Concrete Security from Worst-Case to Average-Case Lattice Reductions

Joel Gärtner

June 16, 2023

Abstract

A famous reduction by Regev shows that random instances of the Learning With Errors (LWE) problem are asymptotically at least as hard as a worst-case lattice problem. As such, by assuming that standard lattice problems are hard to solve, the asymptotic security of cryptosystems based on the LWE problem is guaranteed. However, it has not been clear to which extent, if any, this reduction provides support for the security of present concrete parametrizations.

In this work we therefore use Regev's reduction to parametrize a cryptosystem, providing a reference as to what parameters are required to actually claim security from this reduction. This requires us to account for the concrete performance of this reduction, allowing the first parametrization of a cryptosystem that is provably secure based only on a conservative hardness estimate for a standard lattice problem. Even though we attempt to optimize the reduction, our system still requires significantly larger parameters than typical LWE-based cryptosystems, highlighting the significant gap between parameters that are used in practice and those for which worst-case reductions actually are applicable.

1 Introduction

With the conclusion of the third round of the NIST post-quantum standardization process, lattice-based cryptography is one step closer to see widespread adoption. As a result of this third round, NIST have selected the lattice-based Key Encapsulation Mechanism (KEM) Kyber [35] for standardization.

Kyber is a cryptosystem with security based on the assumed hardness of a structured version of the Learning With Errors (LWE) problem. LWE is a relatively new problem that was first introduced by Regev in 2005 [30]. In the same paper, Regev showed that it is asymptotically at least as hard to solve

random instances of the LWE problem as it is to solve a worst-case instance of a standard lattice problem on a quantum computer. As lattice problems are believed to be hard to solve, Regev could reasonably claim that LWE is a hard problem, even though no one had analyzed this problem before.

The hardness of structured versions of LWE, such as the module-LWE problem that Kyber is based on, are also supported by this type of reduction from worst-case lattice problems [21, 24]. These reductions provide an argument for using these problems for cryptography but are typically not used to argue for the concrete security of cryptosystems. Instead cryptosystems typically base their security directly on an estimate for the concrete hardness of the relevant version of the LWE problem.

When estimating the hardness of LWE, one typically considers the primal and dual lattice attacks. These attacks are based upon transforming an LWE instance into a lattice problem, which is solved by using standard lattice algorithms. As such, the concrete hardness of the LWE problem is extrapolated based on the performance of lattice algorithms [3].

It may further be argued that the worst-case to average-case reductions serve as a qualitative argument for the security of LWE-based cryptosystems with Micciancio and Regev [25] meaning that it "assures us that there are no fundamental flaws in the design of our cryptographic construction". This may serve as a reason to prefer LWE-based schemes over other lattice-based schemes, such as NTRU. Furthermore, as the hardness of the LWE problem is guaranteed if lattice problems are hard, it may have resulted in less focus on non-lattice based algorithms to solve LWE, since we in some sense are guaranteed that these can not perform better than lattice algorithms.

However, although these types of arguments may be reasonable in an asymptotic sense, it has not previously been clear if they, at least to some extent, are applicable to parameters used in practice. While Micciancio and Regev also mention that using the reductions to set parameters seems to be overly conservative, there has not really been any investigation to how large such parameters actually would have to be.

If parameters supported by the reductions have similar size to the ones used in practice, the reductions could serve as a lower bound on how much the security could drop in a cryptosystem. This reasoning has for example been used by Peikert when arguing for the importance of worst-case reductions in lattice-based security [29]. However, if the reductions only support schemes with significantly larger parameters than typical cryptosystems, even at relatively low security levels, then the support these worst-case reductions provide to typical schemes is questionable.

In order to investigate to which extent such reductions provide any security

lower bounds or other qualitative support for typical LWE-based cryptosystems, we investigate a cryptosystem with security that is actually based on such a worst-case to average-case reduction. This cryptosystem is parametrized to take into account the concrete performance of used reductions and is based on a reasonable estimate on the concrete hardness of the underlying problem. This guarantees that the system remains secure as long as there are no significant improvements in the efficiency of lattice algorithms.

The concrete performance of Regev's reduction have been analyzed in some previous works [9, 14, 33]. Furthermore, a cryptosystem parametrized through the reduction was also proposed in [14], with parameter sizes comparable to a typical LWE-based cryptosystem. However, this parametrization mainly focused on the efficiency of the reduction and was not based on a realistic estimate for the hardness of the underlying problem. Therefore, there has not previously been any good reference for which parameters are required by a cryptosystem to have its concrete security supported by Regev's reduction.

Worth noting is that the specification of FrodoKEM [26], another LWE-based cryptosystem very similar to ours, also includes a reduction from a worst-case lattice problem. However, for FrodoKEM, this reduction is mainly as a qualitative argument for the security of the system and there is no analysis of which concrete parameters are supported by the reduction. The reduction essentially corresponds to a single classical step of Regev's full quantum reduction, resulting in the reduction solving a less standard lattice problem but with less requirements on the cryptosystem. We do not expect this approach to support the security of cryptosystems that use significantly smaller parameters than the ones we propose in this paper and we therefore consider Regev's full reduction from a more standard lattice problem.

1.1 Our Contributions

In this work we construct a cryptosystem that is parametrized based on a version of Regev's original quantum reduction from worst-case lattice problems to average case LWE [31]. As such, the security of this cryptosystem is actually guaranteed by the concrete hardness of a well studied standard lattice problem. This provides a reference to what parameters are required for similar reductions to say something meaningful about the concrete security of a cryptosystem.

Using this reduction to parametrize our cryptosystem requires that we keep track of its concrete performance, both in terms of running time and approximation factor for which it solves the underlying lattice problem. To allow our cryptosystem to use smaller parameters, we modify Regev's original reduction in order to improve its concrete efficiency. Even with these modifications, both the running time and the approximation factor of the reduction are relatively large and our cryptosystem requires significantly larger parameters than typical

LWE-based cryptosystems.

Currently proposed LWE based cryptosystems are typically parametrized with a dimension n that is approximately 1000 when targeting 256 bits of security. Meanwhile, our cryptosystem that targets 128 bits of OW-CPA post-quantum security requires using a dimension $n \approx 35800$. We also consider an alternative version of the LWE problem which allows a more efficient version of Regev's reduction. Using this version of the LWE problem for our cryptosystem therefore allows a more efficient parametrization, with this version having the same security guarantees while using $n \approx 29900$.

Even if we completely ignore the running time of the reduction, our cryptosystem requires a dimension $n \approx 9400$ when targeting 128-bits of OW-CPA security. It thus seems like significantly larger parameters than those used by typical LWE-based cryptosystems are required in order for these types of worst-case to average-case reductions to say anything meaningful about the systems security. While this does not indicate that currently proposed LWE-based cryptosystems are insecure, it does mean that the security of these schemes is far from supported by arguments that rely on these types of reductions.

As such, we do not consider it reasonable to use these reductions as an argument for a lower bound on the security of LWE-based cryptosystems used in practice. Furthermore, we do not deem similar worst-case to average-case reductions to be a strong argument in favour of typical LWE-based cryptosystems over other lattice-based systems, such as NTRU.

1.2 Overview

1.2.1 Cryptosystem

The cryptosystem we construct in this paper is based on the Lindner-Peikert scheme [23], in a similar way to FrodoKEM [26]. To parametrize our cryptosystem, we use a reduction from an approximate version of the Shortest Independent Vector Problem (SIVP). This allows us to guarantee the claimed security of our cryptosystem unless algorithms that solve this problem improve significantly.

We also consider a slight modification of traditional LWE-based cryptosystems by letting the system use a variable error distribution. As this version of the cryptosystem allows a more efficient reduction, it can be parametrized with smaller parameters while arguing for the same security.

1.2.2 Considerations for Parametrization

In order to use Regev's reduction to parametrize our cryptosystem, we must analyze it in detail. While the efficiency of a reduction is often considered in an asymptotic sense, this is not sufficient for a concrete parametrization of a cryptosystem. Instead, we must know the concrete time the reduction requires to solve the underlying problem when using an adversary against our cryptosystem. This allows us to guarantee that any adversary against the claimed security of our cryptosystem implies an algorithm that is more efficient than what our conservative hardness estimate for the underlying lattice problem predict to be possible.

To account for the efficiency of a reduction, we must consider its run time T_R and success probability p_R. These depend both on the run time T_O of the oracle used by the reduction as well as its success probability p_O. A combined measure for a reductions efficiency is given by its tightness gap $(T_R p_O)/(p_R T_O)$ as defined in [9]. The tightness gap of Regev's original reduction has been analyzed in previous works [9, 14, 33], but these works did not attempt to optimize the reduction for better concrete efficiency. Because of this, we provide our own analysis of the efficiency of a somewhat optimized version of Regev's reduction.

As defined, the tightness gap of a reduction relates to the efficiency of running the reduction. It does however not take into account the concrete hardness of the underlying problem, something which is arguably more important if using the reduction to parametrize a cryptosystem. This aspect of the reduction was not considered in any detail in any of the previous works that analyzed the tightness gap in Regev's reduction.

Our more realistic hardness estimate for approximate SIVP is the primary reason why our parametrizations use significantly larger parameters than the ones proposed in the master thesis of Gates [14], which also accounted for the concrete efficiency of Regev's reduction. Instead of considering the hardness of approximate SIVP, Gates's parametrization was based on the assumption that this problem is as hard as an exact lattice problem. For the relatively large approximation factor for which the reduction solves approximate SIVP, this significantly overestimates the hardness of the underlying problem.

1.2.3 Security Proof and its Efficiency

We argue for the security of our cryptosystem through a series of reductions. Here we present an outline of the different steps and their efficiency. For readability, we omit some constants in this overview, a luxury we naturally can not afford in the full proof.

For the proof, we assume that there is an adversary against our cryptosystem that requires time T to achieve advantage ε_a. Furthermore, we consider versions of our cryptosystem that are parametrized with a dimension n and with elements in $\mathbb{Z}_q$ for some prime q.

The first step in our security proof is to show that we can use this adversary against the OW-CPA security of our cryptosystem to solve a Decision-LWE (DLWE) problem, which is accomplished through a standard hybrid argument. This is detailed in Theorem 3.3 which, with $k = n$, results in an algorithm that solves a DLWE instance with negligible failure probability in time $\mathcal{O}(n \cdot T/\varepsilon_a)$.

While not having a large tightness gap, this step still has a significant impact on the efficiency of the full reduction. This is due it determining how many samples that are required by the constructed DLWE oracle in order to decide if these samples are from an LWE distribution. As we are required to amplify the initially small success probability of the adversary to essentially 1, we require $N = \mathcal{O}(n^2/\varepsilon_a)$ LWE samples. Since the performance of later steps of the reduction depends on this relatively large N, this greatly affects the efficiency of the full reduction.

Next, we use this DLWE oracle in order to solve the search-LWE problem that appears in Regev's reduction. Part of this is accomplished by a search to decision reduction that requires using an DLWE oracle nq times to solve a search-LWE problem with the same error distribution, as detailed in Lemma 2.8.

The search-LWE problem that must be solved in Regev's reduction is actually stated in terms of an unknown error distribution. To handle this unknown error distribution we may use the same approach as Regev used in [31], with somewhat improved analysis. This results in having to use the DLWE oracle a total of $10N \cdot n^2 q$ times with $M = nN$ LWE samples in order to solve this search-LWE problem, as detailed in Lemma 4.1 with $\tau = n$. As N is large, this step has a significant impact on the running time of the reduction.

We can also solve this search-LWE problem more efficiently by considering an DLWE oracle constructed from an adversary against our modified cryptosystem with variable error distribution. This approach, given by Lemma 4.2, only requires using the DLWE oracle nq times to solve the search-LWE problem, while also only requiring $M = N$ different LWE samples. This step of the reduction is thus significantly more efficient when using our modified cryptosystem, which is the reason why it can be parametrized with smaller parameters.

Finally, Regev's quantum reduction allows us to use our ability to solve search LWE in order to solve an arbitrary approximate SIVP instance. By not considering an intermediate reduction from discrete Gaussian sampling, we somewhat improve the efficiency of this step. This allows us to solve the target approximate SIVP instance by using $3n^2M$ calls to an LWE oracle that handles unknown

error distribution, as detailed in Theorem 4.3.

In conclusion, an adversary against the cryptosystem with fixed error distribution can be used to solve approximate SIVP in time

$$\mathcal{O}(qn^6N^2 \cdot T/\varepsilon_a) = \mathcal{O}(qn^{10} \cdot T/\varepsilon_a^3)$$

with constants given in Theorem 5.1 with $k = \tau = n$. Using the above mentioned improvements, an adversary against our cryptosystem with variable error distribution can be used to solve approximate SIVP in time

$$\mathcal{O}(qn^4N \cdot T/\varepsilon_a) = \mathcal{O}(qn^6 \cdot T/\varepsilon_a^2)$$

with constants given in Theorem 5.2 with $k = n$.

An adversary against the claimed 128 bits of security of our cryptosystems will have $T/\varepsilon_a \leq 2^{128}$. By itself, this bound is not sufficient to calculate the running time of the reduction. Instead, we consider the worst-case, where $T = 1$ and $\varepsilon_a = 2^{-128}$. This results in the dependence on ε_a to be one of the most significant reasons for the long run time of the reduction.

As such, one of the largest contributor to the inefficiency of the reduction is the fact that the LWE oracle constructed from an adversary against our cryptosystem requires a large number of LWE samples. Previous analysis of the tightness gap of Regev's reduction have not accounted for this and instead assumed that the provided LWE oracle only requires as many LWE samples as exposed in a single instance of a cryptosystem. [1] However, thanks to our optimizations compared to previous works, our final reduction is still approximately as efficient as the claimed efficiency in these previous works.

For our actual parametrization, we allow the reduction to have a small, but noticeable, failure probability. This failure probability is accounted for in the parametrization, but allows a more efficient reduction with $\tau < n$ and $k < n$ in Theorems 5.1 and 5.2. Furthermore, allowing the reduction to fail with a noticeable probability allows us to argue that it can be used to solve approximate SIVP with a smaller approximation factor than if it had to have a negligible failure probability.

1.2.4 Hardness Estimate

In order to provide any concrete security guarantees from a reduction, we require that an adversary against the cryptosystem could be used with the reduction in order to solve some concrete problem more efficiently than we believe to be

[1] That the LWE oracle provided to Regev's reduction will require many LWE samples was also noticed in a paper by Koblitz et al. [20] that analyzed a similar reduction for ring-LWE.

possible. As such, to parametrize our cryptosystem we need an estimate for the difficulty of solving the underlying lattice problem.

With our security reduction, an adversary against our cryptosystem can be used to solve a worst-case instance of approximate SIVP. This worst-case instance is obviously at least as hard as a random instance of this problem. As nothing indicates that SIVP on random lattices is an easier problem than SIVP on other lattices, assuming that worst-case SIVP is as hard as SIVP on random lattices should not significantly underestimate the hardness of the problem. As such, the hardness of SIVP on random lattices serves as a reasonable hardness estimate for worst-case SIVP.

For our hardness estimate on random lattices, we relate the hardness of SIVP with a well studied approximate version of the shortest vector problem (SVP), namely Hermite-SVP. This is in fact the same problem that is typically considered when estimating the hardness of LWE, allowing much of the previous research into the concrete hardness of LWE to be directly relevant for our parametrization as well.

It is however important to note that the security of our cryptosystem is dependent on the hardness Hermite-SVP in a completely different way compared to a typical LWE-based cryptosystem. For a typical LWE-based cryptosystem, algorithms that solve Hermite-SVP more efficiently would be sufficient to break the cryptosystem. However, these cryptosystems could potentially be vulnerable to other types of attacks, even without lattice algorithms improving.

In contrast to this, our proposed cryptosystem is provably secure as long as lattice algorithms do not improve significantly. As such, while concrete attacks against our cryptosystem could improve, they will be unable to break the claimed 128 bits of OW-CPA security unless Hermite-SVP is a significantly easier problem than we currently believe it is.

1.3 Paper outline

The remainder of the paper begins with some background in section 2, followed by more details about our cryptosystem presented in section 3. Full details regarding the tightness of the optimized version of Regev's quantum reduction is presented in section 4 where a theorem similar to the main theorem from [31] is proven. The proof keeps track of required oracle calls and failure probabilities and tries to minimize these as much as possible. Some of the lemmas used in this proof are more or less unchanged compared to [31]. These lemmas are still included for completeness but are placed in Appendix A. Finally, in section 5 we present the complete proof of security for our cryptosystem which are then used in section 6 to create the different parametrizations of our cryptosystem.

2 Background

2.1 Notation

Matrices are written with bold upper case letters $\boldsymbol{A}, \boldsymbol{S}$ and vectors bold lower case letters $\boldsymbol{a}, \boldsymbol{s}$. Sampling a value x from distribution $\mathcal{X}$ is expressed as $x \leftarrow \mathcal{X}$ and the result of a randomized function f with input x is similarly denoted $y \leftarrow f(x)$. Distributions are expressed with calligraphic letters $\mathcal{X}$ and $\mathcal{U}(X)$ corresponds to the uniform distribution over the set X. The size of a set X is denoted by $|X|$. Concatenation of bitstrings b_0, b_1 is denoted $b_0 \| b_1$, while for matrices and vectors $\boldsymbol{A} \| \boldsymbol{v}$ corresponds to the matrix generated by the columns of the concatenated matrices and vectors.

2.2 Public key cryptography

A public key encryption (PKE) scheme is defined by $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$, a triplet of algorithms for generating keys, encrypting messages and decrypting ciphertexts respectively. The algorithm for key generation outputs both a public key pk and a secret key sk. The encryption algorithm is defined over a message space M and outputs a ciphertext c when given an arbitrary message $m \in M$ and a public key pk. When the decryption algorithm is given a ciphertext c and a secret key sk it outputs a message $m \in M$. The scheme is δ-correct if

$$E\left[\max_{m \in M} \Pr\left[\mathrm{Dec}(sk, c) \neq m \mid c \leftarrow \mathrm{Enc}(pk, m)\right]\right] \leq \delta$$

with expectation taken over $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Gen}()$. The security notion which is relevant for public key encryption schemes in this paper is one wayness under chosen plaintext attacks (OW-CPA) security which is defined from the OW-CPA game shown in Figure 1.

Definition 1 (OW-CPA security)**.** A public key cryptosystem is (T, ε)-secure if any adversary $\mathcal{A}$ running in time at most T has an advantage at most ε in the OW-CPA game where the advantage is given by

$$\Pr\left[\text{OW-CPA}(\mathcal{A})\right] - \Pr\left[\text{OW-CPA}(\mathcal{U}(M))\right] = \Pr\left[\text{OW-CPA}(\mathcal{A})\right] - \frac{1}{|M|} \ .$$

The cryptosystem is said to have d bits of OW-CPA security if it is (T, ε)-secure for every T, ε such that $T/\varepsilon < 2^d$.

