Achieving secure and efficient lattice-based public-key encryption: the impact of the secret-key distribution

Sauvik Bhattacharya[1], Oscar Garcia-Morchon[1], Rachel Player[2], and Ludo Tolhuizen[1]

[1] Royal Philips N.V., Netherlands. Email: sauvik.bhattacharya@philips.com
[2] Royal Holloway, University of London, UK. rachel.player@rhul.ac.uk

Abstract. Lattice-based public-key encryption has a large number of design choices that can be combined in diverse ways to obtain different tradeoffs. One of these choices is the distribution from which secret keys are sampled. Numerous secret-key distributions exist in the state of the art, including (discrete) Gaussian, binomial, ternary, and fixed-weight ternary. Although the secret-key distribution impacts both the concrete security and the performance of the schemes, it has not been compared in a detailed way how the choice of secret-key distribution affects this tradeoff.

In this paper, we compare different aspects of secret-key distributions from submissions to the NIST post-quantum standardization effort. We consider their impact on concrete security (influenced by the entropy and variance of the distribution), and on decryption failures and IND-CCA2 security (influenced by the probability of sampling keys with "non average, large" norm). Next, we select concrete parameters of an encryption scheme instantiated with the above distributions to identify which distribution(s) offer the best tradeoffs between security and key sizes.

The conclusions of the paper are: first, the above optimization shows that fixed-weight ternary secret keys result in the smallest key sizes in the analyzed scheme. The reason is that such secret keys reduce the decryption failure rate and hence allow for a higher noise-to-modulus ratio, alleviating the slight increase in lattice dimension required for countering specialized attacks that apply in this case. Second, compared to secret keys with independently sampled components, secret keys with a fixed composition (i.e., the number of secret key components equal to any possible value is fixed) result in the scheme becoming more secure against active attacks based on decryption failures.

Keywords: Lattice cryptography · Public-key encryption · Secret keys · Decryption failure · Hybrid attack

1 Introduction

Recent advances in the development of quantum computers [43,61,42,55,41,30] have made a long-standing threat [64] against classical cryptography concrete. At

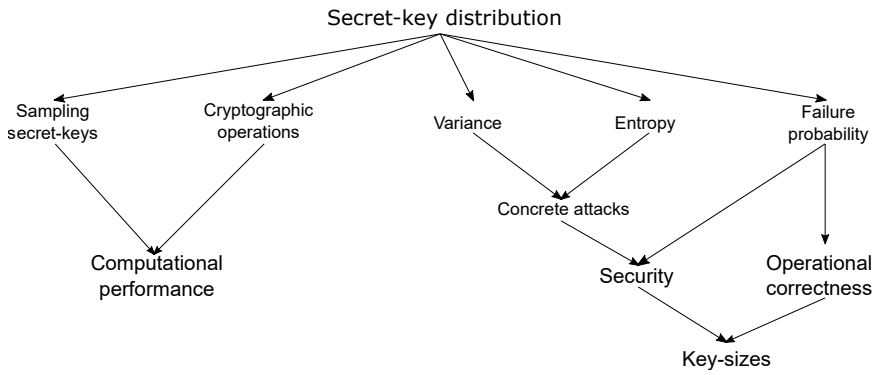

Fig. 1: The impact of a lattice-based encryption/key-encapsulation scheme's secret-key distribution on various aspects of the scheme.

present, it is not clear when a general-purpose quantum computer will become available and this threat will manifest [26,62]. Nevertheless, given the widely recognized difficulties of migrating to new cryptographic infrastructures [32] as well as the long-term confidentiality requirements on data currently exchanged, efforts towards the standardization [56,27] of post-quantum cryptography [16] have already begun. Lattice-based cryptography [57] has received significant attention [19,48] as a candidate for quantum-safe cryptography due to its well-understood mathematical foundations, efficiency, and flexibility. However, the design of lattice-based cryptosystems can be challenging as various tradeoffs and interactions between design aspects must be accounted for. In the design of lattice-based public-key encryption (PKE) schemes, aspects which must be considered include the choice of structure in the underlying lattice [60,52,49], the choice of independent or implicit noise [11], the choice of the noise distribution [17,53,6], and the choice of the secret-key distribution. All these affect the resulting security, operation and performance of the final scheme. Figure 1 summarizes the influence exercised on the scheme by these aspects of the choice of secret-key distribution, which is our focus in this paper.

Secret-key distributions can be characterised by their *variance* and *entropy*, which both impact on security. A secret key with low variance makes concrete attacks on the scheme easier [6,4,2,10]. For example, in attacks utilizing lattice reduction (such as the primal [6] and dual [2] attacks), the secret key is part of a short lattice vector that is recovered as the solution to a lattice problem formulated using the scheme's public key (and ciphertext). These attacks are improved by taking into account the imbalance between the norms of the secret key and error vector [10]. The entropy of the secret-key distribution is relevant for combinatorial attacks (such as the hybrid attack [39], and the sparse variant of the dual attack [2]) in which part of the secret key is recovered by guessing. Secret keys with lower entropy are easier to guess.

The secret-key distribution influences *chosen-ciphertext or active attacks* [28] that exploit decryption failures, because the secret key is directly involved in

Table 1: Performance comparison of a Ring Learning with Rounding [11]-based public-key encryption scheme, considering different secret-key distributions (details in Sec. 4.3). For concrete attacks considered, Q assumes a quantum speedup, while C does not.

Parameters	**Fixed-weight ternary secrets**	**Symmetric ternary secrets**	**Discrete Gaussian secrets**	**Fixed-composition Binomial secrets**	**Binomial secrets**
$\eta, \theta, h, \sigma^2$	1, -, [163], 0.41	1, 0.41, -, 0.41	-, -, -, 0.3	3, -, [194, 77, 12], 1.5	3, 1, -, 1.5
n, q, p, t	796, 2^{13}, 2^9, 2^4	796, 2^{13}, 2^9, 2^5	820, 2^{13}, 2^9, 2^4	828, 2^{14}, 2^{10}, 2^4	828, 2^{14}, 2^{10}, 2^4
Bandwidth	**1937 B**	1961 B	1991 B	2215 B	2215 B
Public key	921 B	921 B	948 B	1060 B	1060 B
Encryption overhead	1016 B	1040 B	1043 B	1155 B	1155 B
Failure rate	2^{-173}	2^{-187}	2^{-172}	2^{-187}	2^{-172}
Primal attack [6] (Q/C)	$\mathbf{2^{175}/2^{192}}$	$\mathbf{2^{175}/2^{192}}$	$\mathbf{2^{176}/2^{194}}$	$2^{176}/2^{194}$	$2^{176}/2^{194}$
Dual attack [2] (Q/C)	$2^{176}/2^{194}$	$2^{176}/2^{194}$	$2^{178}/2^{196}$	$\mathbf{2^{174}/2^{192}}$	$\mathbf{2^{174}/2^{192}}$
Hybrid attack [39] (Q/C)	$2^{183}/2^{193}$	$2^{183}/2^{195}$	$2^{314}/2^{328}$	$2^{253}/2^{271}$	$2^{253}/2^{271}$
Sparse-secrets attack [2] (Q/C)	$2^{175}/2^{192}$	$2^{175}/2^{192}$	$2^{176}/2^{194}$	$2^{176}/2^{194}$	$2^{176}/2^{194}$

decryption and thus affects the probability of such failure events occurring. An attacker who witnesses these failure events can build up statistical information on the secret-key, making recovery of the secret easier [23,33]. Finally, the secret-key distribution also affects the *computational performance* of the scheme, through sampling of keys and (polynomial or matrix) multiplications.

