.87</td>
<td>102.87</td>
<td>102.87</td>
<td>102.87</td>
<td>102.87</td>
</tr>
<tr>
<td>( \log_2(Q_n^{\text{tot}}/Q_1^{\text{tot}}) )</td>
<td>—</td>
<td>0.65</td>
<td>0.66</td>
<td>0.66</td>
<td>0.66</td>
<td>0.66</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Table 4. Quantum work \( W_n^{\text{tot}} \) and queries \( Q_n^{\text{tot}} \) required to find \( n \) failing ciphertexts with probability \( 1 - e^{-1} \). Finding the first ciphertext requires the heaviest amount of computation. After the third failing ciphertext is found, the following ones are essentially for free.

Fig. 6. Quantum work \( W_i \) and number of decryption queries \( Q_i \) required to find a new failing ciphertext, given the \( i \) failing ciphertexts found previously.

7 Discussion and variants

7.1 Comparison with D’Anvers et al. [15]

In figure 7, the total work and queries needed to obtain \( n \) ciphertexts with probability \( 1 - e^{-1} \) is plotted for both the traditional failure boosting, and our directional failure boosting approach. For more information about our method to estimate the total work and queries with success probability \( 1 - e^{-1} \) using [15], we refer to §5C.
Fig. 7. Quantum work $W_n^{\text{tot}}$ and number of decryption queries $Q_n^{\text{tot}}$ required to obtain $n$ failing ciphertexts with probability $1 - e^{-1}$, given the number of previously found failing ciphertexts.

7.2 Minimizing the number of queries instead

In case there is a maximal number of decryption queries is imposed, say $2^K$, the same attack strategy can be followed. However, to limit the number of queries $Q_n^{\text{tot}}$ necessary in the attack, a stronger preprocessing $\sqrt{\alpha^{-1}_{i,f_t}}$ might be necessary to increase the failure probability $\beta_{i,f_t}$ of weak ciphertexts over $2^{-K}$. The only change to accomplish this is selecting the threshold $f_t$ for each $i$ appropriately. Note that for most practical schemes (e.g. Kyber, Saber, New Hope), increasing the failure probability $\beta_{0,f_t}$ over $2^{-K}$ is not practically feasible or would require too much preprocessing $\sqrt{\alpha^{-1}_{0,f_t}}$.

Figure 8 depicts the amount of work $\sqrt{\alpha^{-1}_{i,f_t} \beta^{-1}_{i,f_t}}$ needed to increase the failure probability $\beta_{i,f_t}$ to a certain failure probability (e.g. $2^{-K}$) for the parameters given in Table 1. The various curves correspond to different numbers of available failing ciphertexts. From this figure, one can see that also in this case, the work is dominated by finding the first decryption failure. Another observation is that the attack gets much more expensive as the maximal number of decryption queries $2^K$ gets smaller.

7.3 Application to ss-ntru-pke and improvement of Guo et al. [28]

In [28], an adaptive multtarget attack is proposed on the ss-ntru-pke version of NTRUEncrypt [10], a Ring-LWE based encryption scheme that claims security against chosen ciphertext attacks. The parameters of this scheme are given in Table 5.

The attack performs at most $2^{64}$ queries on at most $2^{64}$ targets and has a classical cost of $2^{216}$ work, and a quantum cost of $2^{140}$ when speeding up the offline phase with Grover’s search. We adapt directional failure boosting to this attack model and propose both a single and multitarget attack.
Fig. 8. Quantum work $W_n^{\text{tot}}$ required to find a new failing ciphertext, as a function of the decryption failure probability of a Mod-LWE scheme.

$$2^{107} 2^{89} 2^{71} 2^{53} 2^{35} 2^{17}$$

weak ciphertext failure rate $(\gamma)$

$$2^{106} 2^{122} 2^{138} 2^{154} 2^{170} 2^{186} 2^{202}$$

total work to generate a failure $(1/\gamma)$

no extra info

1 ciphertext

2 ciphertext

3 ciphertext

Table 5. Parameters of the ss-ntru-pke [10] scheme.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>$l$</th>
<th>$N$</th>
<th>$q$</th>
<th>$\sigma_s$</th>
<th>$\sigma_e$</th>
<th>$P[F]$</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ss-ntru-pke</td>
<td>1</td>
<td>1024</td>
<td>$2^{30} + 2^{13} + 1$</td>
<td>724</td>
<td>724</td>
<td>$&gt; 2^{-80}$</td>
<td>$2^{198}$</td>
</tr>
</tbody>
</table>