A key encapsulation mechanism (KEM) is defined by three different algorithms $(\mathrm{Gen}, \mathrm{Encaps}, \mathrm{Decaps})$ for key generation, encapsulation and decapsulation respectively. The key generation algorithm outputs a pair $(\mathrm{pk}, \mathrm{sk})$ with a

OW-CPA($\mathcal{A}$)	IND-CCA($\mathcal{A}$)
$\overline{(\text{pk}, \text{sk}) \leftarrow \text{Gen}()}$	$\overline{(\text{pk}, \text{sk}) \leftarrow \text{Gen}()}$
$m \leftarrow \mathcal{U}(M)$	$b \leftarrow \mathcal{U}(\{0,1\})$
$c \leftarrow \text{Enc}(\text{pk}, m)$	$(k_0, c) \leftarrow \text{Encaps}(\text{pk})$
$m \leftarrow \mathcal{A}(pk, c)$	$k_1 \leftarrow \mathcal{U}(K)$
return $m' = m$	$b' \leftarrow \mathcal{A}^{\text{Decaps}^*}(c, k_b)$
	return $b' = b$

Figure 1: Games for OW-CPA security of a PKE scheme and IND-CCA security for a KEM. In the IND-CCA game $\mathcal{A}^{\text{Decaps}^*}$ corresponds to the adversary with access to a decapsulation oracle Decaps^* such that $\text{Decaps}^*(c^*) = \text{Decaps}(c^*)$ for all c^* except the challenge ciphertext c.

public and a secret key. The encapsulation algorithm takes a public key as input and outputs a ciphertext c and some key k in the space of possible keys K. The ciphertext is said to encapsulate the key k. The decapsulation algorithm takes a ciphertext and a private key as input and outputs a key k from K.

The security of a KEM in the notion of being indistinguishable under chosen ciphertext attacks (IND-CCA) is given by the following definition with the relevant IND-CCA game shown in Figure 1.

Definition 2 (IND-CCA security)**.** A KEM is (T, ε)-secure if any adversary $\mathcal{A}$ running in time at most T has an advantage at most ε in the IND-CCA game where the advantage of is given by

$$\Pr\left[\text{IND-CCA}(\mathcal{A})\right] - \Pr\left[\text{IND-CCA}(\mathcal{U}(\{0,1\}))\right] = \Pr\left[\text{IND-CCA}(\mathcal{A})\right] - \frac{1}{2}$$

The KEM is said to have d bits of IND-CCA security if it is (T, ε)-secure for every T, ε such that $T/\varepsilon < 2^d$.

2.3 Statistical distance

To measure the similarity between two different distributions, this work makes use of the statistical distance defined as the total variation distance between distributions. The statistical distance between two distributions $\mathcal{X}_1$ and $\mathcal{X}_2$ with probabilities $p_1(x)$ and $p_2(x)$ for possible outcomes $x \in X$ is defined as

$$\Delta(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2} \sum_{\boldsymbol{x} \in X} |p_1(\boldsymbol{x}) - p_2(\boldsymbol{x})| \quad .$$

2.4 Gaussian distributions

Multiple different theorems related to Gaussian distributions are required for the reduction from approximate SIVP to LWE. To begin with we recall that a normal distribution $\mathcal{N}(0, \sigma^2)$ with mean 0 and variance σ^2 has the density function

$$\frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot \exp\left(-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$$

where $\exp(y)$ is used to denote e^y. For an n-dimensional vector $\boldsymbol{x}$, we define $\rho_s(\boldsymbol{x}) = \exp\left(-\pi \left\|\boldsymbol{x}/s\right\|^2\right)$ and we let $\rho = \rho_1$. Furthermore, we let $\nu_s(\boldsymbol{x}) = \rho_s(\boldsymbol{x})/s^n$ which is scaled so that it defines an n-dimensional probability density function and corresponds to a normal distribution with mean 0 and standard deviation $s/\sqrt{2\pi}$.

We also define ρ_s on sets by

$$\rho_s(A) = \sum_{\boldsymbol{x} \in A} \rho_s(\boldsymbol{x})$$

for a countable set A. The discrete Gaussian distribution $D_{\boldsymbol{A},s}$ on a countable set A is defined via

$$\forall \boldsymbol{x} \in A,\ D_{A,s}(\boldsymbol{x}) = \frac{\rho_s(\boldsymbol{x})}{\rho_s(A)}\ .$$

Finally, we define Ψ_α for any $\alpha \in \mathbb{R}^+$ to be a distribution corresponding to sampling from a mean 0 normal distribution with standard deviation $\frac{\alpha}{\sqrt{2\pi}}$ and reducing the result modulo 1, meaning that the distribution can be expressed as

$$\forall r \in [0,1),\ \Psi_\alpha(r) = \sum_{k=-\infty}^{\infty} \frac{1}{\alpha} \cdot \exp\left(-\pi\left(\frac{r-k}{\alpha}\right)^2\right)$$

In order to calculate statistical distance between Ψ_α and Ψ_β the following claim will be used.

Claim 2.1 (Claim 2.2 from [31])**.** *For any* $0 < \alpha < \beta \leq 2\alpha$*,*

$$\Delta(\Psi_\alpha, \Psi_\beta) \leq 9\left(\frac{\beta}{\alpha} - 1\right)$$

The sum of samples from two different mean 0 normal distributions with standard deviation α and β respectively is a mean 0 normal distribution with standard deviation $\sqrt{\alpha^2 + \beta^2}$. This also implies that the sum of a sample from Ψ_α and one sample from Ψ_β is distributed as a sample from $\Psi_{\sqrt{\alpha^2+\beta^2}}$.

The rounded Gaussian distribution $\overline{\Psi}_\alpha$ is defined as a distribution over $\mathbb{Z}_q$ for some implicit q. A sample from this distribution is given by $\lfloor qx \rceil \mod q$ with

$x \leftarrow \Psi_\alpha$. In general we define $\overline{\mathcal{X}}$ for an arbitrary distribution $\mathcal{X}$ over $[0,1)$ to be the rounded distribution over $\mathbb{Z}_q$ given by $\lfloor qx \rceil \mod q$ with $x \leftarrow \mathcal{X}$.

When working with these different versions of Gaussian distributions, we often have to bound the probability of unlikely events. One way we accomplish this is by using this next lemma to bound the probability of sampling an unusually large vector from a discrete Gaussian over a lattice.

Lemma 2.2. *Lemma 2.5 from [31] Let B_n denote the Euclidean unit ball. Then, for any lattice L and any $r > 0$, $\rho_r(L \backslash r\sqrt{n}B_n) < 2^{-2n} \cdot \rho_r(L)$ where $L \backslash r\sqrt{n}B_n$ is the set of lattice points of norm greater than $r\sqrt{n}$.*

Finally, the next two lemmas are also used to limit the probabilities of unlikely events in order to bound the probability of incorrect decryption in the cryptosystem. These lemmas are similar to statements from [23] but differ somewhat as we use a cryptosystem with a rounded Gaussian distribution whereas [23] use a discrete Gaussian distribution.

Lemma 2.3. *For any real $s > 0$, $T > 0$, and any $\boldsymbol{x} \in \mathbb{R}^n$, we have*

$$Pr\Big[\ |\langle \boldsymbol{x}, \nu_s \rangle| \geq T \cdot s \left\|x\right\| \Big] < \cdot \exp\left(-\pi T^2\right)$$

Proof. Multiplying a sample from $\mathcal{N}(0, \sigma^2)$ by a number a results in a sample from $\mathcal{N}(0, (a\sigma)^2)$. The sum of a sample from $\mathcal{N}(0, \sigma_a^2)$ and one from $\mathcal{N}(0, \sigma_b^2)$ results in a sample from $\mathcal{N}(0, \sigma_a^2 + \sigma_b^2)$. As such we have

$$\langle \boldsymbol{x}, \nu_s \rangle = \sum_i x_i \cdot \mathcal{N}\left(0, \frac{s^2}{2\pi}\right) = \sum_i \mathcal{N}\left(0, \frac{s^2 x_i^2}{2\pi}\right) = \mathcal{N}\left(0, \frac{s^2 \left\|x\right\|^2}{2\pi}\right)$$

and the lemma therefore simply states that

$$\Pr\left[\ \left|\mathcal{N}\left(0, \frac{s^2 \left\|x\right\|^2}{2\pi}\right)\right| \geq T \cdot s \left\|x\right\| \right] < 2 \cdot \exp(-\pi T^2)$$

or equivalently

$$\Pr\left[\ \left|\mathcal{N}\left(0, \frac{1}{2\pi}\right)\right| \geq T \right] < 2 \cdot \exp(-\pi T^2) \ .$$

The probability density function for $\mathcal{N}(0, 1/(2\pi))$ is $e^{-\pi x^2}$ and the probability that the absolute value of a sample from this distribution is greater than T is therefore given by

$$2 \cdot \int_{x=T}^{\infty} \exp(-\pi x^2) = 1 - \operatorname{erf}(\sqrt{\pi}T) = \operatorname{erfc}(\sqrt{\pi}T)$$

where erf is the error function and erfc is the complementary error function. As it is known that $\operatorname{erfc}(x) < e^{-x^2}$ [11], the Lemma follows. □

Sampling a vector from $\overline{\Psi}_\alpha^n$ corresponds to sampling a vector from a distribution with density function ν_α, multiplying it by q, rounding each resulting element to its nearest integer and then reducing it modulo q. This following lemma uses this fact to bound the probability of sampling a vector from $\overline{\Psi}_\alpha^n$ which is unusually large by considering vectors sampled from ν_α. When calculating the length of a vector in $\mathbb{Z}_q^n$ we considered the length of the representatives in $(-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$.

Lemma 2.4. *The probability that a vector sampled from $\overline{\Psi}_\alpha^n$ is longer than $c \cdot \sqrt{n}\alpha q/\sqrt{2\pi} + 0.5\sqrt{n}$ is no more than $(c \cdot \exp((1-c^2)/2))^n$ for any $c > 1$.*

Proof. An element from $\overline{\Psi}_\alpha^n$ is a vector sampled from a distribution with probability density function $\nu_{\alpha q}$ with the entires rounded to nearest integer and taken modulo q. The length of vectors does not increase when taken modulo q. Furthermore, rounding a vector to the nearest integer can at most increases the vectors length by $0.5\sqrt{n}$. Because of this, we only have to ensure that the length of a vector sampled from a distribution with density function $\nu_{\alpha q}^n$ is at most $c \cdot \alpha q\sqrt{n/2\pi}$ with the claimed probability in order to prove our statement.

As $\nu_{\alpha q}$ corresponds to a normal distribution with mean zero and standard deviation $\alpha q/\sqrt{2\pi}$, samples from the corresponding distribution can equivalently be considered as a sample from a mean 0 normal distribution with standard deviation 1 times the constant $\alpha q/\sqrt{2\pi}$. The squared length of a vector from a distribution with probability density function $\nu_{\alpha q}$ thus corresponds to a sample from a χ^2 distribution with n degrees of freedom multiplied by $(\alpha q)^2/(2\pi)$.

A sample from a χ^2 distribution is at most $c^2 n$ except for with a probability of at most $(c^2 \exp(1-c^2))^{n/2}$. Thus the probability that a vector sampled from a distribution with probability density function $\nu_{\alpha q}$ is longer than $c \cdot \alpha q\sqrt{n/2\pi}$ is no more than

$$(c^2 \exp(1-c^2))^{n/2} = (c \exp((1-c^2)/2))^n$$

As the length is increased by at most $0.5\sqrt{n}$ by rounding, the total length of a vector from $\overline{\Psi}_\alpha^n$ is no more than

$$c\sqrt{n}\alpha q/\sqrt{2\pi} + 0.5\sqrt{n}$$

with the same probability. □

2.5 Fourier Transform

The Fourier transform is used throughout Regev's reduction from SIVP to LWE. In this paper, we ignore certain technical conditions required by the

Fourier transform as they are always satisfied when relevant. Given a function $h : \mathbb{R}^n \to \mathbb{C}$, its Fourier transform is defined as

$$\hat{h}(\boldsymbol{w}) = \int_{\mathbb{R}^n} h(\boldsymbol{x}) e^{-2\pi i \langle \boldsymbol{x}, \boldsymbol{w} \rangle} d\boldsymbol{x} \,.$$

From this definition of the Fourier transform, it is clear that if $h(\boldsymbol{x}) = g(\boldsymbol{x} + \boldsymbol{v})$ then

$$\hat{h}(\boldsymbol{w}) = e^{2\pi i \langle \boldsymbol{v}, \boldsymbol{w} \rangle} \hat{g}(\boldsymbol{w})$$

while similarly, if $h(\boldsymbol{x}) = e^{2\pi i \langle \boldsymbol{v}, \boldsymbol{x} \rangle} g(\boldsymbol{x})$ then

$$\hat{h}(\boldsymbol{w}) = \hat{g}(\boldsymbol{w} - \boldsymbol{v}) \ .$$

Another property of the Fourier transform that will be used in this paper is that a Gaussian function is its own Fourier transform with $\hat{\rho} = \rho$. More generally we also have that $\hat{\rho}_s = s^n \rho_{1/s}$ for arbitrary s.

2.6 Lattices

An integer lattice L is a subset of $\mathbb{Z}^n$ such that for any two points $\boldsymbol{v}, \boldsymbol{w}$ in the lattice, the difference $\boldsymbol{v} - \boldsymbol{w}$ is also in the lattice. A full rank lattice can be described by an invertible basis $\boldsymbol{B} \in \mathbb{Z}^{n \times n}$, noted as $L = L(\boldsymbol{B})$ where all lattice points $\boldsymbol{v}$ can be written as $\boldsymbol{v} = \boldsymbol{B}\boldsymbol{x}$ for some $\boldsymbol{x} \in \mathbb{Z}^n$. The dual of a lattice is defined as

$$L^* = \{\boldsymbol{v} : \boldsymbol{v} \cdot \boldsymbol{w} \in \mathbb{Z} \ \ \forall \boldsymbol{w} \in L\}$$

and a basis for the dual of a full rank lattice is given by $\boldsymbol{B}^* = (\boldsymbol{B}^{-1})^T$.

The determinant of a full rank lattice is equal to the absolute value of the determinant of an arbitrary basis for the lattice. This is an unambiguous definition as, although there are multiple different bases for a single lattice, it can be shown that the absolute value of the determinant is the same for all of them. The determinant of the dual lattice is the inverse of the determinant of the primal lattice.

The length of the shortest non-zero vector in a lattice L is denoted by $\lambda_1(L)$, which is equivalent to the radius of the smallest ball around the origin that contains a non-zero lattice vector. Similarly, $\lambda_n(L)$ is the radius of the smallest ball that contains n linearly independent lattice vectors.

Several lattice problems are known to be NP-hard. This includes problems such as the closest vector problem (CVP) and the shortest independent vector problem (SIVP). Regev's quantum reduction from [31] is not directly related to any of these NP-hard problems but instead to SIVP_γ, an approximate version of SIVP which is not believed to be NP-hard for the relevant approximation factor. The definition of SIVP_γ follows, with the exact problem corresponding to $\gamma = 1$.

Definition 3 (SIVP_γ)**.** An instance of the approximate shortest independent vector problem (SIVP_γ) is given by an n dimensional lattice L and an approximation factor $\gamma = \gamma(n)$. The goal is to output n linearly independent vectors in L that all are shorter than $\gamma(n) \cdot \lambda_n$.

We will also make use of a more general version of this problem, called the Generalized Independent Vectors Problem (GIVP) with the following definition. The function ϕ in the definition does not have to be efficiently computable and thus SIVP_γ is recovered by using $\phi(L) = \lambda_n(L)$.

Definition 4 ($\mathrm{GIVP}_{\gamma(n)\cdot\phi(L)}$)**.** An instance of $\mathrm{GIVP}_{\gamma(n)\cdot\phi(L)}$ is given by an n dimensional lattice L. The problem is stated with an approximation factor $\gamma = \gamma(n)$ and a real valued function on lattices ϕ. The goal is to output n linearly independent vectors in L that all are shorter than $\gamma(n) \cdot \phi(L)$.

Another problem central to the LWE reduction is the Bounded Distance Decoding (BDD) problem, which is defined next.

Definition 5 ($\mathrm{BDD}_{L,d}$)**.** An instance of the bounded distance decoding problem $\mathrm{BDD}_{L,d}$ is given by a point $\boldsymbol{x}$ that is guaranteed to be at most a distance d from the lattice L. The goal is to output the lattice point in L that is closest to $\boldsymbol{x}$.

When given a point $\boldsymbol{x}$ which is guaranteed to not be further away from the lattice L than $\lambda_1(L)/2$ and thus has a unique closest lattice point, we denote this point by $\kappa_L(\boldsymbol{x})$.

Another lattice problem that will be considered in this work is an approximate version of the Shortest Vector Problem (SVP) that is called Hermite-SVP. In Hermite-SVP a solution is given by a vector that is relatively short compared to the determinant of the lattice. This problem relates to Hermite's constant γ_n, which is given by the maximal value of $\lambda_1(L)^2/\det(L)^{2/n}$ over all n dimensional lattices. For any $\gamma \geq \sqrt{\gamma_n}$ we are thus guaranteed that there is a vector in L which is no larger than $\gamma \cdot \det(L)^{1/n}$. Next follows the definition of γ-Hermite-SVP which consists of finding such a vector that is shorter than $\gamma \cdot \det(L)^{1/n}$.

Definition 6 (γ-Hermite-SVP)**.** An instance of Hermite shortest vector problem (Hermite-SVP) is given by an n dimensional lattice L and an approximation factor γ. The goal is to output a lattice vector of length shorter than $\gamma \cdot \det(L)^{1/n}$.

The exact value of Hermite's constant is only known for some small values of n but can be bounded by by [12]

$$\gamma_n \leq \frac{1.744n}{2\pi e}$$

for large n. In a random lattice, the shortest vector is expected to be shorter than $\sqrt{\gamma_n} \cdot \det(L)^{1/n}$ and its length can instead be predicted by the Gaussian

heuristic. The Gaussian heuristic can asymptotically be proven to be correct for random lattices [10, 32, 37] and predicts that the expected value of $\lambda_1(L)$ is

$$\sqrt{\frac{n}{2\pi e}} \cdot \det(L)^{1/n} \ .$$

The so called smoothing parameter is used extensively throughout Regev's reduction.

Definition 7 (Smoothing parameter)**.** For an n-dimensional lattice L and positive real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(L)$ is defined as the smallest s such that $\rho_{1/s}(L^* \backslash \{0\}) \leq \varepsilon$.

To actually calculate the value of this parameter seems to be an hard problem in it of itself. In the reduction its concrete value is therefore unknown and the following bound is used instead.

Lemma 2.5 (Lemma 2.12 from [31])**.** *For any n-dimensional lattice L and $\varepsilon > 0$,*

$$\eta_\varepsilon(L) \leq \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n(L)$$

One reason for the name of the smoothing parameter is the behaviour that is specified in the following claim. This claim essentially says that a Gaussian distribution with standard deviation greater than the smoothing parameter is more or less uniformly distributed when considered modulo the lattice.