1.1 Our contributions

In this work, we focus on the choice of the secret-key distribution when designing a lattice-based public-key encryption scheme, analyzing how this choice affects the scheme's security, operation and performance. Our contributions are:

1. We compare a number of secret-key distributions used in NIST post-quantum candidates [56] with respect to different criteria such as variance, entropy and resulting probability of decryption failure.
2. We analyze the performance of a lattice-based public-key encryption scheme for the above secret-key distributions. We show in Table 1 (details in Section 4.3) that *fixed-weight ternary secret keys* lead to minimum bandwidth requirements, while remaining secure. Despite allowing specialized attacks, such secrets also allow stronger noise tolerance, increasing the noise-to-modulus ratio and security. This combination leads to smaller keys. Our findings agree with previous recommendations for NTRU-based schemes [37,15], but consider a wider range of secret-key distributions and underlying lattice assumptions.
3. We extend the analysis in [23] on the impact of decryption failures on the chosen-ciphertext (IND-CCA2) security of lattice-based encryption schemes. We show in Figure 2 (details in Section 5) that using fixed-weight ternary secrets, rather than ternary secrets with independently drawn components, makes the scheme less prone to attacks of the above form, since sampling

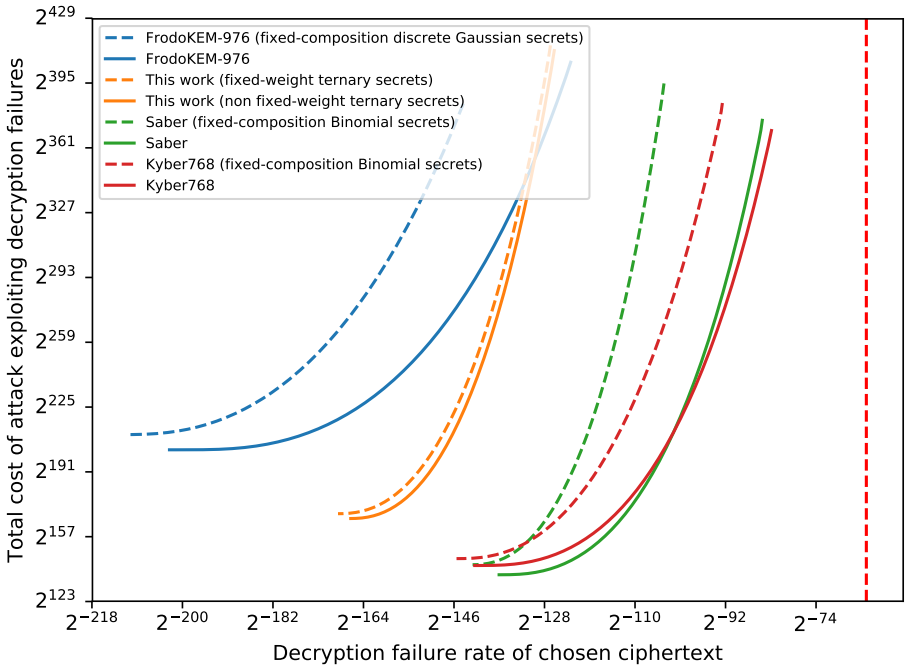

Fig. 2: Total work for a chosen-ciphertext attack on lattice-based encryption schemes by boosting decryption failures as in [23], considering fixed and non fixed-weight secrets. The scheme in this work is similar to a version of the Round5 [9] scheme without error correction (details in Sec. 4.2 and 5.2). The vertical red, dashed line indicates a failure rate of 2^{-64}.

larger-than-expected secrets is impossible. Moreover, we show the same more generally is true for fixed-composition secrets, that is, secrets for which the number of components of each possible value is fixed, and that this is independent of the scheme's error distribution.

1.2 Related work

In the realm of lattice-based public-key encryption, multiple types of secret-key distributions have been proposed in the literature. Ternary secrets have been proposed for NTRU schemes. For example, in NTRU Encrypt [37] these are recommended over binary secrets. In the paper introducing NTRU Prime [15], it is stated that a fixed-weight ternary secret-key distribution appears to improve key-sizes for the same level of security, compared to a wider distribution such as a discrete Gaussian. The potential benefits of fixing the weight for ternary distributions have also been discussed in [12,13].

At the same time, other lattice-based public-key encryption schemes, such as [18], [17], [5] and [24], propose the usage of binomial or Gaussian distributions, citing their stronger resistance against specialized attacks such as [39] or the ease of obtaining tighter proofs [17]. Indeed, the lattice-based proposals that have progressed to the second round of the NIST post-quantum standardization process sample their secret-keys from a wide variety of distributions including discrete Gaussian, centered binomial, symmetric-ternary and fixed-weight ternary secrets. Three of these submissions, viz. LAC [50,51], NTRU Prime [14] and Round5 [9], employ fixed-weight ternary secrets.

The above discussion shows that the trade-offs in security and performance offered by various distributions have not been quantified in a systematic way and a consensus has not been reached by the community. Moreover, to the best of our knowledge, no work provides a detailed and thorough analysis of the role of the secret-key distribution in these tradeoffs.

The possibility of an attacker searching for ciphertexts that lead to a higher than expected probability of decryption failure in lattice-based public-key encryption was proposed by Alperin-Sheriff [7] and Hamburg [35], as part of an analysis of LAC [50]. This was further analyzed by D'Anvers *et al* in [23]. As a possible countermeasure to the above attack, Hamburg [35] suggested to fix the Hamming weight in the LAC cryptosystem, but a rationale or full analysis was not provided. Concurrent to the present work, some techniques we propose – namely, fixing the (Hamming) weight of the secret-keys (and error vectors) – were independently used to stop the above attack [7,35] in an updated version of LAC [51, Section 1] submitted to the second round of the NIST process. However, the authors do not analyze how this technique stops the attack, nor do they generalize it to other schemes and distributions.

In the context of fully homomorphic encryption [29] schemes, design choices implicitly account for the security and performance tradeoffs resulting from the use of secrets having low variance and/or low entropy, which are essential for controlling the noise growth in such schemes. A ternary [63], or fixed-weight ternary [34], secret-key distribution is typically chosen in implementations. The Homomorphic Encryption Security Standard [1] accordingly recommends secure parameters for several choices of secret-key distribution, including ternary.

1.3 Organization

Section 2 introduces preliminaries and notation. Section 3 describes the secret-key distributions considered in this work. Section 4 first analyzes and compares the entropies, variances and probabilities of decryption failure for the different secret-key distributions in Sections 4.1 and 4.2. Next, Section 4.3 analyzes the (bandwidth) performance of a lattice (specifically, rounding)-based encryption scheme instantiated with the various secret-key distributions we consider, showing that fixed-weight ternary keys lead to the smallest bandwidth requirements. Section 5 analyzes the influence of the secret-key distribution on chosen-ciphertext (IND-CCA2) attacks that use decryption failures, showing that fixing

the weight or number of secret key components makes such attacks harder. Section 6 concludes the paper.

2 Preliminaries

2.1 Notation

For each positive integer a, we denote the set of congruences modulo a by $\mathbb{Z}_a$. We identify $\mathbb{Z}_a$ with the set $\{0, 1, \ldots, a-1\}$. For a set A, we denote by $a \xleftarrow{\$} A$ that a is drawn uniformly at random from A. For any polynomial $f(x)$, let $\mathcal{R}_f$ denote the polynomial ring $\mathbb{Z}[x]/f(x)$. For each positive integer a, we write $\mathcal{R}_{f,a}$ for the polynomials of degree less than that of $f(x)$, with all coefficients in $\mathbb{Z}_a$. We call a polynomial *ternary* if all its coefficients are 0, 1 or -1. Throughout this document, regular font letters denote elements from a $\mathcal{R}_f$ defined for a polynomial $f(x)$. For any polynomial, its Hamming weight h is defined as its number of non-zero coefficients. For $x \in \mathbb{Q}$, $\lfloor x \rceil$ denotes rounding to the closest integer (with rounding up in case of a tie). This operation is extended to polynomials coefficient-wise.

2.2 Cryptographic, problem and scheme definitions

We follow the notation used in [23]. A public-key encryption (PKE) scheme is defined as a triple of functions PKE = $(\texttt{Keygen}, \texttt{Enc}, \texttt{Dec})$ with message space $\mathcal{M}$, where given a security parameter λ `Keygen` returns a secret key sk and public key pk, `Enc` encrypts a message $m \in \mathcal{M}$ using pk to produce a ciphertext ct, and `Dec` returns an estimate m' of m given ct and sk.

The decisional Learning with Errors (LWE) [59] problem involves distinguishing the uniform sample $(\boldsymbol{A}, \boldsymbol{U}) \leftarrow \mathcal{U}(\mathbb{Z}_q^{k_1 \times k_2} \times \mathbb{Z}_q^{k_1 \times m})$ from the LWE sample $(\boldsymbol{A}, \boldsymbol{B} = \langle \boldsymbol{AS} + \boldsymbol{E} \rangle_q)$ where $\boldsymbol{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{k_1 \times k_2})$ and where the secret key $\boldsymbol{S}$ and error $\boldsymbol{E}$ are generated from the secret and error distributions $\chi_s(\mathbb{Z}_q^{k_2 \times m})$ and $\chi_e(\mathbb{Z}_q^{k_1 \times m})$ respectively. The search problem is to recover $\boldsymbol{S}$ from the LWE sample.