For the single target attack, our proposed methodology in subsection 6.3 needs more than $2^{64}$ queries to obtain a ciphertext. To mitigate this, we increase the precomputational work $\sqrt{\alpha^{-1}}$ so that the failure probability of weak ciphertexts $\beta$ increases over a certain $f_t$, which is chosen as $2^{-57}$ to make sure the total queries are below $2^{64}$. The effect is a bigger overall computation, but a reduction in the number of necessary decryption queries. The rest of the attack proceeds as discussed in Subsection 6.3. The work or queries needed to obtain an extra ciphertexts with $n$ ciphertexts can be seen in Figure 9a. The cost of this single target attack is $2^{139.6}$, which is close to the cost of their multitarget attack $2^{139.5}$, as can be seen in Table 6.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>claimed security</th>
<th>multitarget attack [28]</th>
<th>our single target attack</th>
<th>our multitarget attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>ss-ntru-pke</td>
<td>$2^{198}$</td>
<td>$2^{139.5}$</td>
<td>$2^{139.6}$</td>
<td>$2^{96.6}$</td>
</tr>
</tbody>
</table>

Table 6. Comparison of costs for different attacks against ss-ntru-pke [10].

In the multitarget case, we can use $2^{64}$, $2^{64}$ queries to find the first ciphertext, after which we use the methodology of the single target attack to obtain further ciphertext with limited amount of queries. In this case, the work is dominated by finding the second decryption failure, as we need to do this in under $2^{64}$ queries. The resulting work to obtain an extra ciphertext is depicted in Figure 9b. The cost of this attack is $2^{96.6}$, which is well below the cost of $2^{139.5}$ reported by Guo et al.
Fig. 9. Quantum work $W_{\text{tot}}^n$ and number of decryption queries $Q_{\text{tot}}^n$ required to find a new failing ciphertext for ss-ntru-pke, given the ones found previously.

7.4 Future Work

Application to schemes with error correction. For schemes with strong error correction, such as LAC [39], the error probability of the ciphertext before error correction is relatively high, which results in a lower correlation between the secret and failing ciphertexts. This would complicate the effectiveness of directional failure boosting in several ways: finding the right rotation of ciphertext to combine them into $\mathbf{E}$ becomes less straightforward, and the resulting $\mathbf{E}$ would be less correlated with the secret. However, by finding one failure, an adversary would already have more than one equation due to the multiple errors. Thus the effectiveness of directional failure boosting in these circumstances can not be derived from our results and would be an interesting future research topic.

Multitarget attack with limited number of queries. In case an attacker has multiple possible targets $2^T$ but can perform at most $2^K$ queries at each target, she can initially search for one of more ciphertexts using a small and thus more optimal $\beta_T > 2^{-(T+K)}$, before focusing on one target and limiting to $\beta_T > 2^{-K}$, which requires more preprocessing and possibly more work. In this case, finding the first decryption failure is not necessarily the limiting factor for the adversary. While the attack strategy has been touched upon in subsection 7.3, the optimal attack strategy and its efficiency is outside the scope of this paper, but would be interesting future work.

Acknowledgements. We thank Henri Gilbert and Alessandro Budroni for discussing decryption errors with us, and for providing advice during the writeup of this paper.

References

44. NIST. Submission requirements and evaluation criteria for the Post-Quantum Cryptography standardization process, 2016.
Supplementary material
Appendix A  Modelling decryption failures

As the failure probability of Learning with Errors based schemes is typically very low, it is not feasible to generate decryption failures by generating ciphertexts and subsequently testing them against the secret. In this section, we derive a method to generate failing ciphertexts with approximately the same distribution as the subset of randomly generated ciphertexts that lead to a failure event $F_0$. In other words, we aim at simulating $C$ following the ciphertext distribution and such that

$$\mathbf{S}^T \cdot \overline{C}^{(0)} > q_t.$$ 

While randomly generated ciphertexts can be constructed as vectors with a certain norm $\|C\|_2 \leftarrow \chi_{l \cdot N, \sigma}$ (recall that the norm is independent of the rotation, see Remark 3) and a uniformly random direction, ciphertexts that lead to $F_0$ will have a slightly different $\|C\|_2$ distribution and will be more aligned with the secret $\mathbf{S}$. Furthermore, these two distributions are not independent. One can separate both elements as follows:

$$P[\cos(\theta_{SC^{(0)}}) \cdot \|C\|_2 \mid \mathbf{S}, F_0] = P[\cos(\theta_{SC^{(0)}}) \mid \|C\|_2, \mathbf{S}, F_0] \cdot P[\|C\|_2 \mid \mathbf{S}, F_0].$$  \hspace{1cm} (38)