Claim 2.6 (Claim 3.8 from [31])**.** *For any lattice L, $\boldsymbol{c} \in \mathbb{R}^n, \varepsilon > 0$ and $r \geq \eta_\varepsilon(L)$*

$$\rho_r(L + \boldsymbol{c}) \in r^n \det(L^*)(1 \pm \varepsilon)$$

2.7 Learning With Errors

The Learning With Errors (LWE) problem was introduced by Regev in [31] and is parametrized by a dimension n, an integer modulus q and an error distribution $\mathcal{X}$. Regev showed a quantum reduction from SIVP_γ for a polynomially sized γ to the LWE problem with q a polynomially bounded prime and $\mathcal{X}$ the rounded Gaussian distribution.

It was later shown that a similar hardness guarantee holds for arbitrary q [28]. Although other values for q, such as powers of 2, can be useful in practice, using such q requires using additional non-tight reductions. Because of this, only the original reduction that works for polynomially sized prime q is considered in this work. In the reduction from SIVP_γ, the following definition of an LWE distribution is used.

Definition 8 (LWE distribution)**.** Let n, q be positive integers, and let ϕ be a distribution over $[0,1)$. For $\boldsymbol{s} \in \mathbb{Z}^n$, the LWE distribution $A_{\boldsymbol{s},\phi}$ is the distribution over $\mathbb{Z}_q^n \times [0,1)$ obtained by choosing $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random and an error $e \in [0,1)$ from ϕ and outputting the pair

$$\left(\boldsymbol{a}, b = \frac{\langle \boldsymbol{a}, \boldsymbol{s} \rangle}{q} + e \mod 1\right) \in \mathbb{Z}_q^n \times [0,1)$$

However, it is usually preferable to only work with integers when using the LWE problem in applications. Therefore, the following discrete version of an LWE distribution is also used.

Definition 9 (Discrete LWE distribution)**.** Let n, q be positive integers, and let $\mathcal{X}$ be a distribution over $\mathbb{Z}_q$. For $\boldsymbol{s} \in \mathbb{Z}^n$ the LWE distribution $A_{\boldsymbol{s},\mathcal{X}}$ is the distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ obtained by choosing $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random and an integer error $e \in \mathbb{Z}_q$ from $\mathcal{X}$ and outputting the pair

$$(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e \mod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$$

In this work, both types of LWE distributions are considered and can be distinguished by the sample space of its error distribution.

The following lemma shows that the discrete version of the problem with a rounded error distribution is no easier than the continuous version of the problem. The statement in [31] is somewhat different from the one we present here as we require additional information to handle the tightness of the reduction. However, our statement follows from the same proof as in [31] and we therefore do not include a different proof of our variant of the statement.

Lemma 2.7 (Lemma 4.3 from [31])**.** *Let $n, q \geq 1$ be some integer, let ϕ be some probability density distribution on $[0,1)$ and let $\overline{\phi}$ be its discretization to $\mathbb{Z}_q$. There is a transform that, given samples from $A_{\boldsymbol{s},\phi}$ produces the same number of samples from $A_{\boldsymbol{s},\overline{\phi}}$.*

There is both a search and a decision version of the LWE problem. The search problem $\mathrm{LWE}(\mathcal{X}, N)$ is to find the secret $\boldsymbol{s}$ when given at most N samples from $A_{\boldsymbol{s},\mathcal{X}}$.

Meanwhile, the decision learning with errors problem, $\mathrm{DLWE}(\mathcal{X}, N)$ is to determine if an unknown distribution $\mathcal{D}$ is a uniform distribution or $A_{\boldsymbol{s},\mathcal{X}}$ for some $\boldsymbol{s}$ when given at most N samples from $\mathcal{D}$.

The following lemma shows that the search and decision LWE problems are more or less equivalent. This is the same statement as in Lemma 4.2 from [31] except for also including the number of calls and the probability of success.

Lemma 2.8. *Let $n \geq 1$ be some integer, $2 \leq q \leq poly(n)$ be a prime and $\mathcal{X}$ be some distribution on $\mathbb{Z}_q$. Assume that we have access to some procedure W that solves the DLWE$(\mathcal{X}, N)$ problem with failure probability at most ε. Then there exists an algorithm W' that solves the LWE$(\mathcal{X}, N)$ problem with probability at least $1 - nq\varepsilon$ while requiring nq calls to W for each call to W'.*

Proof. The idea is to use W in order to sequentially determine each element of $\boldsymbol{s}$. This is accomplished by guessing the value of an element of $\boldsymbol{s}$ and with the help of W determining if the guess is correct. For the first coordinate this is done by transforming samples $(\boldsymbol{a}, b)$ from $A_{\boldsymbol{s},\mathcal{X}}$ into

$$(\boldsymbol{a} + (r, 0, \ldots, 0)^T, b + r \cdot k) = (\boldsymbol{a}', \langle \boldsymbol{a}', \boldsymbol{s} \rangle + e + r \cdot (k - s_1))$$

where r is selected uniformly at random from $\mathbb{Z}_q$, k is the guessed first coordinate of $\boldsymbol{s}$ and s_1 is the actual value of its first coordinate.

We note that $\boldsymbol{a}' = \boldsymbol{a} + (r, 0, \ldots, 0)^T$ is uniformly random in $\mathbb{Z}_q^n$ and thus if the guess is correct, with $k = s_1$, this transforms takes $A_{\boldsymbol{s},\mathcal{X}}$ to itself. If instead the guess is incorrect, we have that, since q is a prime, $r \cdot (k - s_1)$ is uniformly distributed in $\mathbb{Z}_q$. As such, if the guess is incorrect, the transformed samples are uniformly random in $\mathbb{Z}_q^{n+1}$.

Using W with the transformed samples as input, we determine if the transformed samples are uniformly random or $A_{\boldsymbol{s},\mathcal{X}}$, with a result that is correct except for with probability at most ε. This determines if k is the correct guess for s_1 and we have to try at most q different guesses for s_1 in order to be guaranteed to guess correctly once. Each guess for s_1 corresponds to a single call to W and we require that none of these calls fail.

The process is identical for the other n coordinates, resulting in at most nq calls to W in order to recover the full secret $\boldsymbol{s}$. Each time W is used, it requires N samples from the transformed distribution. However, the same N samples $A_{\boldsymbol{s},\mathcal{X}}$ can be reused through all the nq different transformed distributions that are provided to W. The probability of these N samples giving the incorrect answer in any of the nq calls is at most $nq\varepsilon$, which is our error probability.

□

The average case-version of LWE which is used in our cryptosystems is the so called normal form LWE. In normal form LWE, elements of the secret vector are sampled from the error distribution. This following lemma shows that LWE problems with secrets sampled in this way is essentially as hard as a worst case instance of LWE. This specific version of the lemma is directly taken from [4] with additional notes about the parameter k and the number of samples produced.

Lemma 2.9 (Theorem 5.10 from [4]). *Let $q = p^e$ be a prime power. There is a deterministic polynomial time transformation T that, for arbitrary $\boldsymbol{s} \in \mathbb{Z}_q^n$ and error distribution $\mathcal{X}$, maps $A_{\boldsymbol{s},\mathcal{X}}$ to $A_{\boldsymbol{x},\mathcal{X}}$ where $\boldsymbol{x} \leftarrow \mathcal{X}^n$, and maps $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ to itself. The process fails with probability at most 2^{-k} for arbitrary k by using an initial $n + k$ samples from $A_{\boldsymbol{s},\mathcal{X}}$. After these initial samples, samples from $A_{\boldsymbol{s},\mathcal{X}}$ are directly transformed into samples from $A_{\boldsymbol{x},\mathcal{X}}$.*

2.8 Lattice reduction

In practice the BKZ algorithm [34] is the best performing algorithm for solving the approximate shortest vector problem. However, a similar algorithm, called slide-reduction [13], has better provable performance. Both these algorithms work by considering sub-blocks of a lattice basis and solving a (more-or-less) exact SVP problem on the lattice spanned by these blocks. As these blocks span sub-lattices of a lower dimension than the full lattice, it is easier to solve exact SVP in this sub-lattice than it is to solve it in the full lattice.

When evaluating the security of lattice based schemes one often considers the core-SVP metric [3] where the cost of a lattice reduction with blocksize β is conservatively approximated as the cost of a single call to an SVP solver. There exist multiple different methods to solve the exact SVP instances that are solved in the sub-lattices. The asymptotically most efficient of these are sieving algorithms. These algorithms have a large memory requirement, but in this work we only account for the running time of these algorithms.

The running time of sieving algorithms is typically on the form $2^{C\beta+o(\beta)}$ when running in lattice dimension β and where the constant C depends on the specific algorithm. The best asymptotic performance is achieved by sieving algorithms that run on quantum computers [8] where the minimal value for C is 0.2563. If only considering non-quantum algorithms, the best performance is given by [7] where the constant C is 0.292.

The approximation factor achieved by lattice reduction algorithms improves with increasing block-size β. For performance of the BKZ algorithm on an n-dimensional lattice L, we assume as in [3] that it finds a lattice vector of length $\Delta^n \det L^{1/n}$ with

$$\Delta = \left((\pi\beta)^{1/\beta} \frac{\beta}{2\pi e} \right)^{\frac{1}{2(\beta-1)}} .$$

By using some heuristic assumptions this formula is proven to be asymptotically correct for random lattices in [10].

The slide-reduction algorithm has better provable performance while not performing as well in practice. Slide reduction using blocksize β in dimension n is

able to find vectors of length [13]

$$(\gamma_\beta(1+\varepsilon))^{\frac{n-\beta}{\beta-1}}\lambda_1(L)$$

where γ_β is the Hermite constant in dimension β and ε is a reduction-factor greater than zero which also influences the running time of the algorithm. When we consider the concrete performance of the slide reduction, we conservatively assume that $\varepsilon = 0$ without impacting the algorithms performance. Using that the Hermite constant, for large β, can be bounded by [12]

$$\gamma_\beta \leq \frac{1.744\beta}{2\pi e}$$

this gives that we estimate the slide reduction finds vectors of length

$$\left(\frac{1.744n}{2\pi e}\right)^{\frac{n-\beta}{\beta-1}} \cdot \lambda_1(L) \ . \tag{1}$$

2.9 Quantum states

Here we introduce some concepts related to quantum states and quantum computation but assume some previous familiarity with the subject. For an introduction to quantum computation see for example [27].

A mixed quantum state is a probability distributions of quantum states and can be expressed by a density matrix. The pure quantum state $|\phi\rangle$ corresponds to the density matrix $|\phi\rangle\langle\phi|$ while the mixed state

$$\sum_i p_i|\phi_i\rangle\langle\phi_i|$$

corresponds to a distribution where the quantum state $|\phi_i\rangle$ occurs with probability p_i. The trace distance between two different mixed states is defined as

$$T(\rho,\sigma) = \frac{1}{2}\mathrm{Tr}\left[\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right]$$

and performing the same operations on two different states never increases the trace distance between the states.

When only considering completely classical states, the classical statistical distance and trace distance between two different states is the same. Furthermore, performing measurements on two different quantum states results in outcomes from probability distributions that have a statistical distance that is no larger than the trace distance between the original quantum states [27].

A useful special case for calculating the trace distance is when $\rho = |\psi\rangle\langle\psi|$ and $\sigma = |\phi\rangle\langle\phi|$ are pure quantum states as the trace distance between such states is given by $\sqrt{1 - |\langle\psi|\phi\rangle|^2}$. Using this formula, we prove the following version of Lemma 2.2 that limits the trace distance between two quantum states that are proportional to Gaussian distributions.

Lemma 2.10. *For any n-dimensional lattice L and any $r > 0$, the trace distance between the quantum states proportional to*

$$\sum_{\boldsymbol{x}\in L} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle \tag{2}$$

and

$$\sum_{\boldsymbol{x}\in L, \|x\|<r\sqrt{n}} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle \tag{3}$$

is no larger than 2^{-n}.

Proof. We want to limit the trace distance $T(|\psi\rangle\langle\psi|, |\phi\rangle\langle\phi|)$ where we define the quantum states

$$|\psi\rangle = \sum_{\boldsymbol{x}\in L} \frac{\rho_{\sqrt{2}r}(\boldsymbol{x})}{\sqrt{\rho_r(L)}}|\boldsymbol{x}\rangle = \alpha \sum_{\boldsymbol{x}\in L} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle$$

$$|\phi\rangle = \sum_{\boldsymbol{x}\in L, \|x\|<r\sqrt{n}} \frac{\rho_{\sqrt{2}r}(\boldsymbol{x})}{\sqrt{\rho_r(L \cap r\sqrt{n}B_n)}}|\boldsymbol{x}\rangle = \beta \sum_{\boldsymbol{x}\in L, \|x\|<r\sqrt{n}} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle$$

with normalisation constants α and β. For these states, the special case mentioned above is applicable, giving that the squared trace distance is

$$\begin{aligned} 1 - |\langle\psi|\phi\rangle|^2 &= 1 - \left| \sum_{\boldsymbol{x}\in L, \|x\|<r\sqrt{n}} \alpha\beta \cdot \rho^2_{\sqrt{2}r}(\boldsymbol{x}) \right|^2 \\ &= 1 - \alpha^2\beta^2 \rho_r(L \cap r\sqrt{n}B_n)^2 \end{aligned}$$

where we use that $\rho^2_{\sqrt{2}r}(\boldsymbol{x}) = \rho_r(\boldsymbol{x})$. Using Lemma 2.2 we have that

$$\rho_r(L \cap r\sqrt{n}B_n) = \rho_r(L) - \rho_r(L \setminus r\sqrt{n}B_n) \geq (1 - 2^{-2n})\rho_r(L)$$

which gives

$$\begin{aligned} |\langle\psi|\phi\rangle|^2 = \alpha^2\beta^2 \cdot \rho_r(L \cap r\sqrt{n}B_n)^2 &\geq \frac{(1 - 2^{-2n})\rho_r(L)(\rho_r(L \cap r\sqrt{n}B_n)}{\rho_r(L)\rho_r(L \cap r\sqrt{n}B_n)} \\ &= 1 - 2^{-2n} \end{aligned}$$

and thus the squared trace distance is at most

$$1 - |\langle\psi|\phi\rangle|^2 \leq 2^{-2n} \ .$$

This gives that the trace distance between the $|\psi\rangle$ and $|\phi\rangle$ is at most 2^{-n}. □

A classical function f can be modeled as a unitary quantum operator U by $U|x\rangle|0\rangle = |x\rangle|0 \oplus f(x)\rangle$. This operator is its own inverse and this same operator can be used to erase a register

$$U^{-1}|x\rangle|f(x)\rangle = U|x\rangle|f(x)\rangle = |x\rangle|f(x) \oplus f(x)\rangle = |x\rangle|0\rangle \ .$$

This following lemma shows that we can approximate this unitary operator U if we are given a probabilistic function $g(x;r)$ which almost always equals $f(x)$.

Lemma 2.11. *Let $f(x)$ be a function and $g(x;r)$ be a probabilistic function such that $g(x;r) = f(x)$ except for with probability at most ε with probability taken over the internal randomness r. Furthermore, let X be input space to f, g and R randomness space for g. Then the trace distance between*

$$|\psi\rangle = \frac{1}{\sqrt{|R|}} \sum_{x \in X} \sum_{r \in R} \alpha_x |x\rangle|r\rangle|y_x \oplus f(x)\rangle$$

and

$$|\phi\rangle = \frac{1}{\sqrt{|R|}} \sum_{x \in X} \sum_{r \in R} \alpha_x |x\rangle|r\rangle|y_x \oplus g(x;r)\rangle$$

is no more than $\sqrt{2\varepsilon}$ for arbitrary α_x and y_x.

Proof. First, we note that

$$\frac{1}{|R|} \sum_{r \in R} \langle y_x \oplus f(x)|y_x \oplus g(x;r)\rangle \geq (1 - \varepsilon)$$

as $f(x)$ and $g(x,r)$ are different for at most $\varepsilon \cdot |R|$ different values of r. This gives that

$$\langle\psi|\phi\rangle = \frac{1}{|R|} \sum_{\boldsymbol{x} \in X} \alpha_x^2 \langle x|x\rangle \sum_{r \in R} \langle y_x \oplus f(x)|y_x \oplus g(x;r)\rangle \geq (1 - \varepsilon)$$

and since both of the states are pure quantum states, we have that the trace distance between them is bounded by

$$\sqrt{1 - |\langle\psi|\phi\rangle|^2} \leq \sqrt{1 - |(1-\varepsilon)|^2} = \sqrt{2\varepsilon - \epsilon^2} \leq \sqrt{2\varepsilon} \ .$$

□

3 Cryptosystem Specification

In this paper, we construct a PKE scheme which we parametrize to target 128 bits of OW-CPA security. The cryptosystem is essentially constructed

as the Lindner-Peikert scheme [23], in a similar way to FrodoKEM [26]. Our cryptosystem also supports using a variable error distribution and we provide parametrizations both with fixed and variable error distributions. Using a variable error distribution allows the cryptosystem to be supported by a more efficient security reduction and therefore allows the system to use smaller parameters while arguing for the same security.

The algorithms for key generation, encryption and decryption in our cryptosystem are described in Figure 2. Parameters for the cryptosystem are positive integers $n, \overline{n}$ and B, a prime q and a set $I \subset (0, 1)$ which determines which error distributions the cryptosystem uses. The resulting cryptosystem encrypts $\ell = B \cdot \overline{n}^2$ bits per encryption and has a decryption failure probability that depends on the parametrization, as detailed in Section 3.1.

The specific error distributions used by the cryptosystem are determined by the set I, with the squared standard deviation of the error distribution determined by a sample from $\mathcal{U}(I)$ every time that a new LWE distribution is required. An ordinary LWE-based cryptosystem is recovered by using a set with a single element $I = \{\alpha^2\}$ while our other parametrization uses a larger set with $I = [\alpha^2, 3\alpha^2/2]$.

We parametrize all our cryptosystems with $B = 4$ and with $\overline{n}$ either 8 or 12. Remaining parameters are selected such that $\alpha q \approx 2\sqrt{n}$ and so that the decryption failure probability is sufficiently small. This leads to us selecting $q = \mathcal{O}(n^{3/2})$ and $\alpha = \mathcal{O}(n^{-1})$. Thus, all other parameters are determined by the choice of dimension n and a search over values of n allows us to find the smallest dimension that achieves our targeted security. More details regarding the chosen parameters are presented in Section 6.

The LWE sampling algorithm in Figure 2 returns $(\boldsymbol{b}_i = \boldsymbol{A}\boldsymbol{s}_i + \boldsymbol{e}_i, \boldsymbol{s}_i)$ for a given $\boldsymbol{A}$ and is repeated $\overline{n}$ times in both key generation and encryption. From these sampled columns, we construct the matrices $\boldsymbol{B} = \boldsymbol{b}_1 \| \dots \| \boldsymbol{b}_{\overline{n}}$ and $\boldsymbol{S} = \boldsymbol{s}_1 \| \dots \| \boldsymbol{s}_{\overline{n}}$, which is denoted as $(\boldsymbol{B}, \boldsymbol{S}) = \text{LWEGen}_I(.)^{\overline{n}}$ in the algorithm descriptions.