As mentioned in [23], the above problem definitions can be generalized to Ring [52] or Module [49] (R/M)LWE by using vectors of polynomials. To further generalize the definition, independent reduction polynomials $f_1(x)$ and $f_2(x)$ can be considered, the first used to reduce the product of polynomial multiplications during key-generation and the second used similarly during encryption and decryption. The NIST PQC candidate Round5 [9] uses such a construction with different reduction polynomials. Then, the generalized problem is to distinguish the uniform sample $(\boldsymbol{A}, \boldsymbol{U}) \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{k_1 \times k_2} \times \mathcal{R}_{f_1,q}^{k_1 \times m})$ from a generalized LWE sample $(\boldsymbol{A}, \boldsymbol{B} = \langle \boldsymbol{AS} + \boldsymbol{E} \rangle_{f_1})$ where $\boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{k_1 \times k_2})$, $\boldsymbol{S} \leftarrow \chi_s(\mathcal{R}_q^{k_2 \times m})$ and $\boldsymbol{E} \leftarrow \chi_e(\mathcal{R}_q^{k_1 \times m})$. The search problem is analogous to the LWE case.

The decisional generalized Learning with Rounding (LWR) [11] problem involves distinguishing the uniform sample $(\boldsymbol{A}, \lfloor p/q \cdot \boldsymbol{U} \rceil)$ where $\boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{k_1 \times k_2})$

Algorithm 1: Keygen()

$\boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{l \times l})$
$\boldsymbol{S}_A \leftarrow \chi_s(\mathcal{R}_q^{l \times m})$, $\boldsymbol{E}_A \leftarrow \chi_e(\mathcal{R}_q^{l \times m})$
$\boldsymbol{B} = \left\lfloor p/q \cdot \langle \boldsymbol{A}\boldsymbol{S}_A + \boldsymbol{E}_A \rangle_{f_1} \right\rceil$
return $(pk = (\boldsymbol{A}, \boldsymbol{B}), sk = \boldsymbol{S}_A)$

Algorithm 2: Encrypt$(pk = (\boldsymbol{A}, \boldsymbol{B}), m)$

$\boldsymbol{S}'_B \leftarrow \chi_s(\mathcal{R}_q^{l \times m})$, $\boldsymbol{E}'_B \leftarrow \chi_e(\mathcal{R}_q^{l \times m})$
$\boldsymbol{E}''_B \leftarrow \chi_e(\mathcal{R}_q^{m \times m})$
$\boldsymbol{B}_r = \lfloor q/p \cdot \boldsymbol{B} \rceil$
$\boldsymbol{B}' = \left\lfloor p/q \cdot \langle \boldsymbol{A}^T \boldsymbol{S}'_B + \boldsymbol{E}'_B \rangle_{f_1} \right\rceil$
$\boldsymbol{V}' = \left\lfloor t/q \cdot \langle \boldsymbol{B}_r^T \boldsymbol{S}'_B + \boldsymbol{E}''_B + \frac{q}{2}\texttt{encode}(m) \rangle_{f_2} \right\rceil$
return $ct = (\boldsymbol{B}', \boldsymbol{V}')$

Algorithm 3: Decrypt$(sk = \boldsymbol{S}_A, ct = (\boldsymbol{B}', \boldsymbol{V}'))$

$\boldsymbol{m}' = \left\lfloor \frac{2}{q}(\lfloor q/t \cdot \boldsymbol{V}' \rceil - \langle \lfloor q/p \cdot \boldsymbol{B}' \rceil \, \boldsymbol{S}_A \rangle_{f_2}) \right\rceil$
return $\texttt{decode}(m')$

and $\boldsymbol{U} \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{k_1 \times m})$ from the generalized LWR sample $(\boldsymbol{A}, \boldsymbol{B} = \left\lfloor p/q \cdot \langle \boldsymbol{A}\boldsymbol{S} \rangle_{f_1} \right\rceil)$ where $\boldsymbol{A} \leftarrow \mathcal{U}(\mathcal{R}_{f_1,q}^{k_1 \times k_2})$, and $\boldsymbol{S} \leftarrow \chi_s(\mathcal{R}_q^{k_2 \times m})$. Analogous to the LWE case, the search problem is to recover $\boldsymbol{S}$ from the generalized LWR sample.

Using the above generalized problem definitions, we define a generalized public-key encryption scheme in Algorithms 1, 2 and 3 similar to [23, Sec. 2.4].

Note the use of an additional *ciphertext compression modulus* t in addition to the primary modulus q and the rounding modulus p. The function `encode` transforms a message $m \in \mathcal{M}$ into a polynomial representation, and `decode` is the inverse decoding function. As proposed in [23, Sec. 2.4], this generalized PKE framework can be instantiated to describe multiple NIST PQC schemes that are based on LWE/LWR [17,9], RLWE/RLWR [6,9] or MLWE/MLWR [18,25].

3 State of the art: Secret-key distributions

This section presents definitions of the secret-key distributions analyzed in this paper. A number of, but not all, of these distributions feature in second-round NIST post-quantum cryptographic (PQC) candidates [56], this is summarized in Table 2. We start with distributions of secrets of length n obtained by drawing each of the n components independently from one single distribution. Then we describe distributions in which secrets are generated as a whole.

3.1 Discrete Gaussian distribution

For schemes based on the (Ring [52]) Learning with Errors [59] ((R)LWE) problem, security reductions from worst-case lattice problems are feasible if the

Table 2: NIST PQC candidates featuring secret-key distributions analyzed in this work.

Secret-key distribution	NIST PQC candidate
Discrete Gaussian (Sec. 3.1)	Frodo [17]
Centered binomial (Sec. 3.2)	Kyber [18], Saber [25], NewHope [5], LAC [50]
Symmetric ternary (Sec. 3.3)	NTRU [40]
Fixed-weight ternary (Sec. 3.4)	Round5 [9], NTRUPrime [14], 2$^{\text{nd}}$ round LAC [51]

noise follows a sufficiently wide Gaussian distribution [59,20], such as the Frodo scheme [17]. Variants in which the secret follows the same distribution as the error can be proven equivalent to the original problem by putting the system in systematic form, as done in [8]. Therefore, for schemes based on (R)-LWE, components of the secrets commonly (approximately) follow a discrete Gaussian distribution. For implementation reasons, the approximations have a finite support [17]. The discrete Gaussian probability distribution function $D_{\mathbb{Z},\sigma}$ over $\mathbb{Z}$ with mean $\mu = 0$ and parameter σ is defined as

$$D_{\mathbb{Z},\sigma}(X = k) = \frac{1}{S} e^{-k^2/2\sigma^2}. \tag{1}$$

Here X is the random variable over $\mathbb{Z}$, and S is the normalization constant $\sum_{k=\infty}^{\infty} e^{-k^2/2\sigma^2}$. For $\sigma \geq 0.5$, it holds that $\text{var}(D_{\mathbb{Z},\sigma}) \approx \sigma^2$.

3.2 Centered binomial distribution

The centered binomial distribution was introduced in [6] as an easy-to-implement distribution that is a good approximation to a rounded continuous Gaussian distribution with the same variance. For each positive integer η, the centered binomial distribution bin_η of width η has support $\{-\eta, -\eta+1, \ldots, -1, 0, 1, \ldots, \eta\}$ and is defined as

$$\text{bin}_\eta(k) = \binom{2\eta}{k+\eta} 2^{-2\eta} \text{ for } k \in [-\eta, \eta] \cap \mathbb{Z}. \tag{2}$$

Clearly, bin_η is symmetric around zero and so has mean zero. By direct computation, it can be shown that the variance of this distribution is $\text{var}(\text{bin}_\eta) = \frac{\eta}{2}$.

Sampling from bin_η can be done [6] by computing $\sum_{i=0}^{\eta-1}(b_i - b'_i)$ where the $b_i, b'_i \in \{0, 1\}$ are uniform independent bits. The NewHope submission to the NIST standardization [58] uses bin_8 for generating noise and secrets. The Kyber [18] and Saber [24] submissions employ bin_η with [3] $\eta \in \{6, 8, 10\}$ for generating secrets in their three proposed parameter sets. The LAC [50] submission employs two distributions based on bin_1 for generating secrets and noise in its proposed parameter sets.

[3] The two submissions use different notations for the parameter η.

Scaled version. We define the scaled centered binomial distribution $\mathrm{bin}_{\eta,\theta}$ of width η and with scaling factor θ as

$$\mathrm{bin}_{\eta,\theta}(k) = \theta \cdot \mathrm{bin}_{\eta}(k) \text{ for } k \in \mathbb{Z}, 1 \leq |k| \leq \eta, \text{ and } \mathrm{bin}_{\eta,\theta}(0) = 1 - \theta(1 - \mathrm{bin}_{\eta}(0)). \tag{3}$$

In order that $\mathrm{bin}_{\eta,\theta}$ is a probability distribution, it is required that $0 \leq \theta \leq 1/(1 - \mathrm{bin}_{\eta}(0))$. As $\mathrm{bin}_{\eta,\theta}$ is symmetric around zero, its mean equals zero. Its variance satisfies

$$\mathrm{var}(\mathrm{bin}_{\eta,\theta}) = \sum_{k \neq 0} k^2 \theta \cdot \mathrm{bin}_{\eta}(k) = \theta \cdot \mathrm{var}(\mathrm{bin}_{\eta}) = \frac{1}{2}\theta \cdot \eta. \tag{4}$$

By varying over both η and θ, the scaled centered binomial distribution allows a wide range of trade-offs to be investigated. It has the centered binomial distribution (as $\mathrm{bin}_{\eta,1} = \mathrm{bin}_{\eta}$) as a special case. We note that this generic distribution is not actually used in any NIST PQC candidate.