The norm distribution $P[\|C\|_2 \mid \mathbf{S}, F_0]$ can be simplified as

$$P[\|C\|_2 \mid \mathbf{S}, F_0] = P[\|C\|_2 \mid \|\mathbf{S}\|_2, F_0] = C \cdot P[\|C\|_2 = l \mid l \leftarrow \chi_{l \cdot N, \sigma}] \cdot P[\|\mathbf{S}\|_2] \cdot P[F_0 \mid \|C\|_2, \|\mathbf{S}\|_2]$$

$$= C \cdot P[F_0 \mid \|\mathbf{S}\|_2]^{-1} \cdot P[\|C\|_2 = l \mid l \leftarrow \chi_{l \cdot N, \sigma}] \cdot \frac{q_t}{\|\mathbf{S}\|_2} \cdot t \leftarrow \Theta_l,$$

where $C = P[F_0 \mid \|\mathbf{S}\|_2]^{-1}$ is a normalization constant that is independent of $\mathbf{C}$. The distribution of the angle $\theta_{SC^{(0)}}$ can be approximated as follows:

$$P[\cos(\theta_{SC^{(0)}}) \mid \|C\|_2, \mathbf{S}, F_0] = P[\cos(\theta_{SC^{(0)}}) \mid \|\mathbf{C}\|_2, \|\mathbf{S}\|_2, F_0]$$

$$= P[\cos(\theta_{SC^{(0)}}) \mid \|\mathbf{C}\|_2, \|\mathbf{S}\|_2, F_0]$$

$$= P[\cos(\theta_{SC^{(0)}}) = \cos(t) \mid t \leftarrow \Theta_l \text{ with reject if: } |\cos(t)| < \frac{q_t}{\|\mathbf{C}\|_2 \cdot \|\mathbf{S}\|_2}].$$  \hspace{1cm} (46)

A failing ciphertext can now be approximated by first sampling a $\|\mathbf{C}\|_2$ and then an angle $\theta_{SC^{(0)}}$. At this point we want to generate a vector from the uniformly random distribution of vectors with these exact properties, which can
Algorithm 5: GenFailures()

1. \( l_c \leftarrow \text{Distr}_k \{ P \left[ \| C \|_2 = k \mid S, F_0 \right] \} \) // sample the norm
2. \( \psi \leftarrow \text{Distr}_k \{ P \left[ \cos(\theta_{SC(0)}) = \cos(k) \mid \| C \|_2, S, F_0 \right] \} \) // sample the angle
3. \( U \leftarrow N_0, 1 (Z_2N_l q) \) // sample a vector with random direction
4. \( C = l_c \cdot \left( a \frac{U}{\| U \|_2} + b \frac{S}{\| S \|_2} \right) \) // generate vector with norm \( l_c \) and angle \( \psi \)
5. return \( C(0) \)

be done using Algorithm 5 where \( a \) and \( b \) are chosen so that the angle \( \theta_{SU} \) equals the sampled angle \( \psi \):

\[
a := \frac{\sin(\psi)}{\sin(\theta_{SU})} \tag{47}
\]
\[
b := \cos(\psi) - \frac{\sin(\psi)}{\tan(\theta_{SU})} \tag{48}
\]

The notation \( \text{Distr}_k \) takes the distribution on \( k \) associated with the probabilities for each \( k \).

Appendix B Failure probability experiment

In this experiment, we are given an estimate of the secret \( \hat{E} \) and we want to determine the failure probability of a ciphertext \( C \) as a function of two inputs: its norm \( \| C \|_2 \) (recall that the norm is independent of the rotation, see Remark 3) and its angle with \( \hat{E} \). As in section 6.2, we denote

\[
\cos(\theta_{CE}) := \max_i \cos(\theta_{C(i)E}).
\]

We aim at evaluating \( P[F \mid \| C \|_2, \cos(\theta_{CE})] \). Without loss of generality, we will assume that \( \arg\max_i \cos(\theta_{C(i)E}) = 0 \). As the failure events \( F_0 \) to \( F_{N-1} \) can be considered independent (see Assumption 2) we make the following approximation:

\[
P[F \mid \| C \|_2, \cos(\theta_{CE})] \approx P[F_0 \mid \| C \|_2, \cos(\theta_{CE})] + \sum_{i=1}^{N-1} P[F_i \mid \| C \|_2, \cos(\theta_{CE})] \tag{50}
\]
\[
\approx P[F_0 \mid \| C \|_2, \cos(\theta_{CE})] + \sum_{i=1}^{N-1} P[F_i \mid \| C \|_2]. \tag{51}
\]