For encryption, $\boldsymbol{A}$ and $\boldsymbol{B}$ are concatenated and transposed, which is used to generate

$$\boldsymbol{C} = (\boldsymbol{A} \| \boldsymbol{B})^T \boldsymbol{S}' + \boldsymbol{E}' \ .$$

The next step thus splits this matrix to $\boldsymbol{C}_1 \approx (\boldsymbol{S}')^T \boldsymbol{A}$ and $\boldsymbol{V} \approx (\boldsymbol{S}')^T \boldsymbol{B}$ with the approximate equality hiding their respective parts of the error matrix $\boldsymbol{E}'$.

3.1 Correctness of decryption

We choose parameters to our cryptosystems such that there is a small probability δ of incorrect decryption in the system. This decryption failure probability plays

PKE.Gen()
$\boldsymbol{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times n})$
$(\boldsymbol{B}, \boldsymbol{S}) \leftarrow \text{LWEGen}_I(\boldsymbol{A})^{\overline{n}}$
return $(pk = (\boldsymbol{A}, \boldsymbol{B}), sk = \boldsymbol{S})$

PKE.Enc(pk, m)
$(\boldsymbol{A}, \boldsymbol{B}) = pk$
$(\boldsymbol{C}, \boldsymbol{S}') = \text{LWEGen}_I((\boldsymbol{A} \| \boldsymbol{B})^T)^{\overline{n}}$
Split $\boldsymbol{C} \in \mathbb{Z}_q^{(n+\overline{n}) \times \overline{n}}$ into $\boldsymbol{C}_1 \in \mathbb{Z}_q^{\overline{n} \times n}$ and $\boldsymbol{V} \in \mathbb{Z}_q^{\overline{n} \times \overline{n}}$ with $\boldsymbol{C} = \boldsymbol{C}_1^T \| \boldsymbol{V}^T$
$\boldsymbol{M} \in \mathbb{Z}_q^{\overline{n} \times \overline{n}} = \text{encode}(m)$
$\boldsymbol{C}_2 = \boldsymbol{V} + \boldsymbol{M} \mod q$
return $ct = (\boldsymbol{C}_1, \boldsymbol{C}_2)$

PKE.Dec$(\boldsymbol{S} = sk, ct)$
$(\boldsymbol{C}_1, \boldsymbol{C}_2) = ct$
$m = \text{decode}(\boldsymbol{C}_2 - \boldsymbol{C}_1 \boldsymbol{S} \mod q)$
return m

$\text{LWEGen}_J(\boldsymbol{A} \in \mathbb{Z}_q^{m \times k})$
$\alpha^2 \leftarrow \mathcal{U}(J)$
$\boldsymbol{s} \leftarrow \overline{\Psi}_\alpha^k$; $\boldsymbol{e} \leftarrow \overline{\Psi}_\alpha^m$
$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \mod q$
return $(\boldsymbol{b}, \boldsymbol{s})$

Encoding encode(m) with $|m| = \ell = B \cdot \overline{n}^2$
Split m into B-bit substrings and interpret as numbers b_i, for integers $0 \leq i < \overline{n}^2$
Let $\boldsymbol{M}$ be a $\overline{n} \times \overline{n}$ matrix
On position x, y in $\boldsymbol{M}$ let it have the value $b_{x+\overline{n} \cdot y} \cdot \lfloor q/2^B \rfloor$
return $\boldsymbol{M}$

Decoding decode$(\boldsymbol{M})$ with $\boldsymbol{M} \in \mathbb{Z}_q^{\overline{n} \times \overline{n}}$
For integers $0 \leq x < \overline{n}$ and $0 \leq y < \overline{n}$ let $M_{x,y}$ be the element on position x, y in $\boldsymbol{M}$
Let $b_{x+y \cdot \overline{n}} = \lfloor M_{x,y} \cdot 2^B / q \rceil \mod 2^B$
return $m = b_0 \| b_1 \ldots \| b_{\overline{n}^2 - 1}$, bitstring combined from all b_i

Figure 2: Algorithms for the cryptosystem with PKE.Gen() for key-generation, PKE.Enc(pk, m) for encryption and PKE.Dec(sk, ct) for decryption with the other algorithms used as subroutines.

an important role in the FO$^{\not\perp}$ transform detailed in subsection 6.3 and a too large decryption failure probability will also limit the usefulness of our PKE.

Parameters for our cryptosystem are therefore selected so that the decryption failure probability is sufficiently small. The following lemma, adapted from [23], allows us to bound the probability of incorrect decryption in our OW-CPA secure PKE scheme. This lemma gives a probability δ of incorrect decryption for a single symbol, not for the entire ciphertext. As such, the probability that the entire ciphertext is decrypted correctly, meaning that all of the $\overline{n}^2$ symbols are recovered correctly, is at least $1 - \overline{n}^2 \delta$ by a union bound.

Lemma 3.1. *Let $\delta > 2^{-n}$ be a real number, ζ^2 be the maximal value in I and assume that $\zeta q > 2\sqrt{n}$ and $\zeta < n^{-1/2} 2^{-(B+4)}$. Then, the decryption error*

probability per symbol is bounded from above by δ if

$$(\zeta q)^2 \leq \frac{q\sqrt{\pi}}{2^{B+2}\sqrt{2n\ln(1/\delta)}} \ .$$

Proof. The proof of this lemma is similar to the proof of Lemma 3.1 of [23], but using a rounded Gaussian distribution instead of a discrete Gaussian distribution.

First, we note that the decryption of a message is given by

$$\begin{aligned}\text{decode}(\boldsymbol{C}_2 - \boldsymbol{C}_1\boldsymbol{S}) &= \text{decode}(\boldsymbol{S}'\boldsymbol{B} + \boldsymbol{E}'' + \text{encode}(m) - (\boldsymbol{S}'\boldsymbol{A} + \boldsymbol{E}')\boldsymbol{S}) \\ &= \text{decode}(\boldsymbol{S}'(\boldsymbol{A}\boldsymbol{S} + \boldsymbol{E}) + \boldsymbol{E}'' + \text{encode}(m) - (\boldsymbol{S}'\boldsymbol{A} + \boldsymbol{E}')\boldsymbol{S}) \\ &= \text{decode}(\text{encode}(m) + \boldsymbol{S}'\boldsymbol{E} + \boldsymbol{E}'' - \boldsymbol{E}'\boldsymbol{S}) \ .\end{aligned}$$

A single element of $\text{decode}(\text{encode}(m) + \boldsymbol{X})$ can be written as

$$\left\lfloor (m_{x,y} \cdot \left\lfloor q/2^B \right\rfloor + X_{x,y} \mod q) \cdot 2^B/q \right\rceil \mod 2^B$$

where $m_{x,y}$ is a B bit number corresponding to B bits of m. From this we see that if $|X_{x,y}| < q/2^{B+1}$ this coefficient equals $m_{x,y}$ and is therefore correctly decoded. As such, decoding an encoded message is correct even if adding noise $\boldsymbol{X}$ as long as all elements of $\boldsymbol{X}$ are smaller than $t = q/2^{B+1}$. Decryption is therefore correct if the elements of $\boldsymbol{S}'\boldsymbol{E} + \boldsymbol{E}'' - \boldsymbol{E}'\boldsymbol{S}$ are smaller than t. We note that the probability that these elements are larger than t increases as the standard deviation for the error distributions increase. Therefore, we consider the worst case, where all error distributions are $\overline{\Psi}_\zeta$ with the maximal $\zeta^2 \in I$.

As we are bounding the per symbol decryption error probability, we want to bound the probability that an arbitrary element of $\boldsymbol{S}'\boldsymbol{E}+\boldsymbol{E}''-\boldsymbol{E}'\boldsymbol{S}$ is larger than t. A single element of this matrix can be represented by $\langle \boldsymbol{s}', \boldsymbol{e}\rangle + e'' - \langle \boldsymbol{e}', \boldsymbol{s}\rangle$ where $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{s}', \boldsymbol{e}'$ all are vectors in their respective matrices and e'' is an element of $\boldsymbol{E}''$. We introduce $\boldsymbol{v} = \boldsymbol{e}\|(-\boldsymbol{e}')\|e''$ and $\boldsymbol{w} = \boldsymbol{s}'\|\boldsymbol{s}\|1$ and can thus rephrase the required inequality as $\langle \boldsymbol{v}, \boldsymbol{w}\rangle < t$. As $\overline{\Psi}_\zeta$ is a symmetric distribution we know that the elements of $-\boldsymbol{e}'$ follow the same distribution as $\boldsymbol{e}$. Therefore, all elements in $\boldsymbol{v}$ and the first $2n$ elements of $\boldsymbol{w}$ are distributed as if sampled from $\overline{\Psi}_\zeta$. This allows limiting the norm of $\boldsymbol{w}$ by using Lemma 2.4 on the first $2n$ coordinates. With $c = 2$ this gives

$$\|\boldsymbol{w}\| \leq 2\zeta q \cdot \sqrt{\frac{n}{\pi}} + \sqrt{\frac{n}{2}} + 1 \leq 2\zeta q\sqrt{n} \tag{4}$$

where the second inequality holds if $1 + \sqrt{n/2} < 2\zeta q\sqrt{n}(1 - \frac{1}{\sqrt{\pi}})$ which is true with $\zeta q > 2\sqrt{n}$ and $n \geq 1$. The choice of $c = 2$ results in an inequality that holds except for with probability at most $C^n = 2^n \exp(-3n/2) \leq 2^{-n}$.

Next, we consider $\boldsymbol{v}$ that is distributed as $\overline{\Psi}_\zeta^{2n+1}$. The vectors $\boldsymbol{v}$ are thus sampled from a distribution with probability density function $\nu_{\zeta q}$, with the elements of the sampled vector rounded to the nearest integer modulo q. However, as we consider $\zeta < 2^{-(B+4)}/\sqrt{n}$, the probability that elements of the sampled vector are not in the range $[-(q-1)/2, (q-1)/2]$ is exponentially small. Thus, besides with an insignificant probability, the vector $\boldsymbol{v}$ is the same as if sampled from the same distribution and rounded to the nearest integer, without performing the modulo q operation.

Compared to if $\boldsymbol{v}$ was sampled from a distribution with probability density function $\nu_{\zeta q}$, the value of $|\langle \boldsymbol{v}, \boldsymbol{w} \rangle|$ cannot increase by more than $\|\boldsymbol{w}\|$ from the rounding. By instead considering $\boldsymbol{z}$ distributed according to $\nu_{\zeta q}$ we are able to use Lemma 2.3 to see that

$$\Pr\Big[\, |\langle \boldsymbol{z}, \boldsymbol{w} \rangle| \geq T \cdot \zeta q \, \|\boldsymbol{w}'\| \,\Big] < \exp\left(-\pi T^2\right)$$

and thus

$$\Pr\Big[\, |\langle \boldsymbol{v}, \boldsymbol{w} \rangle| \geq T \cdot \zeta q \, \|\boldsymbol{w}\| + \|\boldsymbol{w}\| \,\Big] < \exp\left(-\pi T^2\right) \quad . \tag{5}$$

Next, we select $T = t/(\zeta q \, \|\boldsymbol{w}\|) - 1/(\zeta q)$ in order to bound $\Pr\Big[\, |\langle \boldsymbol{v}, \boldsymbol{w} \rangle| \geq t \,\Big]$ which in combination with (4) gives

$$T \geq \frac{t}{2\zeta^2 q^2 \sqrt{n}} - \frac{1}{\zeta q} \quad .$$

Together with (5) this gives that $|\langle \boldsymbol{v}, \boldsymbol{w} \rangle| \geq t$ with probability at most

$$\delta = \exp\left(-\pi\left(\frac{t^2}{4\zeta^4 q^4 n} - \frac{t}{\zeta^3 q^3 \sqrt{n}} + \frac{1}{4\zeta^2 q^2}\right)\right) \leq \exp\left(-\pi \frac{t^2}{8\zeta^4 q^4 n}\right)$$

with $\zeta < n^{-1/2} 2^{-(B+4)}$ guaranteeing the inequality. This gives a bound on ζq in terms of δ as

$$(\zeta q)^4 \leq \frac{\pi t^2}{8n \cdot \ln(1/\delta)}$$

which, by taking a square root and inserting $t = q/2^{B+1}$, gives the bound claimed by the lemma. We ignore the probability that the inequality in (4) does not hold and the probability that elements from $\nu_{\zeta q}$ are not in the desired range. This is justified as these probabilities are insignificant compared to the actual targeted decryption failure probabilities where $\delta \gg 2^{-n/4}$. □

By using Lemma 3.1, we select parameters so that the probability of decryption failures is limited. Using a fixed error distribution $I = \{\alpha^2\}$ we require that

$\alpha q > 2\sqrt{n}$ in order to use Theorem 5.1. Selecting $\alpha q = \mu\sqrt{n}$ for some $\mu > 2$ and inserting into the bound given by Lemma 3.1 gives

$$\mu^2 n \leq \frac{q\sqrt{\pi}}{2^{B+2}\sqrt{2n\ln(1/\delta)}} \ .$$

This bounds q via

$$q \geq \mu^2 \cdot 2^{B+2} \cdot n^{3/2} \cdot \sqrt{\frac{2\ln(1/\delta)}{\pi}}$$

and also gives

$$\alpha = \frac{\mu\sqrt{n}}{q} \leq \frac{1}{2^{B+2}\mu n\sqrt{2\ln(1/\delta)}}$$

which, at least if $n \geq 2$, is sufficiently small for Lemma 3.1 that requires $\alpha < \frac{1}{2^{B+4}\sqrt{n}}$. In order to use Theorem 5.1 to solve SIVP_γ with the smallest possible approximation factor γ, we select parameters that minimize q. Therefore, we select μ as the smallest value greater than 2 such that q is a prime.

Essentially the same analysis holds for the cryptosystem parametrized with $I = [\alpha^2, 1.5\alpha^2]$. For Theorem 5.2 to be applicable we still require $\alpha q > 2\sqrt{n}$. However, for Lemma 3.1 we now have to consider $\gamma^2 = 1.5\alpha^2$ as this is the maximal value of I. Because of this we select parameters with $\gamma q = \mu\sqrt{n}$ but with $\mu > 2 \cdot \sqrt{1.5} = \sqrt{6}$. The other calculations for the parametrization of the cryptosystem with fixed error distribution are also applicable for this parametrization. We therefore chose parameters similarly with $\mu > \sqrt{6}$ chosen so that q is as small prime as possible, minimizing the approximation factor of Theorem 5.2.

3.2 Security

In order to argue for the OW-CPA security of our cryptosystem we begin by showing that an adversary against this system can be used to solve an arbitrary DLWE instance. By using an adversary with low success probability we are able to solve an arbitrary DLWE instance with essentially probability 1. While this requires using the adversary several times, this is partially compensated for by the increased success probability. However, each time the adversary is used, new samples are required from the target DLWE distribution. An adversary with a low success probability thus leads to an DLWE oracle that requires a large number of samples from the unknown distribution. This greatly impacts the tightness of Regev's quantum reduction from [31] as its performance directly depends on this number of samples that the DLWE oracle requires.

An adversary against our cryptosystem parametrized with a variable error distributions can be used to solve a DLWE problem with variable error distribution. To formalize this, we introduce the following definition for an LWE

distribution with random error distributions. We define this as a distribution over LWE distributions such that all sampled LWE distributions share the same secret. The different LWE distributions do however not necessarily share the same error distribution.

Definition 10 (LWE with variable error distribution)**.** Let n, q be positive integers, $J \subset [0, 1)$ be a set and $\boldsymbol{s} \in \mathbb{Z}_q^n$. We define $A_{\boldsymbol{s},J}$ to be a distribution where a sample consists of the distribution $A_{\boldsymbol{s},\Psi_\zeta}$ with $\zeta^2 \leftarrow \mathcal{U}(J)$.

We also define a decision LWE problem related to these distributions of distributions. This DLWE(J, N) problem is a more general version of the ordinary DLWE(Ψ_α, N) problem, which is recovered by simply letting $J = \{\alpha^2\}$.

Definition 11. Let $J \subset [0, 1)$ be a set and n, q, m be positive integers. Let $\mathcal{D}$ either be $A_{\boldsymbol{s},J}$ or a distribution of distributions that always returns $\mathcal{U}(\mathbb{Z}_q^{n+1})$. The DLWE$(J, N)$ problem is to determine which is the case when given a total of at most N samples in $\mathbb{Z}_q^{n+1}$ from distributions given by $\mathcal{D}$.

This DLWE(J, N) problem does not directly occur in Regev's reduction from SIVP$_\gamma$. Instead, Regev's reduction requires an oracle that recovers $\boldsymbol{s}$ when given $A_{\boldsymbol{s},\Psi_\beta}$ for unknown $\beta \in [\alpha/\sqrt{2}, \alpha]$. However, by using this following lemma and Lemma 2.8 we see that finding this $\boldsymbol{s}$ is possible if we can solve the DLWE(I', N) problem with $I' = [\beta^2, \beta^2 + \alpha^2]$.

Lemma 3.2. *Let α, β be two positive numbers and let $I' = [\beta^2, \beta^2 + \alpha^2]$. Then, an instance of the DLWE(Ψ_β, N) problem can be transformed into an instance of the DLWE(I', N) problem without knowing β.*

Proof. For each new distribution requested in the DLWE(I', N) instance, sample a new ζ uniformly at random from $[0, \alpha^2]$. Samples from this new distribution are produced by sampling $(\boldsymbol{a}, b)$ from the DLWE(Ψ_β, N) instance and returning $(\boldsymbol{a}, b + e \mod 1)$ where e is sampled from $\Psi_{\sqrt{\zeta}}$.

If the input distribution was uniformly random, then so is the output distribution. If instead $b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e'$ for some $\boldsymbol{s}$ and with $e' \leftarrow \Psi_\beta$, we see that the sum $e + e' \mod 1$ is distributed as $\Psi_{\sqrt{\beta^2+\zeta}}$. Therefore, the resulting distribution is $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$. As $\zeta \leftarrow \mathcal{U}([0, \alpha^2])$, this corresponds exactly to $A_{\boldsymbol{s},\Psi_{\sqrt{\zeta'}}}$ with $\zeta' \leftarrow \mathcal{U}([\beta^2, \beta^2 + \alpha^2])$ which is the expected distribution for an DLWE(I', N) instance. □

As β is unknown and varies, this set I' can not directly be used in our cryptosystem. However, we ensure that I' contains I for every possible β and that $|I'| / |I|$ is not too large. This implies that a distribution from $A_{\boldsymbol{s},I'}$ is distributed as if sampled from $A_{\boldsymbol{s},I}$ with relatively high probability. This allows

us to use Theorem 3.3 with an adversary against our cryptosystem to solve the DLWE(I') problem, even though I' is unknown.