3.3 Symmetric ternary distribution

For $0 \leq \alpha \leq 1$, the symmetric ternary distribution $\mathcal{T}_{\alpha}$ with parameter α is defined as

$$\mathcal{T}_{\alpha}(0) = 1 - \alpha, \;\; \mathcal{T}_{\alpha}(1) = \mathcal{T}_{\alpha}(-1) = \frac{1}{2}\alpha. \tag{5}$$

Clearly, $\mathcal{T}_{\alpha}$ has mean zero and variance α. The Lizard submission to the NIST standardization [22] employs $\mathcal{T}_{\frac{1}{2}}$ and $\mathcal{T}_{\frac{1}{4}}$ for secret key generation in Lizard.CCA and Lizard.KEM. This distribution is another special case of the scaled centered binomial distribution defined in Section 3.2 (as $\mathrm{bin}_{1,\theta} = \mathcal{T}_{\frac{1}{2}\theta}$).

3.4 Fixed-weight ternary distribution

This distribution is not defined via a component-wise distribution, rather, the entire secret is generated as a whole, as in the NIST PQC candidates Round5 [9] and NTRUPrime [15], and very recently in the version of the candidate LAC introduced in the second round of the NIST post-quantum standardization process [51]. For positive integers n, h with h even and $1 \leq h \leq n/2$, the fixed-weight ternary distribution $\mathcal{T}_{n,h}$ is the uniform distribution on the set of all ternary vectors with $h/2$ ones, $h/2$ minus ones, and $n - h$ zeroes.

There is a close relationship between fixed-weight ternary secrets and secrets generated according to a symmetric ternary distribution. Indeed, each component of a vector drawn according to $\mathcal{T}_{n,h}$ has distribution $\mathcal{T}_{h/n}$. Specifically, the per-component variance of vectors drawn according to $\mathcal{T}_{n,h}$ equals $\frac{h}{n}$. Conversely, by the law of large numbers, for large n, a vector of length n with each component drawn independently according to $\mathcal{T}_{\alpha}$ with high probability has approximately $\frac{1}{2}\alpha n$ ones, $\frac{1}{2}\alpha n$ minus ones, and $n(1 - \alpha)$ zeroes.

However, as will be shown in Section 5, a certain active attack that utilizes decryption failures [23] is more powerful against secrets with independently generated components according to $\mathcal{T}_{h/n}$ than against secrets generated according to

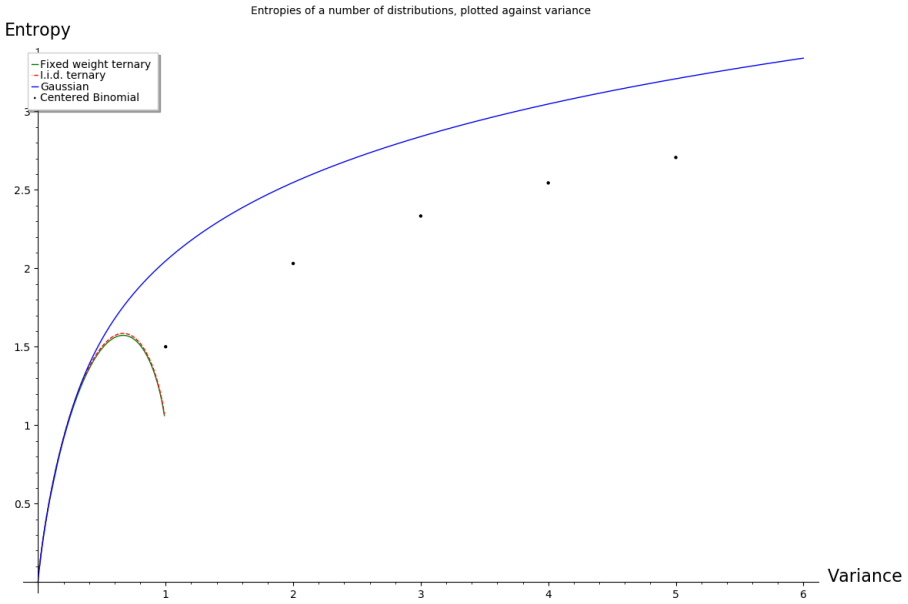

Fig. 3: Comparison of the entropies and variances of the distributions considered in this work; solid green: fixed-weight ternary (with dimension 800 and weight of secret keys ranging from 8 till 792), dotted red: symmetric ternary with independently sampled components), blue (discrete Gaussian), black (centered binomial).

$\mathcal{T}_{n,h}$. The reason is that such an attacker can benefit from the (rare) occurrence of secrets of a weight considerably larger than h.

4 Analysis

We begin with a comparison of two fundamental properties of the distributions considered in Section 4.1, namely their entropies and variances. Next, we look more deeply into the interaction between a secret-key distribution and the underlying cryptographic scheme, and compare in Section 4.2 the probability of decryption failures when different secret-key distributions are considered. In Section 4.3, we compare the key sizes of a lattice-based public-key encryption scheme when instantiated with the different secret-key distributions considered in this work.

4.1 Comparing entropy against variance

In this section, we study the per-symbol entropy of the secret-key distributions considered in this work, for a fixed variance. We note that the symmetric ternary (Section 3.3) and fixed-weight ternary (Section 3.4) distributions cannot have a variance larger than one. On the other hand, the centered binomial distribution

(Section 3.2) cannot have a variance smaller than one. Finally, the discrete Gaussian distribution (Section 3.1) can be parametrized to have variances in both of these regimes.

The comparison of these distributions with respect to entropy achieved for a fixed variance is shown in Figure 3. The per-symbol entropy of the fixed-weight ternary distribution $\mathcal{T}_{n,h}$ is obtained as follows. Clearly, there are exactly $\binom{n}{h}\binom{h}{h/2}$ ternary vectors of length n containing exactly $h/2$ $+1$'s and exactly $h/2$ -1's. As $\mathcal{T}_{n,h}$ is the uniform distribution on this set of ternary vectors, it has entropy $\log_2\left(\binom{n}{h}\binom{h}{h/2}\right)$, and hence a per-symbol entropy of $\frac{1}{n}log_2\left(\binom{n}{h}\binom{h}{h/2}\right)$. Stirling's approximation implies that for large n, the per-symbol entropy of $\mathcal{T}_{n,h}$ is very close to the entropy of $\mathcal{T}_{n/h}$.

In Figure 3, we fix $n = 800$, and compute the variance as h/n, and entropy as above for Hamming weights h ranging from 8 till 792. It can be seen that in the so-called "low-variance" regime (i.e., for variances less than approximately 0.4), the ternary distributions achieve entropies *almost as high as* that of the discrete Gaussian distribution, which is known to maximize the entropy for a given variance [44]. Such regimes can be imagined to be desirable for low probabilities of decryption failure, since a low variance implies a lower probability of (a) large component(s) being sampled in the secret key that can increase the failure probability. The ternary distributions may have other benefits, namely more efficient cryptographic computations and easier sampling, that we do not discuss here.

For such low variances however, the centered binomial distribution cannot be defined and thus cannot be compared with either the discrete Gaussian or the ternary distributions. We must thus consider variances that are greater than one, and there it can be seen in Figure 3 that the entropy of the centered binomial distribution for such variances is *lower* than that of the discrete Gaussian distribution.

4.2 Comparing failure probability against variance

We discuss the impact of the secret-key distribution choice on the decryption failure rate of the generalized lattice-based public-key encryption scheme described in Section 2, algorithms 1, 2 and 3. For concreteness and simplicity, we consider a Ring Learning with Rounding based instantiation of it, i.e., we choose $l = 1$, $\boldsymbol{E}_A = \boldsymbol{E}'_B = \boldsymbol{E}''_B = \mathbf{0}$. To prevent the polynomial degree from being restricted to only powers of 2, we choose the key-generation reduction polynomial $f_1(x) = \Phi_{n+1}(x) = x^n + x^{n-1} + \ldots + 1$, the $(n+1)$-th cyclotomic polynomial for $n+1$ a prime. To avoid correlated errors due to the use of this specific $f_1(x)$ and also to reduce decryption failure rates to the level achieved by sparser cyclotomic polynomials such as in [6], we choose the encryption reduction polynomial $f_2(x) = x^{n+1} - 1$. We refer to [9] for details on this technique, noting that it requires the polynomials in $\boldsymbol{S}_A$ and $\boldsymbol{S}'_B$ to have a factor $(x-1)$.