Let us first estimate the first term of Equation 51. The latter can be rewritten using Bayes’ theorem as

\[
P[F_0 \mid \| C \|_2, \cos(\theta_{CE})] = P[\| C \|_2, \cos(\theta_{CE}) \mid F_0] \cdot P[F_0]
\]
\[
= P[\| C \|_2, \cos(\theta_{CE}) \mid F_0] \cdot P[F_0] + P[\| C \|_2, \cos(\theta_{CE}) \mid S_0] \cdot P[S_0]. \tag{53}
\]
We discretize the probabilities by dividing the $\|C\|_2 \times \cos(\theta_{CE})$ space into a discrete grid with squares $g_{\|C\|_2, \cos(\theta_{CE})}$. Experimentally, we can determine $P[\|C\|_2, \cos(\theta_{CE}) \mid F_0]$ for a certain roster, by generating $n_f$ failing ciphertexts $C_f$ and $n_s$ successful ciphertexts $C_s$ that fail in the first coefficient and counting the number of matching $\|C\|_2, \cos(\theta_{CE})$ for each square in the grid. Our probability can now be approximated as

$$P[\|C\|_2, \cos(\theta_{CE}) \mid F_0] = P[g_{\|C\|_2, \cos(\theta_{CE})} \mid F_0] \approx \frac{\#(C_f \in g_{\|C\|_2, \cos(\theta_{CE})})}{n_f}.$$  

(54)

Similarly we can say that

$$P[\|C\|_2, \cos(\theta_{CE}) \mid S_0] = P[g_{\|C\|_2, \cos(\theta_{CE})} \mid S_0] \approx \frac{\#(C_s \in g_{\|C\|_2, \cos(\theta_{CE})})}{n_s}.$$  

(55)

Moreover, as the failure probability $P[F_0]$ is extremely low, we assume $P[S_0] = 1 - P[F_0] \approx 1$, so that

$$P[F_0 | \|C\|_2, \cos(\theta_{CE})]$$

$$= \frac{\#(C_f \in g_{\|C\|_2, \cos(\theta_{CE})}) P[F_0]}{\#(C_f \in g_{\|C\|_2, \cos(\theta_{CE})}) P[F_0] + \#(C_s \in g_{\|C\|_2, \cos(\theta_{CE})})},$$

(57)

A similar derivation can be made for $\sum_{i=1}^{N-1} P[F_i | \|C\|_2]$, where the norms are discretized in bands $b_{\|C\|_2}$, which leads to

$$P[F_i | \|C\|_2] = \frac{\#(C_f \in b_{\|C\|_2}) P[F_0]}{\#(C_f \in b_{\|C\|_2}) P[F_0] + \#(C_s \in b_{\|C\|_2})}.$$  

(58)

Using these equations, one can generate a 2D discrete grid with the failure probability for a given ciphertext, as shown in Figure 5b. A zoomed version where domain of the experiments and that of the theoretical values is approximately the same can be found in Figure 10.

**Appendix C  Work and queries derived using the computation technique of [15]**

In the attack model of [15], the attacker does not use any information of already found failing ciphertexts. Hence, the probability of obtaining a certain number of ciphertexts $n$ using $Q$ decryption queries can be calculated using a cumulative binomial distribution as:

$$P_n = 1 - \sum_{i=0}^{n-1} \binom{Q}{i} \beta_{0,f_0}^i (1 - \beta_{0,f_0})^{Q-i},$$

(59)
(a) Theoretical (experimental domain)

Fig. 10. Failure probability of ciphertexts as a function of $\|C\|_2$ and $\cos(\theta_{CE})$. Same values as in Figure 5, with matching domains for an easier comparison.
which can be approximated using the Poisson distribution since \( \beta^{(0)}_{0, f_t} \) is very small and \( Q^{(0)}_{0, f_t} \) is approximately 1. This results in

\[
P_n = 1 - e^{-q_{0, f_0}} \sum_{i=0}^{n-1} \frac{(Q_{0, f_0})^i}{i!}.
\]  (60)

When \( n \ll Q \), the number of queries \( q \) needed to obtain \( n \) failing ciphertexts with probability \( P_n = 1 - e^{-1} \) is then approximately \( Q = n/\beta_{0, f_0} \). Thus, to obtain at least \( n \) ciphertexts with probability \( 1 - e^{-1} \), \( n/\beta_{0, f_0} \) queries and a total precomputational work of \( n/(\sqrt{\alpha_{0, f_0}} \beta_{0, f_0}) \) are required.