Theorem 3.3 also details the number of required calls to the adversary and the required number of samples from the unknown distribution in the target DLWE(I') problem. As adversaries with small advantage ε_a must be considered, we see that the primary contributor to both number of oracle calls and required samples is a factor $1/\varepsilon_a$. However, as the success probability is amplified from approximately ε_a to essentially 1, the actual tightness gap of this reduction does not depend on ε_a.

Theorem 3.3. *Let $I' \subseteq (0,1)$ be some unknown set that contains I such that $|I| / |I'| = \kappa \leq 1$ and let $k > 0$ be an integer. Furthermore, assume that $\mathcal{A}$ is an algorithm with advantage ε_a against the OW-CPA security of our PKE and that $\varepsilon_a \kappa^{2\overline{n}}/2 \geq 2^{-\ell}$. Then, we can solve a DLWE(I', N) instance where $N = 3n \cdot 2^{14} \frac{\overline{n}^2 k}{\varepsilon_a \kappa^{4\overline{n}}}$ by using $\mathcal{A}$ no more than $2^{15} \frac{\overline{n}^3 k}{\varepsilon_a \kappa^{4\overline{n}}}$ times. The resulting DLWE algorithm has a failure probability of at most $2(\overline{n}+1)2^{-k}$.*

The proof of Theorem 3.3 uses the fact that an adversary will have no advantage against a version of the cryptosystem that use uniformly random samples instead of LWE samples. As multiple different secrets are used for the same encryption, each sharing the same public matrix $\boldsymbol{A}$, we can not directly replace all LWE distributions in the cryptosystem with the unknown distribution. Instead, a hybrid argument over all the different LWE distributions in the cryptosystem shows that an adversary will have a noticeable difference in success probability between some hybrid games that only differ by one LWE distribution.

We then consider these hybrid versions of our cryptosystem with the critical LWE distribution replaced with the input unknown distribution. The adversary will thus have a noticeable difference in success probability depending on if the unknown distribution is an LWE distribution or is uniformly random. Estimating the success probability of the adversary in this case thus allows us to solve the input DLWE problem. To actually prove this theorem while keeping track of all the details requires quite a bit of work and we therefore dedicate the next subsection to this proof.

In comparison to Theorem 3.3 with a tightness gap that does not depend on ε_a, previous tightness analysis have directly used the worst-case to average-case reduction from [31]. Using this worst-case to average-case reduction results in a much larger tightness gap that depends on ε_a. The smaller tightness gap of our theorem is possible partly thanks to directly relating the systems OW-CPA security to the hardness of worst-case DLWE, without considering an intermediate average-case version of this problem.

3.3 Proof of Theorem 3.3

This section is dedicated to the proof of Theorem 3.3, which shows that an adversary against the OW-CPA security of our cryptosystem can be used to solve DLWE. In order to use it for our concrete parametrization we keep track of the efficiency of the reduction as well as the success probability of the resulting algorithm. Furthermore, the proof also keeps track of the number of samples the resulting decision algorithm requires from the unknown distribution.

The proof of Theorem 3.3 uses a number of hybrid PKE schemes, as defined in Figure 3. As these hybrid schemes are only used to test an adversary, they do not include any secret keys or an algorithm for decryption. The hybrid cryptosystem H_j^I is the original cryptosystem but with the j first LWE distributions replaced by uniform distributions. As both key-generation and encryption use $\overline{n}$ separate LWE distributions, this defines different cryptosystems for $0 \leq j \leq 2\overline{n}$.

An adversary against the OW-CPA security of our cryptosystem, corresponding to H_0^I, will correctly decrypt messages with a noticeable probability. Meanwhile, no algorithm has an advantage over random guessing when messages are hidden by uniformly random noise, which is the case in hybrid cryptosystem $H_{2\overline{n}}^I$. As such, a standard hybrid argument shows that there is some j such that the adversary can distinguish between H_j^I and H_{j+1}^I, allowing using the adversary to solve DLWE(I, N).

Theorem 3.3 actually claims something stronger, namely that the adversary can be used to solve DLWE(I', N) for some unknown I' that contains I. This is shown by also considering hybrid versions $H_j^{I'}$ that are similar to H_j^I but with the set I replaced with I' for one of the LWE distributions that the cryptosystem uses. For each encryption with cryptosystem H_j^J only a single sample is used from $\mathcal{U}(J)$. As such, if I and I' are similar, an adversary against cryptosystem H_i^I will work almost as well against cryptosystem $H_i^{I'}$.

By using an adversary with a sufficiently large advantage compared to random guessing, we can see that there is some j such that it has a noticeable difference in success probability between $H_j^{I'}$ and H_{j+1}^I. As such, the adversary can be used to distinguish between $H_j^{I'}$ and H_{j+1}^I, allowing us to solve DLWE(I').

The following lemma details how we transform a DLWE(I') instance into different hybrid versions of our cryptosystem dependent on the answer to the DLWE instance.

Lemma 3.4. *Let $n, q, \overline{n}$ be positive integers, $0 \leq j < 2\overline{n}$ be an integer and $\mathcal{D}$ a distribution over distributions which either always returns a uniform distribution over $\mathbb{Z}_q^{n+1}$ or is $A_{\boldsymbol{s},I'}$ for some set I' and $\boldsymbol{s} \in \mathbb{Z}_q^n$. It is then possible to encrypt an arbitrary message using H_h^J where $(h, J) = (j, I')$ if $\mathcal{D} = A_{\boldsymbol{s},I'}$ and $(h, J) = (j+1, I)$ if samples from $\mathcal{D}$ are uniform distributions. For each encrypted*

PKE.Gen()
$\boldsymbol{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times n})$
$\boldsymbol{B}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times \min(j,\overline{n})})$
Let $k = \overline{n} - j - 1$
$(\boldsymbol{b}, \boldsymbol{s}) \leftarrow \mathrm{LWEGen}_J(\boldsymbol{A})$
$(\boldsymbol{B}_2, \boldsymbol{S}) \leftarrow \mathrm{LWEGen}_I(\boldsymbol{A})^k$
return $pk = (\boldsymbol{A}, \boldsymbol{B}_1 \| \boldsymbol{b} \| \boldsymbol{B}_2)$

PKE.Enc(m)
$\boldsymbol{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times n})$
$\boldsymbol{B} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n\times \overline{n}})$
Let $h = \max(0, 2\overline{n} - j - 1)$
Let $\boldsymbol{A'} = (\boldsymbol{A} \| \boldsymbol{B})^T$
$\boldsymbol{E}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{(\overline{n}+n)\times(j-\overline{n})})$
$(\boldsymbol{b}, \boldsymbol{s}) \leftarrow \mathrm{LWEGen}_J(\boldsymbol{A'})$
$(\boldsymbol{E}_2, \boldsymbol{S'}) = \mathrm{LWEGen}_I(\boldsymbol{A'})^h$
Let $\boldsymbol{C} = \boldsymbol{E}_1 \| \boldsymbol{b} \| \boldsymbol{E}_2$
Split $\boldsymbol{C} \in \mathbb{Z}^{(n+\overline{n})\times\overline{n}}$ into $\boldsymbol{C}_1 \in \mathbb{Z}^{\overline{n}\times n}$ and $\boldsymbol{V} \in \mathbb{Z}^{\overline{n}\times\overline{n}}$ with $\boldsymbol{C} = \boldsymbol{C}_1^T \| \boldsymbol{V}^T$
$\boldsymbol{M} \in \mathbb{Z}^{\overline{n}\times\overline{n}} = \mathrm{encode}(m)$
$\boldsymbol{C}_2 = \boldsymbol{V} + \boldsymbol{M} \mod q$
return $ct = (\boldsymbol{C}_1, \boldsymbol{C}_2), pk = (\boldsymbol{A}, \boldsymbol{B})$

Figure 3: The hybrid version H_j^J of the cryptosystem. If $j < \overline{n}$ only the key generation is altered and the encryption procedure is identical to original cryptosystem. If instead $j \geq \overline{n}$ no key generation is required as it is uniformly random and generated as part of the encryption algorithm. The $\mathrm{LWEGen}_I(\boldsymbol{A})$ method is defined with the ordinary cryptosystem in Figure 2.

message, we require $3n$ *samples from a single distribution sampled from* $\mathcal{D}$ *in order to ensure that the encryption succeeds except for with a probability of at most* $2^{\overline{n}-n}$.

Proof. The transform is described in Figure 4 and uses a subroutine denoted by NormalForm($\mathcal{D}$), corresponding to Lemma 2.9, in order to transform a random distribution $\mathcal{C}'$ from $\mathcal{D}$ into $\mathcal{C}$. If $\mathcal{C}'$ initially was uniformly distributed then $\mathcal{C}$ is also uniformly distributed while if $\mathcal{C}'$ was $A_{\boldsymbol{s},\Psi_\gamma}$ for some $\gamma^2 \leftarrow I'$ then $\mathcal{C}$ is distributed as $A_{\boldsymbol{s}',\overline{\Psi}_\gamma}$ where $\boldsymbol{s}' \leftarrow \overline{\Psi}_\gamma$. Note that we also let this subroutine transform the input distribution to a rounded distribution over $\mathbb{Z}_q^{n+1}$.

Comparing the algorithms in Figure 3 and Figure 4 the correctness of the transform is clear. Instead of using an LWE distribution from $\mathrm{LWEGen}_J(\cdot)$, the system uses the unknown distribution. Thus, it corresponds to $H_j^{I'}$ if the unknown distribution is an LWE distribution and otherwise is H_{j+1}^I.

Selecting $k = n - \overline{n}$ for the normal form transformation of Lemma 2.9 leads to the claimed failure probability. As at most $n + \overline{n}$ samples are required from the transformed distribution, this corresponds to a total requirement of at most $3n$ samples. □

The proof of Theorem 3.3 requires that we distinguish between different prob-

PKE.Gen()
$\overline{\mathcal{C}} \leftarrow \text{NormalForm}(\mathcal{D})$
$(\boldsymbol{A}, \boldsymbol{b}) \leftarrow \mathcal{C}^n$
Let $k = \overline{n} - j - 1$
$\boldsymbol{B}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{n \times \min(j, \overline{n})})$
$(\boldsymbol{B}_2, \boldsymbol{S}) \leftarrow \text{LWEGen}_I(\boldsymbol{A})^k$
return $pk = (\boldsymbol{A}, \boldsymbol{B}_1 \| \boldsymbol{b} \| \boldsymbol{B}_2)$

PKE.Enc(m)
$\overline{\mathcal{C}} \leftarrow \text{NormalForm}(\mathcal{D})$
$(\boldsymbol{A}', \boldsymbol{b}) \leftarrow \mathcal{C}^{n+\overline{n}}$
Let $h = \max(0, 2\overline{n} - j - 1)$
$\boldsymbol{E}_1 \leftarrow \mathcal{U}(\mathbb{Z}_q^{(\overline{n}+n) \times (j-\overline{n})})$
$(\boldsymbol{E}_2, \boldsymbol{S}') = \text{LWEGen}_I(\boldsymbol{A}')^h$
Let $\boldsymbol{C} = \boldsymbol{E}_1 \| \boldsymbol{b} \| \boldsymbol{E}_2$
Split $\boldsymbol{C} \in \mathbb{Z}^{(n+\overline{n}) \times \overline{n}}$ into $\boldsymbol{C}_1 \in \mathbb{Z}^{\overline{n} \times n}$ and $\boldsymbol{V} \in \mathbb{Z}^{\overline{n} \times \overline{n}}$ with $\boldsymbol{C} = \boldsymbol{C}_1^T \| \boldsymbol{V}^T$
$\boldsymbol{M} \in \mathbb{Z}^{\overline{n} \times \overline{n}} = \text{encode}(m)$
$\boldsymbol{C}_2 = \boldsymbol{V} + \boldsymbol{M} \mod q$
Let $\boldsymbol{A} \| \boldsymbol{B} = (\boldsymbol{A}')^T$
return $ct = (\boldsymbol{C}_1, \boldsymbol{C}_2), pk = (\boldsymbol{A}, \boldsymbol{B})$

Figure 4: The method to use an unknown distribution $\mathcal{D}$ to encrypt messages with hybrid cryptosystem H_h^J for (h, J) depending on $\mathcal{D}$. When $j < \overline{n}$ only the key generation is altered, while if $j \geq \overline{n}$ no key generation is required with pk created in the encryption algorithm instead.

ability distributions that are noticeably different. This is accomplished with the following lemma that corresponds to using a version of the Chernoff-Hoeffding theorem multiple times.

Lemma 3.5. *Let A and B be Bernoulli distributed random variables that have unknown success probabilities p_A and p_B respectively. Also, let $k \geq 3$ be an arbitrary integer. Given some c and δ such that $\delta < p_A \leq c\delta$ it is possible to determine if $p_A - p_B < \delta$ or $p_A - p_B > \delta/2$ with either answer being correct if both inequalities hold. This requires at most $2^8 k \frac{c}{\delta}$ samples from each of A and B while giving a correct result except for with a probability of at most 2^{-k}.*

The proof of this lemma essentially consists of using the Chernoff-Hoefding theorem multiple times, letting it distinguish two different Bernoulli distributions with a sufficiently large difference in success probability. We therefore begin by presenting the version of the Chernoff-Hoeffding theorem that we will be using.

Theorem 3.6 (Variant of Theorem 1 from [17])**.** *If $X_1, \ldots, X_n$ are independent and $0 \leq X_i \leq 1$ for $i = 1, \ldots, n$ then for $0 < t < 1 - \mu$*

$$\Pr\left[\frac{1}{n}\sum_i X_i \geq \mu + t\right] \leq e^{-D(\mu+t\|\mu)n}$$

and

$$\Pr\left[\frac{1}{n}\sum_i X_i \leq \mu - t\right] \leq e^{-D(\mu-t\|\mu)n}$$

where $\mu = \frac{1}{n}E(\sum_i X_i)$ *is the expected mean of all the* X_i*.*

In these expression $D(x\|y)$ is the Kullback-Leibler which is bounded via

$$D(x\|y) \geq \frac{(x-y)^2}{2x}, \qquad x \geq y \tag{6}$$

and

$$D((1+x)p\|p) \geq \frac{1}{4}x^2 p, \qquad -\frac{1}{2} \leq x \leq \frac{1}{2}\ . \tag{7}$$

We now prove Lemma 3.5 by using this version of the Chernoff-Hoeffding Theorem multiple times. This allows us to distinguish between two input Bernoulli distributions and determine which of the input distribution has the highest success probability.

Proof of Lemma 3.5. By taking N samples from the different distributions we get S_A and S_B success respectively. These number of successes depend on the unknown success probabilities p_A and p_B and we determine that $p_A - p_B > \delta$ if $S_A - S_B$ is at least $3N\delta/4$ and otherwise we determine that $p_A - p_B < \delta/2$. In the case that $p_A > p_B + \delta$ the Chernoff-Hoeffding Theorem with inequality (7) shows that $S_A - S_B$ is at least $3N\delta/4$ with high probability.

To more exactly bound the probability that $S_A - S_B > 3N\delta/4$, we first see that S_A is at least $N(p_A - \delta/8)$ except for with probability at most

$$\exp\left(-\frac{1}{4}\left(\frac{\delta}{8p_A}\right)^2 p_A N\right) = \exp\left(-2^{-8}\frac{\delta}{p_A}\delta N\right) \leq \exp\left(-2^{-8}\frac{N}{c}\delta\right)$$

as $p_A \leq c\delta$. Furthermore, with $p_A \geq p_B + \delta$ we have that S_B is less than $\left(p_B + \frac{1}{8}\delta\right)N \leq \left(p_A - \frac{7}{8}\delta\right)N$ except for with probability at most

$$\exp\left(-\frac{1}{4}\left(\frac{\delta}{8p_B}\right)^2 p_B N\right) \leq \exp\left(-2^{-8}\cdot\delta\cdot\frac{\delta}{p_A-\delta}N\right) \leq \exp\left(-2^{-8}\frac{N}{c-1}\delta\right)$$

using $p_B \leq p_A - \delta \leq (c-1)\delta$, assuming $p_B > \delta/4$ so that inequality (7) is applicable. If instead $p_B < \delta/4$ we use inequality (6) to see that the probability of $S_B > \left(p_B + \frac{1}{8}\delta\right)N$ is at most

$$\exp\left(-\frac{\left(\frac{\delta}{8}\right)^2}{p_B + \frac{\delta}{8}}N\right) \leq \exp\left(-2^{-6}\frac{\delta^2}{\frac{3\delta}{8}}N\right) = \exp\left(-2^{-3}\delta N/3\right)$$

which is smaller than $\exp\left(-2^{-8}\frac{N}{c-1}\delta\right)$. As such we can use the same bound when $p_B < \delta/4$ and we have $S_A - S_B > 3N\delta/4$ except for with a probability of at most

$$\exp\left(-2^{-8}\frac{N}{c-1}\delta\right) + \exp\left(-2^{-8}\frac{N}{c}\delta\right) \leq 2\cdot\exp\left(-2^{-8}\frac{N}{c}\delta\right)$$

if $p_A - p_B > \delta$.

In the second case we have $p_A - p_B < \delta/2$ and we want to bound the probability of getting the incorrect answer. To this end, we note that the probability of getting more than $\left(p_A + \frac{1}{8}\delta\right) N$ successes for A to be at most

$$\exp\left(-\frac{1}{4}\left(\frac{\delta}{8p_A}\right)^2 p_A N\right) = \exp\left(-2^{-8}\frac{\delta}{p_A}\delta N\right) \leq \exp\left(-2^{-8}\frac{N}{c}\delta\right) \ .$$

Meanwhile, the probability of getting less than $(p_B - \frac{1}{8}\delta)N$ successes for B is at most

$$\exp\left(-\frac{1}{4}\left(\frac{\delta}{8p_B}\right)^2 p_B N\right) = \exp\left(-2^{-8}\frac{\delta}{p_B}N\right) \leq \exp\left(-2^{-8}\frac{N}{c}\delta\right)$$

by using $p_B \leq p_A \leq c\delta$. As such, if $p_A - p_B < \delta/2$ we have $S_A - S_B < 3N\delta/4$ except for with a probability of at most

$$\exp\left(-2^{-8}\frac{N}{c}\delta\right) + \exp\left(-2^{-8}\frac{N}{c}\delta\right) \leq 2 \cdot \exp\left(-2^{-8}\frac{N}{c}\delta\right)$$

giving that the total probability of answering incorrectly is at most

$$\max\left(2 \cdot \exp\left(-2^{-8}\frac{N}{c}\delta\right), 2 \cdot \exp\left(-2^{-8}\frac{N}{c}\delta\right)\right) = 2 \cdot \exp\left(-2^{-8}\frac{N}{c}\delta\right)$$

and thus with $N = 2^8 k\frac{c}{\delta}$ the error probability is at most

$$2 \cdot \exp\left(-k\right)) \leq 2^{-k}$$

with inequality holding for $k \geq 3$. □

With this lemma, we now finally present the actual proof of Theorem 3.3.

Proof of Theorem 3.3. Let a sample from B_j^I for $0 \leq j < 2\overline{n}$ be generated by taking a random message m, encrypting it with H_j^I and letting $\mathcal{A}$ attempt to decrypt it, with the sample being 0 unless $\mathcal{A}$ correctly recovers the message m, in which case it is 1. As such, samples from B_j^I are Bernoulli distributed with unknown success probability p_j^I.