Section 4.1 mentioned that the (fixed-weight and symmetric) ternary distributions can only have variance at most one, while the centered binomial dis-

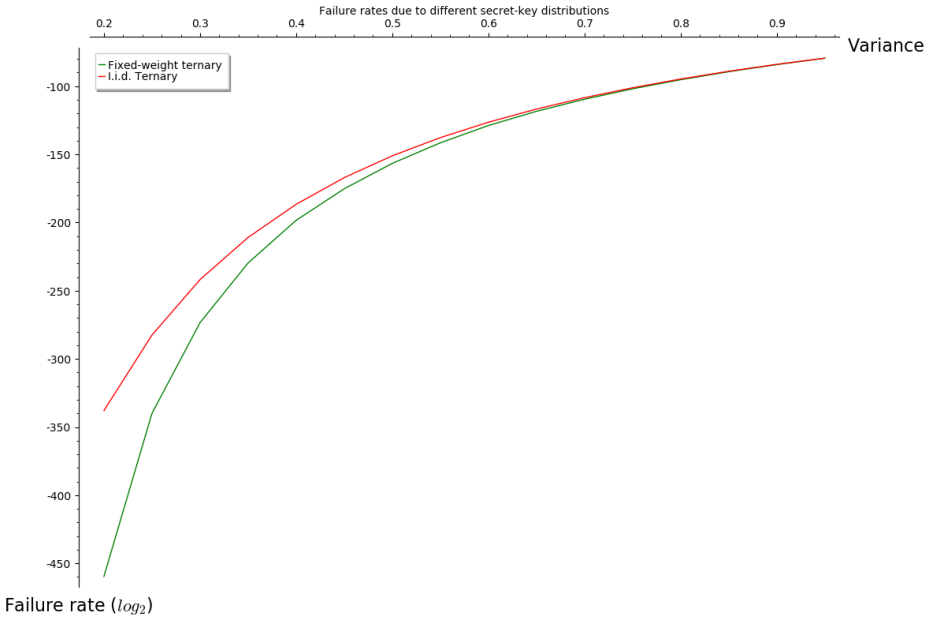

Fig. 4: Comparison of the failure rates of the lattice-based encryption scheme described in Section 4.2, when using fixed-weight ternary (green) and symmetric, non fixed-weight ternary (red) secret keys.

tribution can only have variance at least one. This complicates a direct (i.e., with variances equalized) comparison of the failure rates resulting from these distributions. Instead, in this section we analyze the effect of fixing the composition of the secrets on the decryption failure rate, i.e., the effect of fixing the *exact number* or *weight* of secret key components for each possible component value. In other words, we will compare the failure rate of the above-mentioned public-key encryption scheme when instantiated with the symmetric ternary distribution as opposed to the fixed-weight ternary distribution. Next, we will do the same and compare the centered binomial distribution $\mathrm{bin}\_\eta(k)$, with a *fixed-composition variant* of it, i.e., with the uniform distribution on the set of all vectors in $\{-\eta, \ldots, \eta\}^n$ with *exactly* $\lfloor \binom{2\eta}{k+\eta} 2^{-2\eta} n \rfloor$ components equal to k (for $k \in \{-\eta, \ldots, \eta\} \setminus \{0\}$), and the remaining components are zero.

For fixed scheme parameters $n = 800$, $q = 2^{11}$, $p = 2^9$, $t = 2^7$, Figure 4 compares the failure rates achieved by the above scheme for the symmetric ternary and fixed-weight ternary distributions. In the former case, the failure probability can be computed by iteratively convolving the symmetric ternary secret distribution and that of the "rounding" error, similar to [25]. In case of fixed-weight ternary secrets, assuming independence, the failure probability can be computed similarly as in [9, Sec. 4.3] where one term in the decryption error polynomial is distributed as the sum of *exactly* h independent uniform random variables on

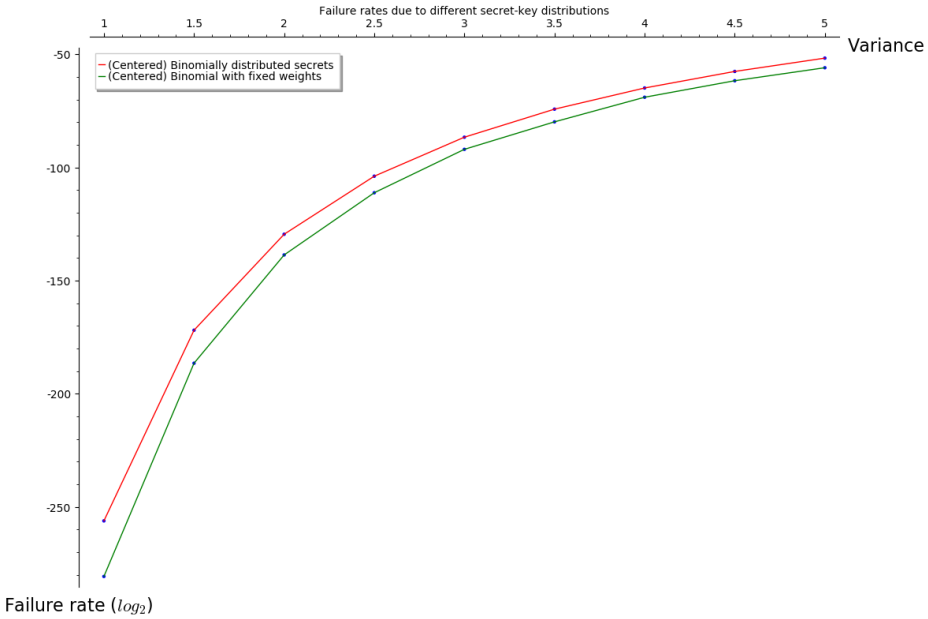

Fig. 5: Comparison of the failure rates of the lattice-based encryption scheme described in Section 4.2, when using secrets drawn from a (centered) binomial distribution (red) and one where for each given value the numbers of secret coefficients are fixed (green), i.e., a fixed-composition variant.

$(-q/2p, q/2p] \cap \mathbb{Z}$, minus the sum of h independent uniform random variables on $(-q/2p, q/2p] \cap \mathbb{Z}$.

Figure 4 shows that the use of fixed-weight ternary secrets in the encryption scheme reduces its decryption failure rates. The explanation is that, unlike their fixed-weight counterparts, secret keys sampled from the symmetric ternary distribution may have a Hamming weight that is *higher* than expected. The same reasoning holds for secret keys sampled from distributions that for which the number of each possible symbol is fixed.

Conversely, there also exists a non-negligible probability that secrets sampled from the symmetric ternary distribution have a weight that is *lower* than expected, which leads to the possibility of a multi-target attack along the lines of [35] that reduces the cost of key recovery attacks such as [39,2]. In this attack, the attacker can perform a one-time precomputation in order to find an encryption randomness that results in higher than expected probability of decryption failure, use this to compute ciphertexts for multiple targets, out of which targets with secret keys that have *smaller than expected weight* can be identified if decryption unexpectedly succeeds. Fixing the weight of the secret keys, in addition to improving the decryption failure rate, also stops this attack.

The same result as for the ternary distributions can be seen again in Figure 5 that compares the decryption failure rate of the encryption scheme for the centered binomial distribution and its fixed-composition variant introduced above.

The comparison is done for the fixed parameters $n = 828$, $q = 2^{14}$, $p = 2^{10}$, $t = 2^4$. It is again seen that fixing the weight of the non-zero components reduces the failure rate, since it strictly limits the number of non-zero components that are sampled in the secret key, as well as removing the possibility of sampling many large components.

4.3 Comparing security and performance trade-offs

The variance, entropy, failure probability and computational performance can be computed given a type of secret-key distribution. However, it is not straighforward to derive a conclusion about the overall system looking at them individually, since their effect is interlinked, as shown in Figure 1. While a secret distribution aspect (e.g., low variance) can have a positive impact on the encryption scheme, e.g., low decryption failure, it can also have a negative impact, e.g., lower concrete security. Therefore, we choose to analyze the eventual effect of this interaction on the final scheme, by performing a parameter search with the aim of minimizing the bandwidth requirements. Our chosen scheme is the same Ring Learning with Rounding based instantiation of the generalized PKE scheme of Section 2, that we instantiated in Section 4.2, i.e., by choosing $l = 1$, $\boldsymbol{E}_A = \boldsymbol{E}'_B = \boldsymbol{E}''_B = \mathbf{0}$ $f_1(x) = \Phi_{n+1}(x)$, $f_2(x) = x^{n+1} - 1$. Parameters are chosen to encrypt a 192-bit message, while offering a minimum targeted security level (NIST security category 3 [56]) and ensuring a negligibly low decryption failure rate so that standard transformations [38] can be applied on the scheme to obtain an IND-CCA2 secure scheme.