**Appendix D  Vertex probabilities**

As discussed in §5.2, the edge between two nodes \( C_i \) with rotation \( r \) and \( C_j \) with rotation \( s \) is weighted with \( m_{i,j}(r, s) = P[\delta_{i,j} = r - s] \). The latter can be detailed as

\[
m_{i,j}(r, s) = P[\delta_{i,j} = r - s] \cos \left( \theta_{C_i^0 C_j^0} \right), \ldots, \cos \left( \theta_{C_i^0 C_j^2 N-1} \right).
\]  (61)

Applying Bayes’ theorem gives

\[
m_{i,j}(r, s) = \frac{P[\delta_{i,j} = r - s] \cdot P[\cos \left( \theta_{C_i^0 C_j^0} \right), \ldots, \cos \left( \theta_{C_i^0 C_j^2 N-1} \right) | \delta_{i,j} = r - s]}{P[\cos \left( \theta_{C_i^0 C_j^0} \right), \ldots, \cos \left( \theta_{C_i^0 C_j^2 N-1} \right)].}
\]  (62)

The prior probability \( P[\delta_{i,j} = r - s] \) is constant for all \( r - s \). Since the denominator does not depend on \( r \) and \( s \), both terms can be grouped into a normalization constant denoted by \( C \) that makes the probabilities over all possible \( r - s \) add to 1:

\[
m_{i,j}(r, s) = C \cdot P[\cos \left( \theta_{C_i^0 C_j^0} \right), \ldots, \cos \left( \theta_{C_i^0 C_j^2 N-1} \right) | \delta_{i,j} = r - s].
\]  (63)

Furthermore, the probabilities of \( \cos(\theta_{C_i^0 C_j^0}), \ldots, \cos(\theta_{C_i^0 C_j^2 N-1}) \) can be considered to be independent, while \( \cos(\theta_{C_i^0 C_j^k}) \) equals \( -\cos(\theta_{C_i^0 C_j^{k+1}}) \). Considering the probability function of \( C(r) \) for \( r = \delta_{i,j} \), which we will denote as \( \phi \), and the probability function of \( C(r) \) for \( r \neq \delta_{i,j} \), which equals \( \Theta_n \), as depicted in Figure 3, we can simplify this expression further as
\( m_{i,j}(r, s) \)

\[
= C \cdot \prod_{k=0}^{N-1} P \left[ \cos \left( \theta_{C_0 j} \right), \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right] \tag{64}
\]

\[
= C \cdot \prod_{k=0}^{N-1} P \left[ \cos \left( \theta_{C_0 j} \right) \right] \left| \delta_{i,j} = r - s \right] \tag{65}
\]

\[
= C \cdot \left[ \prod_{k=0}^{N-1} P \left[ \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right] \right. \tag{66}
\]

\[
= C \cdot \left[ \prod_{k=0}^{N-1} P \left[ \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right] \right. \tag{67}
\]

Replacing the normalization constant \( C \) by the normalization constant \( C' \), where

\[
C' = C \cdot \prod_{k=0}^{N-1} P \left[ \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right] \tag{69}
\]

which is also independent of \( r - s \) and thus does not change the probability distribution, we obtain

\[
m_{i,j}(r, s) = C' \cdot \frac{P \left[ \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right]}{P \left[ \cos \left( \theta_{C_0 j} C_+ \right) \right] \left| \delta_{i,j} = r - s \right]} \tag{70}
\]

Appendix E  IND-CCA KEM algorithms

**Algorithm 6: KEM.Encaps(pk)**

1. \( m \leftarrow U(\{0, 1\}^{256}) \)
2. \((K, r) := G(pk, m)\)
3. \( ct := \text{PKE.Enc}(pk, m, r)\)
4. \( K := H(K, r)\)
5. return \((ct, K)\)
Algorithm 7: \texttt{KEM.Decaps}(sk, pk, ct, K)

\begin{enumerate}
\item $m' := \text{PKE.Dec}(sk, ct)$
\item $(\overline{K}, r') := G(pk, m')$
\item $ct' := \text{PKE.Enc}(pk, m'; r')$
\item if \( ct = ct' \) then
\item \quad \text{return } K := (\overline{K}, r')$
\item else
\item \quad \text{return } K := \perp \quad \text{// Could return a pseudo-random string to implicitly reject}
\end{enumerate}