The advantage of the adversary against the actual PKE is at least ε_a, corresponding to a success probability of at least $\varepsilon_a + 2^{-\ell}$ and thus $p_0^I \geq \varepsilon_a + 2^{-\ell} > \varepsilon_a$. We also know that $p_{2\overline{n}}^J = 2^{-\ell}$ as no strategy in decrypting random messages encrypted with $H_{2\overline{n}}^J$ performs better than random guessing, independent of J.

The remaining success probabilities p_j^J are unknown. However, we can relate the probabilities $p_j^{I'}$ and p_j^I by using the fact that only a single sample from J

is used for encryptions with H_j^J for $j < 2\overline{n}$. As $|I| / |I'| = \kappa$, this implies that $p_j^{I'} \geq \kappa p_j^I$. We want to use $\mathcal{A}$ in order to distinguish between $H_j^{I'}$ and H_{j+1}^I for some j. As I' is unknown, we can not directly encrypt with $H_j^{I'}$ and we therefore instead find a j such that $\mathcal{A}$ can be used to distinguish between H_j^I and H_{j+1}^I with sufficiently large advantage.

We claim that for $\Delta = \frac{\varepsilon_a \kappa^{2\overline{n}}}{4\overline{n}}$, it is guaranteed to be a j such that

$$p_j^{I'} - p_{j+1}^I \geq \kappa p_j^I - p_{j+1}^I \geq \Delta \ . \tag{8}$$

This is seen by noting that the second inequality in (8) equivalently says

$$p_j^I \geq (p_{j+1}^I + \Delta)/\kappa \ .$$

If there is no j where this inequality holds, it would imply

$$p_0^I < \frac{p_{2\overline{n}}}{\kappa^{2\overline{n}}} + \Delta \cdot \sum_{j=1}^{2\overline{n}} \kappa^{-j} \leq 2^{-\ell}\kappa^{-2\overline{n}} + 2\overline{n}\kappa^{-2\overline{n}} \cdot \frac{\varepsilon_a \kappa^{2\overline{n}}}{4\overline{n}} \leq 2^{-\ell}\kappa^{-2\overline{n}} + \varepsilon_a/2 \leq \varepsilon_a$$

with the final inequality given by the assumption that $\varepsilon_a \kappa^{2\overline{n}}/2 \geq 2^{-\ell}$. This contradicts $p_0^I > \varepsilon_a$, showing that (8) must hold for some j.

We now find a j such that $\kappa p_j^I - p_{j+1}^I > \Delta/2$. To accomplish this, we define κB_j^I as a distribution that with probability κ is a sample from B_j^I and otherwise is 0. Thus a sample from κB_{j+1}^I is non-zero with probability κp_{j+1}^I. For $j = 0$ to $j = 2\overline{n} - 1$, we test κB_j^I against B_{j+1}^I by using Lemma 3.5 with $\delta = \Delta$. As it is guaranteed to be some j where

$$\kappa p_j^I - p_{j+1}^I \geq \Delta \ , \tag{9}$$

comparing these distributions with Lemma 3.5 for all different values of j will give us a j where $\kappa p_j^I - p_{j+1}^I > \Delta/2$, assuming that the lemma never fails.

The tests are performed sequentially with j increasing from 0 to $2\overline{n} - 1$, aborting the process with the first j where the difference in success probability is found to be sufficiently large. The failure probability of Lemma 3.5 is at most 2^{-k} and, for the at most $2\overline{n}$ different j used, the probability that there is some failure is thus at most $2\overline{n} \cdot 2^{-k}$.

The preconditions for Lemma 3.5 are fulfilled for the relevant j, unless the lemma has already failed for a smaller value of j. First of all, we see that the condition $\Delta < \kappa p_j^I$ must be fulfilled unless there is some $i < j$ such that $\kappa p_i^I - p_{i+1}^I > \Delta$. As this i should have already been found by Lemma 3.5 we can assume that this precondition is true.

Furthermore, with $c = 2\varepsilon_a/\Delta$, we can assume that $\kappa p_j^I \leq c\Delta$. Otherwise, there is some j such that $p_j^I \geq 2\varepsilon_a/\kappa > \varepsilon_a + 2^{-\ell}$ which would allow a more

efficient way to distinguish between the hybrid games by only considering the versions with $i \geq j$.

The same samples from B_j^I can be used when κB_{j-1}^I is compared to B_j^I as when κB_j^I is compared to B_{j+1}^I and thus it is sufficient with $2^9 \varepsilon_a k/\Delta^2$ samples from each distribution. This results in a total requirement of

$$2\overline{n} \cdot 2^9 \varepsilon_a \cdot \frac{k}{\Delta^2} = 2^{14} \frac{\overline{n}^3 k}{\varepsilon_a \kappa^{4\overline{n}}}$$

samples from different B_j^I and each such sample corresponds to one call to $\mathcal{A}$.

Having found this j, we use Lemma 3.4 with the unknown distribution from the target DLWE(I') instance in order to encrypt random messages. This corresponds to an encryption with H_i^J where $(i, J) = (j, I')$ if the unknown distribution is $A_{\boldsymbol{s},I'}$ and $(i, J) = (j+1, I)$ otherwise. We define a Bernoulli distribution B_i^J for the outcome of an adversary against this cryptosystem in the same way as the other Bernoulli distributions.

Using Lemma 3.5 with $\delta = \Delta/2$, we compare this distribution with B_{j+1}^I. If $(i, J) = (j, I')$ the difference in success probability is at least $\Delta/2$ by the choice of j. As such, the only correct response by the lemma is that the difference in success probability is greater than $\delta/2 = \Delta/4$. If instead $(i, J) = (j+1, I)$ the difference in success probability is 0 and thus the only correct answer from the lemma is that the difference is less than $\delta = \Delta/2$.

The preconditions for Lemma 3.5 are satisfied in the comparison between $B_j^{I'}$ and B_{j+1}^I as they held in the process of finding this j. When B_{j+1}^I is compared to itself, the precondition that $\delta < p_{j+1}^I$ might not hold. The answer is however still correct, as if both distribution have the same low success probability, there is only an insignificant probability that there is a large difference in number of successful trials.

For this application of Lemma 3.5 to have a failure probability of 2^{-k} we require $T = 2^{10} \varepsilon_a \frac{k}{\Delta^2} = 2^{14} \frac{\overline{n}^2 k}{\varepsilon_a \kappa^{4\overline{n}}}$ samples from both distributions. As each sample from either of the distributions require using $\mathcal{A}$, this corresponds to $2T$ calls to $\mathcal{A}$. This also corresponds to T encryptions with samples from the unknown distribution which, because of the usage of Lemma 3.4, implies that we require $3n \cdot T$ samples from the unknown distribution.

Lemma 3.4 has a failure probability of at most $2^{\overline{n}-n}$ per use and thus the probability of at least one failure is no more than $T \cdot 2^{\overline{n}-n}$. Combining this with the probability that Lemma 3.5 fails at least once gives the total failure probability of at most $T \cdot 2^{\overline{n}-n} + (2\overline{n}+1)2^{-k} \leq 2(\overline{n}+1)2^{-k}$ with inequality holding for sufficiently large n. Meanwhile, the number of required calls to $\mathcal{A}$ directly corresponds to the samples from the different Bernoulli distributions B_i^J and is

therefore at most

$$2^{14}\frac{\overline{n}^3 k}{\varepsilon_a \kappa^{4\overline{n}}} + 2^{15}\frac{\overline{n}^2 k}{\varepsilon_a \kappa^{4\overline{n}}} \leq 2^{15}\frac{\overline{n}^3 k}{\varepsilon_a \kappa^{4\overline{n}}}$$

with inequality holding for $\overline{n} \geq 2$. □

4 Discrete Gaussian sampling from Worst-Case DLWE

In this section we detail the reductions which prove that a $DLWE(I, M)$ oracle can be used to solve SIVP_{γ_R}. Combined with Theorem 3.3, this allows the security of our cryptosystem to be based upon the assumed hardness of this standard lattice problem.

In order to solve an SIVP_{γ_R} instance, we consider a version of Regev's quantum reduction from [31] which we detail in subsection 4.2. Our version of this reduction requires an LWE oracle that, with overwhelming probability, solves $\text{LWE}(\Psi_\beta, N)$ for some unknown $\beta \in [\alpha/\sqrt{2}, \alpha]$ with α a known parameter. Such an oracle is not directly given by using an adversary against our cryptosystem with Theorem 3.3 and we require an additional step to construct such an LWE oracle. This oracle is constructed in different ways depending on if we consider a parametrization with fixed or variable error distribution. We begin by detailing these constructions in subsection 4.1.

4.1 Solving LWE with a DLWE Oracle

For our version of the main reduction of [31] we must solve instances of the $\text{LWE}(\Psi_\beta, N)$ for some unknown $\beta \in [\alpha/\sqrt{2}, \alpha]$. This version of the LWE problem is solved in two separate ways depending on which type of DLWE oracle an adversary against the specific parametrization of our cryptosystem provides. For parametrizations with $I = \{\alpha^2\}$ we use the same approach as in [31]. For the parametrizations with $I = [\alpha^2, 3\alpha^2/2]$ we have an alternative, more efficient way, to construct the required LWE oracle.

We first consider our parametrizations that use a fixed error distribution with $I = \{\alpha^2\}$. In this case, using Theorem 3.3 allows using an adversary against our cryptosystem to solve $\text{DLWE}(\Psi_\alpha, N)$. Using Lemma 2.8, this allows us to also solve $\text{LWE}(\Psi_\alpha, N)$. Finally, we use the following lemma in order to actually solve $\text{LWE}(\Psi_\beta, \tau \cdot N)$ for unknown $\beta \in [\alpha/\sqrt{2}, \alpha]$ and the value of τ determining the success probability of the algorithm.

Lemma 4.1 (Variant of Lemma 3.7 from [31])**.** *Let n, q, τ be positive integers, $\alpha \in (0, 1)$ and W be an algorithm that solves $LWE(\Psi_\alpha, N)$, with a failure proba-*

bility of at most $1/20$. *Then there is an algorithm* W' *that, by using* W *at most* $10\tau N$ *times, solves LWE*$(\Psi_\beta, \tau N)$ *for arbitrary* $\beta \in [\alpha/\sqrt{2}, \alpha]$, *except for with probability at most*

$$2^{-\tau} + 10N\tau \exp(-9N/80) \ .$$

Proof. We define a set Z to be all integer multiples of $\alpha^2/(20N)$ between $\alpha^2/2$ and α^2. For each of the elements $\zeta \in Z$, we add a sample from $\Psi_{\sqrt{\zeta}}$ to each of the nN samples provided from $A_{\boldsymbol{s},\Psi_\beta}$. This results in nN samples from $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$ for each $\zeta \in Z$. This guarantees that there is some $\zeta \in Z$ such that

$$\alpha^2 \leq \beta^2 + \zeta \leq \left(1 + \frac{1}{20N}\right)\alpha^2$$

and Lemma 2.1 gives us that, for this ζ, the statistical distance between $\Psi_{\sqrt{\beta^2+\zeta}}$ and Ψ_α is at most $9/(20N)$. As such, the statistical distance between N samples from $A_{\boldsymbol{s},\Psi_\alpha}$ and N samples from $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$ is at most $9/20$ for this choice of ζ.

As the LWE oracle has a failure probability of at most $1/20$, this guarantees that the oracle succeeds in recovering $\boldsymbol{s}$ with success probability at least $1/2$ when given N samples from $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$ instead of N samples from $A_{\boldsymbol{s},\Psi_\alpha}$. Repeating this same procedure τ times, with independent samples, thus ensures that, except for with probability at most $2^{-\tau}$, we will produce the correct solution at least once.

As we perform this procedure for every $\zeta \in Z$, we are thus guaranteed that, except for with probability at most $2^{-\tau}$, the correct $\boldsymbol{s}$ is found at least once. By using the procedure from Lemma A.7, corresponding to Lemma 3.6 of [31], we are able to verify when we have the correct solution. This is performed on at most $\tau \cdot 10N$ candidate solutions and if the lemma at any point incorrectly claims that some $\boldsymbol{s}' \neq \boldsymbol{s}$ equals $\boldsymbol{s}$ we will return an incorrect solution $\boldsymbol{s}'$. We therefore require that this procedure never fails, giving a total failure probability of at most

$$2^{-\tau} + 10\tau N \exp(-9N/80)$$

by using N samples from $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$ that were not used to find the candidate solution. Finally, the number of required calls to W is $10\tau N$ as there is $10N$ different $\gamma \in Z$ and each γ requires τ calls to W. □

Compared to previous analysis of the efficiency of Lemma 3.7 in [31], our version in Lemma 4.1 requires a factor $N/10$ less LWE oracle calls and as N is large, this is a significant difference. This is accomplished by choosing the set Z more carefully compared to how it was chosen in Regev's original proof [31].

For our alternative parametrization, where $I = [\alpha^2, 3\alpha^2/2]$, Theorem 3.3 allows an adversary to be used to solve an DLWE(I', N) instance for some I'

that contains I. With $\beta \in [\alpha/\sqrt{2}, \alpha]$ we are guaranteed that $I' = [\beta^2, \beta^2 + \alpha^2]$ contains I and that $|I'| / |I| = 2$. This motivates the following lemma which allows us to solve DLWE(Ψ_β, N), even though β is unknown. Using Lemma 2.8 this also allows us to solve the corresponding search-LWE problem.

Lemma 4.2. *Let α, β be two positive numbers and let $I' = [\beta^2, \beta^2 + \alpha^2]$. Then, an instance of the DLWE(Ψ_β, N) problem can be transformed into an instance of the DLWE(I', N) problem without knowing β.*

Proof. For each new distribution requested in the DLWE(I', N) instance, sample a new ζ uniformly at random from $[0, \alpha^2]$. Samples from this new distribution are produced by sampling $(\boldsymbol{a}, b)$ from the DLWE(Ψ_β, N) instance and returning $(\boldsymbol{a}, b + e \mod 1)$ where e is sampled from $\Psi_{\sqrt{\zeta}}$.

If the input distribution was uniformly random, then so is the output distribution. If instead $b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e'$ for some $\boldsymbol{s}$ and with $e' \leftarrow \Psi_\beta$, we see that the sum $e + e' \mod 1$ is distributed as $\Psi_{\sqrt{\beta^2+\zeta}}$. Therefore, the resulting distribution is $A_{\boldsymbol{s},\Psi_{\sqrt{\beta^2+\zeta}}}$. As $\zeta \leftarrow \mathcal{U}([0, \alpha^2])$, this corresponds exactly to $A_{\boldsymbol{s},\Psi_{\sqrt{\zeta'}}}$ with $\zeta' \leftarrow \mathcal{U}([\beta^2, \beta^2 + \alpha^2])$ which is the expected distribution for an DLWE(I', N) instance. □

4.2 Solving SIVP with the Help of an LWE Oracle

By using a version of Regev's quantum reduction from [31], we are able to solve SIVP with the help of an LWE oracle. The efficiency of Regev's reduction has already been investigated in previous works [9,14,33] with the most recent result concluding that $2n^3 \cdot 3n^3 N^3$ LWE oracle calls are required when using an LWE oracle that requires N LWE samples.

Previous works that analyzed the efficiency of this reduction have not considered its failure probability. Furthermore, they have analyzed the reduction exactly as stated by Regev and have not altered it in order to improve its concrete efficiency. Our version of this reduction has somewhat improved efficiency compared to the original and we keep track of its concrete failure probability.

To allow as tight proof as possible, we somewhat alter some of the steps of Regev's reduction. However, many of the lemmas from [31] are used essentially as is, without major modifications. As the original proofs did not concretely handle tightness, the proofs must still be redone in greater detail. In this section we only present the main part of the reduction, which has a large impact on the non-tightness of the reduction and has been altered somewhat to improve its efficiency. For most other lemmas, the analysis of their tightness is similar to the analysis in previous works that analyzed this reductions tightness. We therefore

do not present these lemmas here, instead including them in Appendix A for completeness.

To solve an SIVP_γ instance, Regev's reduction from [31] constructs a discrete Gaussian distribution over the lattice with a relatively small standard deviation. The length of vectors sampled from this distribution are related to its standard deviation and vectors sampled from this distribution are therefore relatively short. With sufficiently many sampled vectors, there is an overwhelming probability that a subset of n sampled vectors are linearly independent. These vectors thus solve a SIVP_γ instance for some γ related to the standard deviation of the discrete Gaussian distribution.

To produce samples from a discrete Gaussian distribution, the reduction uses multiple iterative steps. Each iterative step makes use of samples from a discrete Gaussian distribution in order to produce samples from a discrete Gaussian distribution with smaller standard deviation. This process can be repeated with the same input samples in order to produce an arbitrary number of samples from the output discrete Gaussian distribution. As such, the process can be applied iteratively in order to sample from discrete Gaussian distributions with successively smaller standard deviations.

The iterative step can be considered as two separate parts. One part of the iterative step is a classical algorithm that solves a BDD problem on the dual of the lattice. This is accomplished by using the provided LWE oracle together with samples of a discrete Gaussian distribution over the lattice. The other part of the reduction consists of a quantum algorithm that produces samples from a discrete Gaussian distribution. This requires solving a BDD problem on the dual lattice which is accomplished by using the first part of the reduction.

The classical part of the algorithm requires one input discrete Gaussian sample for each of the LWE samples required by the LWE oracle. With the LWE oracle requiring N LWE samples, each iterative step must therefore be repeated at least N times in order to provide enough input samples for the next iterative step. As the number of input samples N that the LWE oracle constructed by Theorem 3.3 is quite large, this repetition to produce enough samples is one of the largest contributors to the non-tightness of the reduction.

In order to improve the efficiency of the reduction, our version of this theorem solves SIVP_{γ_R} directly. The original reduction solves the target SIVP_{γ_R} instance by solving multiple discrete Gaussian sampling problems, as detailed in Lemma 3.17 of [31]. We instead use that the solution to the discrete Gaussian sampling problem already samples from many intermediate discrete Gaussian distributions. By using these samples, we are able to solve SIVP_{γ_R} directly, saving a factor $2n^3$ in the number of required oracle calls compared to using Lemma 3.17 from [31].