While choosing parameters, concrete security is analyzed considering the best known current attacks, which are ones that utilize lattice basis reduction, under the conservative core-sieving model [6] assuming sieving [46] as the underlying SVP oracle in basis reduction. Although the exact cost of lattice reduction is considered unclear in the literature [21,36,54,3], it is dominated by that of running the SVP oracle on b-dimensional lattices. Ignoring asymptotic factors, this cost is estimated as $2^{0.292b}$ and $2^{0.265b}$ [45,47,46] respectively, depending on whether a quantum speedup by Grover's algorithm [31] is assumed or not. Attacks considered include the *primal* or decoding attack [6], and the *dual* or distinguishing attack [2], extended to utilize *lattice rescaling* [9,10,2], to exploit the fact that a number of secret-key distributions in this work are relatively *narrower* than the error, and result in unbalanced short lattice vectors. To further account for such *narrow* secret-key distributions, specialized or combinatorial attacks such as the hybrid lattice reduction and meet-in-the-middle attack [39], and a sparse-secret attack [2] are also considered.

Table 1 summarizes the computed parameters and compares the achieved performance. The first row shows parameters related to the secret-key distribution, namely η, θ, and h where applicable, as defined in Section 3, and variance σ^2 of the distribution. For schemes with fixed-composition secret-key distributions, the parameter $h = [h_1, h_2, \ldots, h_\eta]$ describes the composition: for $1 \leq i \leq \eta$, the secret key has h_i components equal to i and h_i components equal to $-i$. The second row includes the size n of the reduction polynomial and the moduli – q,

p, and t – involved in the Ring Learning with Rounding (RLWR) problem [11]. Note that each secret has $n - 2\sum_{i=1}^{\eta} h_i$ components equal to zero. We observe that all configurations achieve a classical level of at least 192 bits of security, and a failure rate of at most 2^{-170}.

While not a formal proof, the conclusion from the comparison in Table 1 is: for the same security and decryption failure rate targets, a rounding-based public-key encryption scheme with smallest bandwidth requirements is obtained when the secret keys are sampled from a fixed-weight ternary distribution. This result can be explained considering the behaviour of entropy/variance in Figure 3 and decryption failure probability in Figure 4 and 5. First, a Gaussian distribution achieves the highest entropy for any given variance (Figure 3). In a low variance regime (e.g., for variances less than 0.4), the ternary secret-key distributions have entropy almost as high as that of the Gaussian, with the advantage that ternary secrets do not admit larger components as might happen with a Gaussian distribution, and thus lead to a lower failure probability. Further, secret-keys sampled from fixed-composition distributions cannot have more components than expected, also lowering the failure probability (Figures 4 and 5).

Thus, it is logical to expect that when a cryptographic scheme's parameters are optimized in the low variance regime, fixed-weight ternary secrets will lead to the lowest failure rates and smallest key sizes overall. In the high variance regime, Figure 3 shows that binomial distributions have lower entropy than that of Gaussian for a given variance, although this may still be high enough to deal with specialized or combinatorial attacks such as [39,2]. Important to note however, is that the failure probability of any distribution in this high variance regime is worse than that seen in the low variance regime. This is why even while possibly allowing specialized attacks [39,2], fixed-weight ternary keys also enable stronger noise tolerance and a higher noise-to-modulus ratio, improving security. The combination of these two competing effects while optimizing parameters of the encryption scheme, allow fixed-weight ternary secrets to provide the smallest key sizes. It is an open question whether this result can be formalized.

5 Resistance against decryption failure-based chosen-ciphertext attacks

An important aspect to consider while designing public-key encryption and encapsulation schemes is their security against active or chosen-ciphertext attacks, formalized in the notion of IND-CCA2 security. Lattice-based encryption and encapsulation schemes typically have a probability of decryption failure, which depends on the instantiation of the secret key and noise of both parties. In case of schemes based on (Ring) Learning with Rounding [11], it depends only on the secret keys since the noise is deterministic and based on the secret keys and the public parameter. One possible attack against such schemes in the IND-CCA2 model involves an attacker who chooses ciphertexts with the goal of *causing* a

decryption failure, and hopes to gain information on the decryptor's secret key by observing such failure events.

D'Anvers *et al.* [23] present an attack framework of this form on a number of NIST post-quantum standardization candidates, and compare how the different schemes fare against it. However, the roles played by different aspects of the schemes' designs in withstanding active attacks of the above form are not completely studied. In this section we analyze the role played by the secret-key distribution of a scheme in this attack. We analyze and quantify the cost of the above-mentioned decryption failure-based attack [23] against each of the secret-key distributions considered in this paper, in the context of the rounding-based encryption scheme from Section 4.2. We show that fixing the weight of the (non-zero) secret key components makes the attack harder, independent of any other scheme parameters such as the error distribution.

We first recall the basic intuition behind the attack: Typically, most lattice-based public-key encryption schemes consist of a core building block that is an IND-CPA secure public-key encryption scheme. Applying a KEM variant [38] of the Fujisaki-Okamoto transform on this scheme yields an IND-CCA2 secure scheme, whose security can be proven in the random oracle model. A core component of the above transform is a *re-encryption step* that intuitively requires the encryptor to prove knowledge of the message that is being encrypted/encapsulated. Thus, an active or chosen ciphertext attacker can do no better than exhaustively search for messages that result in so-called "weak" ciphertexts [23] which cause a decryption failure with probability higher than a threshold f_t – this is a parameter chosen by the attacker. The probability of finding such weak ciphertexts is denoted in [23] by the parameter α, and the (increased) decryption failure rate resulting from them is denoted by the parameter β. Assuming that the attacker has no quantum access to the decryption oracle, the overall attack cost of this so-called *failure boosting* attack, is thus $(\alpha\beta)^{-1}$ (using a classical computer) or $(\sqrt{\alpha}\beta)^{-1}$ (with a quantum speedup [31]). Once weak ciphertexts are found, the attacker uses information gained by observing decryption failure events to speed up standard secret key recovery attacks [6,39].

We recall some notation from [23] before proceeding to the analysis of the attack and our extension of it. In the context of the generalized public-key encryption scheme described in Section 2, algorithms 1, 2 and 3, we define the errors introduced by the rounding operation (if applicable) as:

$$\begin{gathered}\boldsymbol{U}_A = \boldsymbol{A}\boldsymbol{S}_A + \boldsymbol{E}_A - \boldsymbol{B}_r,\ \boldsymbol{U}'_B = \boldsymbol{A}^T\boldsymbol{S}'_B + \boldsymbol{E}'_B - \boldsymbol{B}'_r, \\ \boldsymbol{U}''_B = \boldsymbol{B}_r^T\boldsymbol{S}'_B + \boldsymbol{E}''_B + \left\lfloor \frac{q}{2}m \right\rceil - \boldsymbol{V}'_r.\end{gathered} \tag{6}$$

Further, let

$$\boldsymbol{S} = \begin{pmatrix} -\boldsymbol{S}_A \\ \boldsymbol{E}_A + \boldsymbol{U}_A \end{pmatrix},\ \boldsymbol{C} = \begin{pmatrix} \boldsymbol{E}_B + \boldsymbol{U}'_B \\ \boldsymbol{S}'_B \end{pmatrix},\ \boldsymbol{G} = \boldsymbol{E}''_B + \boldsymbol{U}''_B \tag{7}$$

The above-mentioned attack cost of failure boosting can be minimized by the attacker over the choice of the failure probability threshold f_t. This minimization

requires obtaining the (distribution of the) variance of the coefficients of the polynomial $\boldsymbol{S^T C}_{ij}$ [23, Eq. 8] (where i, j are used to vary over the coefficients of the polynomials in $\boldsymbol{S}^T\boldsymbol{C}$):

$$\mathrm{var}(\boldsymbol{S^T C}_{ijk}) = \|(\boldsymbol{E}'_B + \boldsymbol{U}'_{B:j})\|_2^2 \mathrm{var}(\chi_s) + \|(\boldsymbol{S}'_{B:j})\|_2^2 \mathrm{var}(\chi_{e+u}) \tag{8}$$

Computing (the distribution of) this variance is an essential step towards *modeling the failure probability* in each component of the polynomials involved in the decryption error term that is represented by $\boldsymbol{S}^T\boldsymbol{C} + \boldsymbol{G}$. The attacker chooses $(\boldsymbol{C}, \boldsymbol{G})$ with the eventual goal of finding a "weak" ciphertext that causes a higher than expected probability of decryption failure.

Computing the distribution of the failure probability in each component of the decryption error eventually allows determining whether the attacker succeeds in causing the probability of the decryption failure rate in each component of the above polynomials to be greater than the chosen f_t, for a chosen $(\boldsymbol{C}, \boldsymbol{G})$ pair – this qualifies as an *attack success*. This involves computing the resulting α and β parameters, and thus the overall attack cost. A key assumption made by [23] in the above computations is that the coefficients of $\boldsymbol{S}^T\boldsymbol{C}$ are normally distributed. This assumption also allows applying this analysis to a number of different schemes with varying secret-key and noise distributions. However, as mentioned by the authors themselves in [23, Section 3.1], this assumption is also the source of potential inaccuracies in the analysis. In the following section we show that this inaccuracy exists for specific distributions, and can be removed by refining the analysis of [23] for said distributions.

5.1 Adapting failure boosting to fixed-composition secrets

In this section, we refine the of failure boosting analysis from [23] for schemes which have secret keys with fixed composition, for which there are integers $h_{-\eta}, \ldots, h_\eta$ such that for each $i \in \{-\eta, -\eta+1, \ldots, 0, 1, \ldots, \eta\}$, each secret has h_i components equal to i. Specifically, the Gaussian approximation step involved in computing the distribution of the decryption failure probabilities per component of $\boldsymbol{S}^T\boldsymbol{C} + \boldsymbol{G}$ can be made more precise for such schemes. This allows for a more accurate computation of the distribution of the coefficients in the polynomials comprising $\boldsymbol{S}^T\boldsymbol{C}$, improving the analysis of [23].

The decryption failure probability f_{ijk} in the ijk-th component of $\boldsymbol{S}^T\boldsymbol{C}+\boldsymbol{G}$, given a chosen pair $(\boldsymbol{C}, \boldsymbol{G})$ is computed in [23, Eq. 11] as:

$$\begin{aligned} f_{ijk} &= \Pr\left(|(\boldsymbol{S}^T\boldsymbol{C} + \boldsymbol{G})_{ijk}| > q_t | \boldsymbol{G}, \boldsymbol{C}\right) \\ &\approx \Pr\left(|x + \boldsymbol{G}_{ijk}| > q_t | \boldsymbol{G}, x \leftarrow \mathcal{N}\left(0, \mathrm{var}(\boldsymbol{S^T C}_{ijk})\right)\right). \end{aligned} \tag{9}$$

$q_t = q/2^B$ is a decryption threshold, where q is the system modulus and B bits of information are extracted from each component of the shared secret. The previously mentioned assumption made by [23] that the coefficients of $\boldsymbol{S}^T\boldsymbol{C}$ are normally distributed results in the approximation in the second step of the

above equation. We refine this approximation by first noting that (recalling the definitions of $\boldsymbol{S}$, $\boldsymbol{C}$ and $\boldsymbol{G}$ in Eq. 7):

$$\begin{aligned}\boldsymbol{S}^T\boldsymbol{C} &= \left(-\boldsymbol{S}_A^T\ (\boldsymbol{E}_A^T + \boldsymbol{U}_A^T)\right)\begin{pmatrix}\boldsymbol{E}_B + \boldsymbol{U'}_B \\ \boldsymbol{S'}_B\end{pmatrix} \\ &= -\boldsymbol{S}_A^T(\boldsymbol{E}_B + \boldsymbol{U'}_B) + (\boldsymbol{E}_A^T + \boldsymbol{U}_A^T)\boldsymbol{S'}_B \\ &= \mathbf{S}_A\mathbf{E}_B + \mathbf{E}_A\mathbf{S}_B\end{aligned} \tag{10}$$

where

$$\mathbf{S_A} = -\boldsymbol{S}_A^T,\ \mathbf{E}_B = \boldsymbol{E}_B + \boldsymbol{U'}_B,\ \mathbf{E}_A = \boldsymbol{E}_A^T + \boldsymbol{U}_A^T,\ \mathbf{S_B} = \boldsymbol{S'}_B. \tag{11}$$

Rewriting $\boldsymbol{S}^T\boldsymbol{C}$ in the above manner makes it clear that the computation of f_{ijk} in Eq. 9 can be refined as follows:

$$f_{ijk} \approx \Pr\left(|x_1 + x_2 + \boldsymbol{G}_{ijk}| > q_t | \boldsymbol{G}, x_1 \leftarrow \mathcal{N}\left(0, \mathrm{var}((\mathbf{S}_A\mathbf{E}_B)_{ijk})\right), x_2 \leftarrow \chi_{\mathbf{E}_A\mathbf{S}_B}\right) \tag{12}$$

where firstly, $\mathrm{var}((\mathbf{S}_A\mathbf{E}_B)_{ijk})$ is adapted from $\mathrm{var}(\boldsymbol{S^T C}_{ijk})$ in Eq. 8 as:

$$\mathrm{var}((\mathbf{S}_A\mathbf{E}_B)_{ijk}) = \|(\boldsymbol{E'}_B + \boldsymbol{U'}_{B:j})\|_2^2 \mathrm{var}(\chi_s) \tag{13}$$

and secondly, $\chi_{\mathbf{E}_A\mathbf{S}_B}$ is the distribution of each component in the polynomial product $(\mathbf{E}_A\mathbf{S}_B)_{ij}$, where polynomials in $\mathbf{S}_B$ have components from $\{-\eta, -\eta+1, \ldots, \eta\}$, with exactly h_i components equal to i. The variable x_2 is thus distributed as $\sum_{i=-\eta}^{\eta} i \sum_{j=1}^{h_i} X_{i,j}$, where the random variables $X_{i,j}$ for $1 \leq i \leq \eta$ and $1 \leq j \leq h_i$ are independently drawn from χ_{e+u}. This distribution can be computed by convolving, over all $i \in [-\eta, \ldots, \eta]$, the h_i-fold iterative convolution of χ_{e+u} scaled with a factor i. Equations 12 and 13 thus summarize our adaptation to the analysis of the failure boosting phase of the decryption failure-based active attack in [23].

5.2 Results of our adaptation

Applying our above adapted analysis, Figure 2 depicts the cost $(\alpha\beta)^{-1}$ of applying failure boosting to find a weak ciphertext on a classical computer, against the decryption failure rate of the ciphertext. This is depicted for a number of NIST post-quantum standardization candidates, and the rounding-based public-key encryption scheme we defined in Section 4.2 – instantiated with both fixed-weight and non fixed-weight ternary secrets. The parameters used for these are the same as computed in Section 4.3. Note that the construction of the encryption scheme using these distributions is similar to some instantiations of the NIST PQC candidate Round5 [9], albeit a version that does not use error correction and also considers non fixed-weight ternary secrets. The results clearly indicate that fixing the weight makes the attack harder. As a benchmark, Figure 2 also shows the upper bound placed by the NIST post-quantum standardization call [56], on the number of decryption oracle queries (2^{64}) that an active attacker can make.

To further demonstrate this, and to show that the results of our adaptation carry over to schemes using secret-keys with a larger support and also independent errors (instead of only rounding-based errors), we include results for

fixed-composition variants of the NIST PQC candidates Saber [24] (rounding errors), Kyber [18] (secrets and independent errors sampled from a centered Binomial distribution), and Frodo [17] (secrets and independent errors sampled from a discrete Gaussian distribution). The fixed-composition variants of Saber and Kyber that we consider sample their secret keys from centered binomial distributions [6] analogous to the original schemes, however the number of occurrences of the non-zero components are fixed in a manner similar to that mentioned in Section 4.2. Similarly, the number of each of the secret-key components for the fixed-composition variant of Frodo is fixed to the expectation of the discrete Gaussian distribution from which the original scheme samples its secret-key components. Figure 2 shows that the cost of the failure boosting attack increases visibly faster for these *fixed-composition* variants of Saber, Kyber and Frodo than the original schemes. Since these schemes were originally designed to use secret key components that are larger than 1, fixing the number of non-zero secret key components provides them with an even greater security benefit against the failure boosting attack of [23] than schemes using secret keys with components that are only ternary.

6 Conclusions and future work

Of all the different design aspects involved in the construction of lattice-based public-key encryption schemes, an important one that has so far not been analyzed in depth in the literature is the role played by the secret-key distribution in the tradeoff between performance and security of the scheme. We initiate study in this area by comparing a number of secret-key distributions currently being considered as part of candidates to the NIST post-quantum standardization process, with respect to different criteria such as variance, entropy, resulting probability of decryption failure, and resistance against chosen-ciphertext attacks based on decryption failures.