Theorem 4.3 (Version of Theorem 3.1 in [31])**.** *Let L be an arbitrary lattice,*

n, q, M be positive integers, $\alpha < 1$ some positive number such that $\alpha q > 2\sqrt{n}$ and let $\varepsilon = \varepsilon(n) > 0$. Furthermore, let W be an oracle that solves $LWE(\Psi_\beta, M)$ for arbitrary $\beta \in [\alpha/\sqrt{2}, \alpha]$, except for with probability at most ε. Then there is a quantum algorithm that, by using W at most $3n^2M$ times, solves $GIVP_{q\sqrt{2n}\cdot\eta_\varepsilon(L)}$ except for with probability

$$3nM \cdot \sqrt{18Mn\varepsilon} + 10Mn^2 \cdot 2^{-n/2} \ .$$

Proof. We begin by using LLL on the input n-dimensional lattice L. This gives us a basis for L where the length of the longest vector is $\tilde{\lambda}_n(L)$ and where it is known that

$$\lambda_n(L) \leq \tilde{\lambda}_n(L) \leq 2^n \lambda_n(L) \ .$$

We let $r = r_0 = \tilde{\lambda}_n(L)2^{-n} \leq \lambda_n(L)$ and $r_i = r \cdot (\alpha q/\sqrt{n})^i$. As $\alpha q > 2\sqrt{n}$ implies $r_{3n} > 2^{3n} r > 2^{2n}\lambda_n(L)$, we can use Lemma A.4 to efficiently produce samples from a distribution that is statistically close to $D_{L,r_{3n}}$. Starting with samples from this distribution, the iterative step given by Lemma A.13 is used to produce samples that are statistically close to samples from another discrete Gaussian distribution.

The initial samples from Lemma A.4 have a width $r_{3n} > 2^{2n}\lambda_n$ and are of statistical distance less than $2^{-n/2}$ from discrete Gaussian samples on the lattice. We use M samples from this distribution and the total statistical distance of these samples from their desired distribution is thus no more than $M \cdot 2^{-n/2}$. With these M samples, the iterative step is used M times in order to produce M samples from a distribution that is statistically close to $D_{L,r_{3n-1}}$. These samples are then similarly used to produce samples from $D_{L,r_{3n-2}}$ and the process continues similarly until samples corresponding to D_{L,r_i} are produced for all relevant i.

We have that $r_i \leq \sqrt{2}q\eta_\varepsilon(L)$ for sufficiently small i. This implies that r_i is to small for samples from D_{L,r_i} to be useable as input to the iterative step of Lemma A.13. However, $\eta_\varepsilon(L)$ is unknown and we therefore have no efficient way to verify if r_i is large enough for produced samples to be used as input to Lemma A.13. Furthermore, it may not necessarily be easy to detect that the lemma fails to produce the expected output. Therefore, we always continue the process until we produce samples that would correspond to D_{L,r_0} if every step succeeds.

With $r_i < \sqrt{2}q\eta_\varepsilon(L)$ there is no guarantee that the produced samples follow a distribution that is even remotely close to $D_{L,r_{i-1}}$. However, the smallest lattice vectors produced by this iterative process must be at least as short as vectors provided by a distribution close to D_{L,r_j} where r_j is big enough for Lemma A.13 to be applicable. Because of this, taking the shortest set of n linearly independent vectors produced by the iterative step for all r_j gives the desired solution.

To calculate the approximation factor for this reduction, we note that r_0 is too small for Lemma A.13 to be applicable while r_{3n} is guaranteed to be large enough that it is applicable. There must therefore be some largest $j \geq 0$ such that $r_j < \sqrt{2}q\eta_\varepsilon(L)$, implying that $r_{j+1} \geq \sqrt{2}q\eta_\varepsilon(L)$. As such, Lemma A.13 can use input samples from $D_{L,r_{j+1}}$ to produce samples statistically close to D_{L,r_j}.

Corollary A.15 shows that M vectors produced from D_{L,r_j} contains a set of n linearly independent vectors except for with probability at most $n(9/10)^{M/n}$. Furthermore, by Lemma 2.2 a vector produced from D_{L,r_j} is of length less than $\sqrt{n}r_j < q\sqrt{2n} \cdot \eta_\varepsilon(L)$ except for with probability

$$\frac{\rho_{r_j}(L \backslash \sqrt{n}r_j B_n)}{\rho_{r_j}(L)} < 2^{-2n} \quad .$$

As such, with high probability, the procedure produces n vectors shorter than $q\sqrt{2n} \cdot \eta_\varepsilon(L)$ and thus solve $\mathrm{GIVP}_{q\sqrt{2n}\cdot\eta_\varepsilon(L)}$.

In total this process uses $3Mn$ repetitions of the iterative step, which includes producing M samples from the distributions corresponding to D_{L,r_i} for all $0 \leq i \leq 3n$. As each iterative step requires n calls to W, this full process requires $3Mn^2$ calls to W. Furthermore, if given the expected input, each iterative step results in samples that are at most a statistical distance of

$$\sqrt{18Mn\varepsilon} + 3n \cdot 2^{-n/2}$$

from the desired distribution. Thus, M samples produced by Lemma A.13 have a statistical distance of at most

$$\sqrt{18M^3n\varepsilon} + 3Mn \cdot 2^{-n/2}$$

from M samples from the desired distribution. As at most $3n$ steps are performed the total statistical distance between M samples at step j and M samples from D_{L,r_j} is at most

$$9\left(\sqrt{2M^3n^3\varepsilon} + Mn^2 \cdot 2^{-n/2}\right) \quad . \tag{10}$$

This assumes that r_j is large enough for Lemma A.13 to produce samples from a distribution close to D_{L,r_j} and also assumes that the initial distribution was $D_{L,r_{3n}}$. As the initial samples are a statistical distance $M \cdot 2^{-n/2}$ from the expected input, the actual statistical distance between the samples from the produced distribution and samples from D_{L,r_j} is bounded by (10) plus $M \cdot 2^{-n/2}$.

For the failure probability of this procedure, we must account for the statistical distance from the desired distribution, the probability that all elements from this distribution are shorter than $\sqrt{2n}q\eta_\varepsilon(L)$ and the probability that these vectors contain n linearly independent vectors. This results in a total failure

probability of at most

$$9Mn\left(\sqrt{2Mn\varepsilon}+n\cdot 2^{-n/2}\right)+M2^{-n/2}+M2^{-2n}+n(9/10)^{M/n}$$
$$\leq 9\sqrt{2M^3n^3\varepsilon}+10Mn^2\cdot 2^{-n/2}$$

with the final inequality holding for all relevant parameters. □

5 Security from lattices

5.1 Cryptosystem with fixed error distribution

The concrete OW-CPA security of our cryptosystem with fixed error distribution is given by a combination of Theorems 3.3 and 4.3. This is detailed in the following theorem, which directly relates the concrete security of our cryptosystem with the hardness of worst-case SIVP.

Theorem 5.1. *Let τ, k be arbitrary integers and let $\varepsilon > 0$ be a real number. Furthermore, let our PKE scheme be parametrized by n, $\overline{n}$, B, q and with $I = \{\alpha^2\}$ such that $\alpha q > 2\sqrt{n}$. Assume that $\mathcal{A}$ is an adversary that, running in time at most T, achieves an advantage of at least 2^{-d} against the OW-CPA security of this parametrization. Then there exists a quantum algorithm that solves worst-case SIVP$_\gamma$ in time*

$$270\tau^2k^3n^5\overline{n}^7q\cdot 2^{3d+43}\cdot T$$

with approximation factor

$$\gamma = q\cdot\sqrt{\frac{2n\cdot\ln(2n(1+1/\varepsilon))}{\pi}}\ .$$

This algorithm has a failure probability of at most

$$2^{3d/2+21}\cdot\tau(3n\overline{n})^3\cdot\sqrt{6k^3\varepsilon}+30\tau kn^3\overline{n}^2\cdot 2^{d+14-n/2}$$

as long as $2^{-\tau} < 3n\overline{n}^2k\cdot 2^{d+14}\varepsilon$, $k \geq \log(40nq(\overline{n}+1))$, $2^{-\ell} < 2^{-d-1}$.

Proof. This is a combination of Theorems 3.3 and 4.3 with Lemmas 4.1, 2.7 and 2.8, combining the number of calls required. Furthermore, in this case $I = I' = \{\alpha^2\}$ which allows Theorem 3.3 to be used with $\kappa = 1$.

The number of required calls to the adversary is at most

$$3Mn\cdot 10\tau N\cdot 2^{d+15}k\overline{n}^3\cdot nq = 270\tau^2k^3n^5\overline{n}^7q\cdot 2^{3d+43}$$

where $M = \tau N$ is given by Lemma 4.1 while $N = 3n\overline{n}^2 k \cdot 2^{d+14}$ is given by Theorem 3.3 with $\kappa = 1$. The running time is then given from this combined number of calls and the fact that each call to $\mathcal{A}$ takes time at most time T.

Theorem 4.3 produces vectors of length at most $q\sqrt{2n} \cdot \eta_{\varepsilon(L)}$. By Lemma 2.5 we see that the length of these vectors is at most

$$q \cdot \sqrt{\frac{2n \ln(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n$$

and the process therefore solves SIVP_γ for $\gamma = q \cdot \sqrt{\frac{2n \ln(2n(1+1/\varepsilon))}{\pi}}$.

The failure probability of this procedure is given by Theorem 4.3 to be at most

$$\begin{aligned} &9\tau N n\sqrt{2Nn\varepsilon} + 10\tau N n^2 \cdot 2^{-n/2} \\ &\quad = 2^{3d/2+21} \cdot \tau(3n\overline{n})^3 \cdot \sqrt{6k^3\varepsilon} + 30\tau k n^3 \overline{n}^2 \cdot 2^{d+14-n/2} \end{aligned}$$

as long as the provided LWE oracle has a failure probability of at most $1/20$. As we are using the LWE oracle created by combining Theorem 3.3 and Lemma 2.8 it has an error probability of at most $2nq(\overline{n}+1)2^{-k}$ and we therefore require $k \geq \log(40nq(\overline{n}+1))$.

Due to Theorem 4.3, we also require that the failure probability of oracle produced by Lemma 4.1 is at most $N\varepsilon$, which corresponds to

$$2^{-\tau} < N\varepsilon = 3n\overline{n}^2 k \cdot 2^{d+14}\varepsilon \ .$$

□

5.2 Cryptosystem with variable error distribution

For the cryptosystem with variable error distribution we have the following, more efficient, version of Theorem 5.1. This improved efficiency is possible as an adversary against this cryptosystem can be used to solve an LWE problem with a variable error distribution. Together with Lemma 4.2, this allows solving LWE instances where the error distribution is unknown. The resulting LWE oracle is significantly more efficient than the LWE oracle given by Theorem 5.1.

Theorem 5.2 (Alternative version of Theorem 5.1)**.** *Let our PKE scheme be parametrized by n, $\overline{n}$, B, q and with $I = [\alpha^2, 1.5\alpha^2)$ such that $\alpha q > 2\sqrt{n}$. Furthermore, let k be an arbitrary integer and $\varepsilon > 0$ be some real number. Assume that $\mathcal{A}$ is an adversary that runs in time at most T and achieves an advantage at least 2^{-d} against the OW-CPA security of this parametrization of*

our PKE scheme. Then, there exists a quantum algorithm that solves worst-case SIVP$_\gamma$ in time

$$9k^2n^4\overline{n}^5q \cdot 2^{2d+29+4\overline{n}} \cdot T$$

for approximation factor

$$\gamma = q\sqrt{\frac{2n \cdot \ln(2n(1+1/\varepsilon))}{\pi}}$$

with a failure probability of at most

$$2^{3d/2+6\overline{n}+21} \cdot (3n\overline{n})^3\sqrt{6k^3\varepsilon} + 30n^3\overline{n}^2k \cdot 2^{d+14+4\overline{n}-n/2}$$

as long as $3k\overline{n}^2 2^{d+14} > 4n$, $2^{-\ell} < 2^{-d-2\overline{n}-1}$ *and*

$$k \cdot 2^k \geq 2^{-d-13-4\overline{n}} \cdot \frac{q(\overline{n}+1)}{3\overline{n}^2\varepsilon} \quad .$$

Proof. Using Theorem 3.3, an adversary against the OW-CPA security of our PKE scheme can be used to solve the DLWE(I', N) problem for arbitrary I' that contains I. Using Lemmas 2.8, 4.2 and 2.7 this allows solving the LWE(Ψ_β, N) problem for any $\beta \in [\alpha/\sqrt{2}, \alpha]$. Finally, using Theorem 4.3 provides a solution to an arbitrary GIVP$_{q\sqrt{2n}\cdot\eta_\varepsilon(L)}$ instance.

With $\beta^2 \in [\alpha^2/2, \alpha^2]$ and $I' = [\beta^2, \beta^2 + \alpha^2]$ we have that I' contains I and $|I| \,/\, |I'| = 1/2$. As such we have $\kappa = 1/2$ in Theorem 3.3 which means that the number of required calls to $\mathcal{A}$ for every use of Theorem 3.3 is $2^{d+15+4\overline{n}}\overline{n}^3k$. The total number of required calls to solve this GIVP instance is

$$3Nn^2 \cdot k\overline{n}^3 \cdot 2^{d+15+4\overline{n}} \cdot nq = 9k^2n^4\overline{n}^5q \cdot 2^{2d+29+4\overline{n}}$$

where $N = 3n\overline{n}^2k \cdot 2^{d+14+4\overline{n}}$ is given by Theorem 3.3.

For Theorem 4.3 to be applicable, the failure probability of the LWE oracle can be at most $N\varepsilon$. As the LWE oracle is given by a combination of Theorem 3.3 and Lemmas 2.8 and 2.7 it has a failure probability of at most $2nq \cdot (\overline{n}+1) \cdot 2^{-k}$ and to have this smaller than εN corresponds to

$$k \cdot 2^k \geq 2^{-d-13-4\overline{n}} \cdot \frac{q(\overline{n}+1)}{3\overline{n}^2\varepsilon}$$

with value of N inserted. With this inequality holding the total failure probability of the reduction is no more than

$$\begin{aligned} &9nN \cdot \sqrt{2Nn\varepsilon} + 10Nn^2 \cdot 2^{-n/2} = \\ &\quad 2^{3d/2+6\overline{n}+21} \cdot (3n\overline{n})^3\sqrt{6k^3\varepsilon} + 30n^3\overline{n}^2k \cdot 2^{d+14+4\overline{n}-n/2} \end{aligned}$$

as given by Theorem 4.3. Finally, the analysis for SIVP approximation factor is identical to the one given in Theorem 5.1 and therefore not repeated here. □

6 Parametrization

In this section we parametrize our cryptosystem while accounting for the reductions efficiency. Besides its running time, we must also account for the failure probability of the reduction, and how we do this is detailed in subsection 6.1. The security must also be based on a reasonable estimate for the concrete hardness of the underlying problem, which we detail in subsection 6.2.

The primary parametrizations in this paper is for a PKE that targets 128-bits of OW-CPA security. We also provide separate parametrizations of a KEM that target 128-bits of IND-CCA security. These parametrizations are based on the $\text{FO}^{\not\perp}$ transform, which introduces additional parameter constraints and another non-tight reduction. Details about the $\text{FO}^{\not\perp}$ transform and these IND-CCA secure parametrizations are provided in subsection 6.3

Next, in subsection 6.4 it is detailed how the concrete parameters are selected to target 128-bits of security by using the reductions presented in this paper.

Our concrete parametrizations are based on conservative estimates of the concrete hardness of worst-case SIVP. There does not seem to be anything that indicates that the worst-case problem is significantly harder than the average-case version of the problem. As such, there does not seem to be to be any reasonable way to argue that the worst-case nature of the underlying problem contributes to the concrete hardness of this problem and the concrete security of our cryptosystem. We therefore base our parametrization on a hardness estimate for average-case SIVP.

As a secondary result, we also want to provide an indication as to what extent typical LWE-based cryptosystems qualitatively can argue for security from the reduction. Our concrete parametrizations provide a first indication for this. However, these do not account for the worst-case nature of the underlying problem as we do not deem it reasonable to account for this when basing the security on the reduction. However, the security of cryptosystems that use smaller parameters could still be argued to be partially supported by the reduction based on the fact that it solves a worst-case problem.

As there are no known instances of SIVP_γ that are significantly harder than typical instances of the problem, such an argument can not reasonably be used for a concrete parametrization of our cryptosystem. Furthermore, assuming the existence of hard problem instances is not not an easily falsifiable assumption. Even if lattice algorithms improve against some class of lattices, we could still claim that there exist some other, harder class of lattices. However, in order to bound which parameters the to reduction could provide any support for, we still investigate what parameters are supported by such an argument.

Our bound on parameters supported by the reduction is such that attacks

against cryptosystems that use smaller parameters would not even result in improved algorithms against a hypothetical worst-case instance of SIVP_γ. This worst-case instance may not be efficiently solvable with the heuristic algorithms typically considered. However, using a less efficient, but provably correct lattice algorithm, we should be able to solve an arbitrary instance of a lattice problem. Thus, we provide such a parameter bound by comparing the reduction to the efficiency of provably correct lattice algorithms, which we detail in subsection 6.5.

6.1 For low reduction failure probability

The failure probability of the reduction in Theorem 5.1 is given by

$$2^{3d/2+21} \cdot \tau(3n\overline{n})^3 \cdot \sqrt{6k^3\varepsilon} + 30\tau k n^3 \overline{n}^2 \cdot 2^{d+14-n/2}$$

while it for the reduction in Theorem 5.2 is

$$2^{3d/2+6\overline{n}+21} \cdot (3n\overline{n})^3 \sqrt{6k^3\varepsilon} + 30 n^3 \overline{n}^2 k \cdot 2^{d+14+4\overline{n}-n/2} \quad .$$

To limit these failure probabilities, we choose a sufficiently small value for ε. Both the running time of the reductions and the approximation factor the reductions achieve increase with smaller values of ε. Therefore, ε is chosen as large as possible while still achieving an acceptable reduction failure probability.

The concrete failure probability of the reduction is accounted for in the same way as in the calculation of the tightness gap. As such, if the reduction running in time T achieves a success probability p it is considered equivalent to an alternative reduction running in time T/p that never fails. With this way of accounting for the failure probability of the reduction, there should be some optimal choice for ε that achieves as efficient parametrization as possible. However, altering this variable only has a minor impact on the final parametrization and we therefore accept a suboptimal choice for this parameters.

For simplicity we target a total reduction failure probability of approximately 1/10. To achieve this, we ignore the contribution of the second term of the failure probability, as it is exponentially small in n, and is insignificant for relevant values of n. We therefore select

$$\varepsilon = \frac{1}{600k^3} \cdot \frac{2^{-3d-42}}{\tau^2(3n\overline{n})^6}$$

in the system with fixed error distribution and

$$\varepsilon = \frac{1}{600k^3} \cdot \frac{2^{-3d-12\overline{n}-42}}{(3n\overline{n})^6}$$

in the system with variable error distribution. We also have requirements on k and τ from the respective theorems, and we select these parameters as small as possible while these inequalities still hold.