Our results indicate that out of the secret-key distributions considered in this work: *firstly*, fixed-weight ternary secrets reduce the decryption failure rate of the encryption scheme and allow for a higher noise-to-modulus ratio while ensuring a large enough dimension secure against concrete attacks, thus leading to the smallest key sizes when parameters are optimized for bandwidth. *Secondly*, fixing the number of components in the secret key for each given value increases security against decryption failure-based chosen-ciphertext attacks, as compared to secrets with independently sampled components. An interesting area of further research is to analyze the effect of having fixed-composition errors on the security and performance of the encryption scheme.

Acknowledgements

We thank Thijs Laarhoven for helpful discussions on comparing the entropy and variance of the fixed-weight and symmetric ternary distributions, and for pointing out the possibility of multi-target attacks on schemes using the latter.

We thank Jan-Pieter D'Anvers for providing helpful insights on the analysis of the "failure boosting" technique in [23]. The research of Player was supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701).

References

1. Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018.
2. Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *EUROCRYPT*, pages 103–129, 2017.
3. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. Cryptology ePrint Archive, Report 2019/089, 2019. To appear in Eurocrypt 2019.
4. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
5. Erdem Alkim, Roberto Avanzi, Joppe W. Bos, Léo Ducas, Antonio de la Piedra, Thomas Pöppelmann, Peter Schwabe, and Douglas Stebila. NewHope. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
6. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - a new hope. In *USENIX Security Symposium*, pages 327–343, 2016.
7. Jacob Alperin-Sheriff. Public discussion, https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/LAC-official-comment.pdf, Page 19, February 2018. Messages on the NIST PQC mailing list; OFFICIAL COMMENT: LAC.
8. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, pages 595–618, 2009.
9. Hayo Baan, Sauvik Bhattacharya, Scott Fluhrer, Oscar Garcia-Morchon, Thijs Laarhoven, Ludo Tolhuizen, Ronald Rietman, Markku-Juhani O. Saarinen, and Zhenfei Zhang. Round5: Compact and Fast Post-quantum Public-Key Encryption. Cryptology ePrint Archive, Report 2019/090. To appear in PQCrypto 2019.
10. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In *ACISP*, pages 322–337, 2014.
11. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, pages 719–737, 2011.
12. Daniel J. Bernstein. Public discussion, https://groups.google.com/a/list.nist.gov/forum/#!msg/pqc-forum/VK9dROwgY0Y/5PRkb6TYCQAJ, January 2019. Messages on the NIST PQC mailing list: Really fast NTRU.
13. Daniel J. Bernstein. Public discussion, https://groups.google.com/a/list.nist.gov/forum/#!msg/pqc-forum/l5IaJTe_pUI/QKaLZ4uMAAAJ, January 2019. Messages on the NIST PQC mailing list: OFFICIAL COMMENT: NTRU Prime.

14. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU Prime. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
15. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU Prime: reducing attack surface at low cost. In *SAC*, pages 235–260, 2017.
16. Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography - dealing with the fallout of physics success. *IACR Cryptology ePrint Archive*, 2017:314, 2017.
17. Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In *CCS*, pages 1006–1018, 2016.
18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. In *Euro S&P*, pages 353–367, 2018.
19. Matt Braithwaite. Experimenting with Post-Quantum Cryptography (CECPQ1), 2016. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html.
20. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.
21. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.
22. Jung Hee Cheon, Sangjoon Park, Joohee Lee, Duhyeong Kim, Yongsoo Song, Seungwan Hong, Dongwoo Kim, Jinsu Kim, Seong-Min Hong, Aaram Yun, Jeongsu Kim, Haeryong Park, Eunyoung Choi, Kimoon Kim, Jun-Sub Kim, and Jieun Lee. Lizard Public Key Encryption. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
23. Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. On the impact of decryption failures on the security of LWE/LWR based schemes. In *PKC*, pages 565–589, 2019.
24. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
25. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In *AFRICACRYPT*, pages 282–305, 2018.
26. Mikhail Dyakonov. The Case Against Quantum Computing, 2018. Available at https://spectrum.ieee.org/computing/hardware/the-case-against-quantum-computing.
27. ETSI. "ETSI launches Quantum Safe Cryptography specification group", March 2015. Available at http://www.etsi.org/news-events/news/947-2015-03-news-etsi-launches-quantum-safe-cryptography-specification-group.
28. Scott Fluhrer. Cryptanalysis of ring-LWE based key exchange with key share reuse. Cryptology ePrint Archive, Report 2016/085, 2016.
29. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. https://crypto.stanford.edu/craig/craig-thesis.pdf.
30. Google. Google AI Quantum. https://ai.google/research/teams/applied-science/quantum-ai/.

31. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219, 1996.
32. Emily Grumbling and Mark Horowitz, editors. *Quantum Computing: Progress and Prospects*. The National Academies Press, Washington, DC, 2018. Consensus Study Report.
33. Qian Guo, Thomas Johansson, and Alexander Nilsson. A Generic Attack on Lattice-based Schemes using Decryption Errors with Application to ss-ntru-pke. Cryptology ePrint Archive, Report 2019/043, 2019.
34. Shai Halevi and Victor Shoup. Bootstrapping for HElib. In *EUROCRYPT*, pages 641–670, 2015.
35. Mike Hamburg. Public discussion, https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/official-comments/LAC-official-comment.pdf, Page 24, April 2018. Messages on the NIST PQC mailing list; OFFICIAL COMMENT: LAC.
36. Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing blockwise lattice algorithms using dynamical systems. In *CRYPTO*, pages 447–464, 2011.
37. Philip S. Hirschhorn, Jeffrey Hoffstein, Nick Howgrave-Graham, and William Whyte. Choosing NTRUEncrypt Parameters in Light of Combined Lattice Reduction and MITM Approaches. In *ACNS*, pages 437–455, 2009.
38. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC*, pages 341–371, 2017.
39. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *CRYPTO*, pages 150–169, 2007.
40. Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In *CHES*, pages 232–252, 2017.
41. IBM. IBM Q: The future is quantum. https://www.research.ibm.com/ibm-q/.
42. Intel. Intel Advances Quantum and Neuromorphic Computing Research, 2018. https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/#gs.C4B2JhaK.
43. IonQ. IonQ harnesses single-atom qubits to build the world's most powerful quantum computer. https://ionq.co/news/december-11-2018.
44. Adrienne W. Kemp. Characterizations of a discrete normal distribution. *Journal of Statistical Planning and Inference*, 63:223–229, 1997.
45. Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. Cryptology ePrint Archive, Report 2014/744, 2014.
46. Thijs Laarhoven. *Search problems in cryptography. From fingerprinting to lattice sieving*. PhD thesis, Eindhoven University of Technology, 2016. http://thijs.com/docs/phd-final.pdf.
47. Thijs Laarhoven, Michele Mosca, and Joop van de Pol. Finding shortest lattice vectors faster using quantum search. *Designs, Codes and Cryptography*, 77(2–3):375–400, 2015.
48. Adam Langley. CECPQ2, 2018. https://www.imperialviolet.org/2018/12/12/cecpq2.html.
49. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 77(3):565–599, 2015.
50. Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, and Zhenfei Zhang. LAC. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

51. Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, and Zhenfei Zhang. Round 2 Submissions – LAC. Technical report, National Institute of Standards and Technology, 2019. Available at https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions.
52. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.
53. Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO*, pages 21–39, 2013.
54. Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *EUROCRYPT*, pages 820–849, 2016.
55. Microsoft. Microsoft, Quantum. https://www.microsoft.com/en-us/quantum/.
56. NIST. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. POST-QUANTUM CRYPTO STANDARDIZATION. Call For Proposals Announcement, 2016.
57. Chris Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015.
58. Thomas Pöppelmann, Erdem Alkim, Roberto Avanzi, Joppe W. Bos, Leo Ducas, Antonio de la Piedra, Peter Schwabe, and Douglas Stebila. NewHope. Technical report, National Institute of Standards and Technology, 2017. Available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
59. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
60. Oded Regev. The learning with errors problem (invited survey). In *CCC*, pages 191–204, 2010.
61. Max F Riedel, Daniele Binosi, Rob Thew, and Tommaso Calarco. The European quantum technologies flagship programme. *Quantum Science and Technology*, 2:030501, June 2017.
62. David Schneider. The U.S. National Academies Reports on the Prospects for Quantum Computing, 2018.
63. Microsoft SEAL (release 3.2). https://github.com/Microsoft/SEAL, February 2019. Microsoft Research, Redmond, WA.
64. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, October 1997.