As such, in the analysis of the cryptosystem with fixed error distribution we select k to be the smallest integer greater than $\log(40nq(\overline{n}+1))$ while τ is the smallest integer such that

$$2^{-\tau} < 3n\overline{n}^2 k \cdot 2^{d+14}\varepsilon = \frac{1}{600k^2} \cdot \frac{\overline{n} \cdot 2^{-2d-28}}{\tau^2(3n\overline{n})^5}$$

which corresponds to

$$\tau^{-2}2^{\tau} > 600k^2 \cdot 2^{2d+28} \cdot (3n)^5\overline{n}^4 \ .$$

For the system with variable error distribution, we instead select k as the smallest integer such that

$$k \cdot 2^k \geq 2^{-d-13-4\overline{n}} \cdot \frac{q(\overline{n}+1)}{3\overline{n}^2\varepsilon} = 600k^3 \cdot 2^{2d+8\overline{n}+29}3^5n^6\overline{n}^4 \cdot q(\overline{n}+1)$$

which corresponds to

$$k^{-2} \cdot 2^k \geq 600 \cdot 2^{2d+8\overline{n}+29}3^5n^6\overline{n}^4 \cdot q(\overline{n}+1)$$

while τ is not a parameter for the reduction.

6.2 Hardness estimate

There seems to have been barely any research into the concrete hardness of SIVP with the approximation factors relevant for the theorems in Section 5. Meanwhile, the concrete hardness of Hermite-SVP, which is a similar problem, has been studied extensively. Both problems are solved by finding short vectors in a lattice. The vectors required to solve SIVP_γ must be shorter than $\gamma \cdot \lambda_n(L)$ while the solution to a η-Hermite-SVP instance is a vector that is shorter than $\eta \cdot \det(L)^{1/n}$.

In order to relate solutions of SIVP_γ and solutions to η-Hermite-SVP we want to relate the different approximation factors. However, in general it is not possible to bound $\lambda_n(L)$ in terms of $\det(L)^{1/n}$. This can for example be seen in the lattice generated by the basis

$$\begin{bmatrix} k & 0 \\ 0 & \frac{1}{k} \end{bmatrix}$$

which has determinant 1 but $\lambda_2 = k$ for arbitrary $k \geq 1$. However, on a random lattice the Gaussian heuristic predicts the length of the shortest vectors in terms of the determinant. This allows the following lemma to bound λ_n in terms of the determinant of the lattice, showing that on a random lattice, a solution to SIVP_γ also implies a solution to $(\sqrt{2n\pi e} \cdot \gamma)$-Hermite-SVP.

Lemma 6.1. *Given a lattice L such that the Gaussian heuristic holds on L^*, we have $\lambda_n \leq \sqrt{2n\pi e} \cdot \det(L)^{1/n}$.*

Proof. The lemma is directly given by combining that $\lambda_1^* \lambda_n \leq n$, which is a transference theorem from [6], and the Gaussian heuristic for the length of the first minima on the dual lattice. □

The concrete hardness of η-Hermite-SVP is estimated from the efficiency of currently known algorithms that solve this problem. In practice, the most efficient algorithms that solve Hermite-SVP are variants of the BKZ lattice reduction algorithm [34]. Assuming that this is the most efficient algorithm that solves Hermite-SVP on random lattices also bounds the performance of algorithms that solve SIVP_γ.

Our parametrizations are based on a conjectured optimal performance of algorithms that solves SIVP_γ, as detailed in Conjecture 1. The conjecture follows from the assumed optimal performance of BKZ in solving Hermite-SVP. With Lemma 6.1 this also limits the performance of algorithms that solve SIVP_γ on random lattices. As the reduction is a quantum algorithm, we consider the quantum performance of BKZ. The potential failure probability of the algorithm is also accounted for in the conjecture in the same way as in the tightness gap definition from [9].

Conjecture 1 (Concrete hardness of approximate SIVP)**.** *There is no quantum algorithm that, with probability ε_C, solves SIVP_{γ_C} on random lattices in time T_C such that $T_C/\varepsilon_C < 2^{0.2563\beta}$ for approximation factor*

$$\gamma_C = \left((\pi\beta)^{1/\beta} \frac{\beta}{2\pi e} \right)^{\frac{n}{2(\beta-1)}} \cdot \frac{1}{\sqrt{2n\pi e}} \ .$$

Conjecture 1 does not directly correspond to an algorithm that solves SIVP_{γ_C}. It is therefore possible that SIVP_{γ_C} is significantly harder than assumed by the conjecture. However, it seems likely that an algorithm with similar performance to that claimed optimal by the conjecture can be used to solve SIVP_{γ_C} on random lattices. Running BKZ with a randomized initial lattice basis, solving $(\sqrt{2n\pi e} \cdot \gamma_C)$-Hermite-SVP multiple times, can reasonably be assumed to produce n linearly independent lattice vectors shorter than $\gamma_C \cdot \lambda_n$ after a small polynomial number of repetitions. This algorithm should thus reasonably solve SIVP_{γ_C} on random lattices with performance similar to the conjectured optimal performance. As such, it does not seem like the conjecture significantly underestimates the hardness of SIVP_{γ_C}.

6.3 QROM IND-CCA security

We use the $\text{FO}^{\not\perp}$ transform [18] in order to transform our OW-CPA secure PKE scheme into an IND-CCA secure KEM. The security of this KEM is guaranteed by a non-tight proof in the quantum random oracle model. While the $\text{FO}^{\not\perp}$ transform was used by many of the NIST candidates for post-quantum cryptography, these schemes did not account for the non-tightness of its security proof. This is motivated by the facts that no attacks are able to make use of the non-tightness and that tighter proofs exist in the classical random oracle model. However, the non-tight proof does imply a loss of provable security in the quantum random oracle model and is therefore accounted for in this paper. The concrete loss of provable security is quantified in the following theorem from [19], where KEM-I is the $FO^{\not\perp}$ transform of PKE.

Theorem 6.2 (Theorem 1 from [19])**.** *If* PKE *is δ-correct, for any* IND-CCA $\mathcal{B}$ *against* KEM-I*, issuing at most q_D queries to the decapsulation oracle* Decaps*, at most q_G queries to the random oracle G and at most q_H queries to the random oracle H, there exists a* OW-CPA *adversary $\mathcal{A}$ against* PKE *such that*

$$\text{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq \frac{2q_H}{\sqrt{|M|}} + 4q_G\sqrt{\delta} + 2(q_G + q_H) \cdot \sqrt{\text{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$$

and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.

By using this theorem, we want to argue that a KEM defined as the $FO^{\not\perp}$ transform of a parametrization of our PKE scheme achieves d bits of IND-CCA security. We limit the adversary to 2^Q QROM queries and parametrize our PKE so that it has a message space size of $|M| = 2^\ell \geq 2^{2d+4+2Q}$. Furthermore, we parametrize it to have a decryption failure probability $\delta \leq 2^{-(2d+6+2Q)}$ while the PKE is instantiated to have at least $(2d+4+2Q)$ bits of OW-CPA security. Considering an adversary $\mathcal{B}$ against our KEM that runs in time T with advantage ε_B this parametrization guarantees that $T/\varepsilon_B > 2^d$. This is seen by noting that $\varepsilon_B < 2^{-d}$ directly follows from this parametrization in the worst-case where $T = 1$.

For the concrete parametrizations presented in this paper, we target 128 bits of IND-CCA security and limit the adversary to 2^{128} QROM queries. Therefore, we instantiate our underlying PKE to target 516 bits of OW-CPA security. We also require that the underlying PKE scheme should have a decryption failure probability of $\delta \leq 2^{-518}$ with a message space M that at least has size 2^{516}. Furthermore, Theorems 5.1 and 5.2 require that $1/|M| = 2^{-\ell}$ is sufficiently small. We therefore parametrize the PKE schemes with $B = 4$ and $\overline{n} = 12$, giving $2^\ell = |M| = 2^{576}$. With $\overline{n} = 12 < 2^4$ we target a per-symbol decryption error probability of 2^{-526} by using Lemma 3.1.

6.4 Selected parametrizations

Assuming Conjecture 1, the security of the proposed concrete parametrizations of our cryptosystems are guaranteed by either Theorem 5.1 or Theorem 5.2. All our parametrizations use $B = 4$, encrypting 4 bits in each element of $\boldsymbol{C}_2$. The parametrizations of our PKE scheme that target OW-CPA security use $\overline{n} = 8$ and therefore have a message space of size $2^\ell = 2^{256}$. For our parametrizations of a KEM that targets 128 bits of IND-CCA security by using the $FO^{\not\perp}$ transform, we instead use $\overline{n} = 12$ and therefore have a message space size of $2^\ell = 2^{576}$.

Our parametrizations target a per symbol decryption failure probability of 2^{-64} for the OW-CPA secure PKEs and 2^{-526} for the IND-CCA secure KEMs, as detailed in Section 6.3.

The systems with fixed error distribution use error interval $I = \{\alpha^2\}$ while the variable error distribution systems use $I = [\alpha^2, 3\alpha^2/2]$. In both cases α and q are selected so that q is a prime, $\alpha q > 2\sqrt{n}$ and α is as large as possible, as detailed in Section 3.1. The following theorem shows that, assuming that Conjecture 1 holds, our cryptosystem parametrized in this way, with n and q detailed in Table 1, provably has 128 bits of the claimed security. Furthermore, these specific parametrizations use the smallest possible n for which this is guaranteed by Conjecture 1 with our approach to choosing parameters.

Theorem 6.3. *Assuming that Conjecture 1 holds, there is no adversary against 128-bits of OW-CPA (IND-CCA) security of the parametrizations of our PKE (KEM) scheme that are detailed in table 1.*

Proof. First we consider the parametrizations of our PKE scheme that are claimed to have 128-bits of OW-CPA security. Any adversary against the claimed security of these systems could be used to create a quantum algorithm for solving approximate SIVP, using either Theorem 5.1 or Theorem 5.2. This quantum algorithm is more efficient than what is possible according to Conjecture 1, meaning that no such adversary can exist if the Conjecture is true.

In more detail, both the algorithms corresponding to Theorem 5.1 and Theorem 5.2 solve SIVP_{γ_R} with the approximation factor

$$\gamma_R(n) = q \cdot \sqrt{\frac{2n \cdot \ln(2n(1 + 1/\varepsilon))}{\pi}} \ ,$$

in time T_R that is detailed in these theorems. An adversary against the OW-CPA security of our PKE scheme that achieves an advantage ε_a in time T_a breaks the claimed d bits of security if $T_a/\varepsilon_a < 2^d$. For our parametrizations we consider the worst possible case, where $T_a = 1$ and $\varepsilon_a = 2^{-d}$. By considering such an adversary we can calculate a concrete reduction running time for our parametrizations. This running time is presented in the column $\lceil \log(T_R) \rceil$ of Table 1.

Security	OW-CPA	OW-CPA	IND-CCA	IND-CCA
Error distribution	Fixed	Variable	Fixed	Variable
n	39419	32890	104056	79510
$\lceil \log(q) \rceil$	34	34	37	37
$\lceil \log(T_R) \rceil$	614	491	1796	1313
$\lceil \log(\gamma_R) \rceil$	50	50	55	54
β	2395	1915	7007	5122

Table 1: Values for n and $\lceil \log(q) \rceil$ for different parameterizations that target 128 bits of the type of security specified in "Security" row with type of system given by the "Error distribution" row. The $\lceil \log(T_R) \rceil$ and $\lceil \log(\gamma_R) \rceil$ rows indicates the reduction running time and its approximation factor respectively. Finally, the β row shows which block-size β the reductions performance is compared against.

Next, we compare this computed reduction running time to the conjectured optimal performance from Conjecture 1. To this end, we let p_R be the success probability of the reduction and set $\beta = \beta(n) = \log(T_R/p_R)/0.2563$. This gives that the conjectured optimal approximation factor that the reduction can achieve is

$$\gamma_C(n) = \left((\pi\beta)^{1/\beta} \frac{\beta}{2\pi e} \right)^{\frac{n}{2(\beta-1)}} \cdot \frac{1}{\sqrt{2n\pi e}} \ .$$

The parameters are chosen such that we have $\gamma_C(n) > \gamma_R(n)$ and the reduction thus solves SIVP_{γ_R} more efficiently than what Conjecture 1 claims to be possible. Thus, no adversary against the claimed security of the proposed parametrizations can exist, unless Conjecture 1 is incorrect.

The concrete parameters presented in Table 1 are chosen by performing a search over n. For a given n, the values for q and α are selected so that q is a prime, $\alpha q > 2\sqrt{n}$ and the decryption failure probability is sufficiently small, as detailed in subsection 3.1. With α,q and n we then calculate γ_C and γ_R and compare them in order to determine if the parametrization is provably secure under Conjecture 1. The parameters presented in Table 1 are with the minimal n that, with parameters selected in this way, have $\gamma_C > \gamma_R$.

For the parametrizations of our IND-CCA secure KEM, the same reasoning is used to prove the OW-CPA security of the PKE scheme that is used in the $FO^{\not\perp}$ transform. In order for the resulting parametrizations to claim 128 bits of IND-CCA security, these PKE schemes target 516 bits of OW-CPA security as detailed in Section 6.3. Besides also targeting different decryption failure probabilities, the presented parameters are found in exactly the same way as for our OW-CPA secure PKE schemes. □

The proposed parametrizations use significantly larger dimensions n than

other lattice based cryptosystems. Part of the reason for the larger parameters is the tightness gap of the used reductions. However, even if we completely disregard the tightness gap of the reduction, our conservative hardness estimate for approximate SIVP still leads to a parametrization with $n \approx 9400$. As such, an arguably bigger reason for the large parameters in our parametrizations is due to the large approximation factors γ_R for the SIVP_{γ_R} instances solved by the reduction, resulting in the problem seemingly being easier than corresponding LWE instances.

In order to compare the performance of our reduction to the analysis in [9,33] we consider the tightness gap of our reduction. The tightness gap is calculated in the same way as in these works and is thus given by $T_R/p_R \cdot 2^{-128}$ where T_R is the reduction running time and p_R is its success probability. With our parametrization, the success probability is at least $9/10$ and our reduction running time is given in Table 1.

We note that the tightness gaps for the OW-CPA secure cryptosystems constructed in this paper are smaller than the gap of 2^{524} that was calculated for $n = 1024$ in [33]. This is the case even though the previous analysis did not take account for the large number of LWE samples required in order to use an adversary to solve DLWE with a negligible failure probability. Furthermore, the concrete value of the tightness gap depends on n, meaning that the larger values for n used for our parametrization also increase the tightness gap. As the calculated tightness gap is still comparable to the previously claimed values, it is clear that the more thorough analysis of the reduction allowed a significantly more efficient reduction.

6.5 Parameter lower bound

The cryptosystems parametrized in the previous section are provably secure based only on assuming that Conjecture 1 is correct. This conjectured hardness of SIVP_γ essentially corresponds to assuming that the heuristic lattice algorithms that are known today are optimal. However, Theorems 5.1 and 5.2 solve worst-case SIVP_{γ_R}, a problem that potentially could be significantly harder than the average-case problem. As there seems to be no research directly indicating that this is the case, this can not reasonably be accounted for in the parametrization of a cryptosystem.

There is however nothing guaranteeing that there exist some worst-case instances of SIVP_{γ_R} that are significantly harder to solve than typical instances. In particular, it is not guaranteed that efficient heuristic lattice algorithms are able to always solve this worst-case problem. If there exists such hard instances of SIVP_{γ_R}, we could base the security of our cryptosystem on their hardness instead of on the hardness of the problem on random lattices. This would allow the reduction to support the security of cryptosystems that use smaller parame-

ters than the ones in Table 1. While not a reasonable assumption for a concrete parametrization, the potential of harder SIVP_{γ_R} instances should be accounted for if we want to bound which parameters could be argued to have security, at least partially, supported by the reduction.

We consider the performance of provably correct lattice algorithms in order to bound the hardness of a worst-case instance of SIVP_{γ_R}. These algorithms are significantly less efficient than the heuristic algorithms that are typically considered, but as they are provably correct, these algorithms should be able to solve arbitrary SIVP_{γ_R} instances. In this section, we parametrize a our cryptosystem by using such a bound on the hardness of worst-case SIVP_{γ_R}. This serves as a bound on which parameters can claim any support from this reduction. Attacks against these parametrizations does not necessarily result in improved lattice algorithms in practice. Instead, such an attack would result in an improved provably correct lattice algorithms.

There does not seem to have been any significant research of algorithms that provably solve SIVP_{γ_R} for the relevant approximation factors. Instead, it seems that the most efficient way to provably solve SIVP_{γ_R} is via the following lemma from [36].

Lemma 6.4 (Corollary 4.2 from [36])**.** *For any $\gamma = \gamma(n) > 1$ there is a dimension preserving reduction from $SIVP_{\gamma'}$ to γ-SVP where*

$$\gamma' = \frac{\sqrt{n+3}}{2} \cdot \gamma$$

and the reduction use the γ-SVP oracle n times.

There are other algorithms that provably solve approximate SIVP directly, such as the algorithm from [1] that provably solves $\text{SIVP}_{\mathcal{O}(\sqrt{n \log n})}$ in time $2^{n/2+o(n)}$. However, this approximation factor is significantly smaller than the reduction approximation factor γ_R. Using Lemma 6.4 with an algorithm that solves $(2\gamma_R/\sqrt{n+3})$-SVP is therefore a more efficient method to solve SIVP_{γ_R}. This allows comparing the performance of well studied provably correct lattice reduction algorithms against the performance of the reductions from Theorems 5.1 and 5.2.

The slide-reduction [13] algorithm has the best provable performance in solving the relevant approximate SVP instances. As with BKZ, slide-reduction works by solving solving (almost) exact SVP in projected sublattices of dimension β. It seems likely that the SVP instances that must be solved during a slide-reduction can be solved by heuristic algorithms. Even in a potential worst-case lattice, we can randomize the initial basis which randomizes the projected sublattices in which SVP must be solved. However, the distribution of these sublattices is hard to predict and it could be the case that there exist worst-case lattices that fails to be slide-reduced when using a heuristic SVP algorithm.

We therefore consider the performance of SVP solvers that are provably correct. The most efficient provably correct SVP solver is the one presented in [2], which has an asymptotic run time of $2^{k/2+o(k)}$. To exactly analyze the concrete performance of this algorithm is outside the scope of this paper. Instead, we conservatively assume that using slide reduction with this SVP solver has a running time of $2^{0.5\beta}$ for blocksize β while finding a vector of the length given in (1). This ignores both polynomial factors in running time of the algorithm as well as more complicated expression for the approximation factor present in the algorithm from [2]. However, this still gives an indication as to what parameters can claim partial support from the reduction.

Combining the performance of slide reduction using this provable SVP solver together with Lemma 6.4 provides an upper bound on how hard a worst-case SIVP_{γ_R} instance could be, as detailed in the following remark. This remark is used to lower bound parameters that can claim partial support from our reduction, similarly to how Conjecture 1 is used for the parametrization in section 6.4.

Remark 1 (Upper bound on hardness of approximate SIVP)**.** *There is a quantum algorithm that provably solves worst-case $SIVP_\gamma$ in time $n \cdot 2^{\beta/2}$ for approximation factor*

$$\gamma = \frac{\sqrt{n+3}}{2} \cdot \left(\frac{1.744n}{2\pi e}\right)^{(n-\beta)/(\beta-1)} .$$

The parameters in Table 2 are chosen so that any attack against the claimed security would imply a more efficient algorithm than the one corresponding to Remark 1. The parameters are chosen minimal in the same way as in Section 6.4. Therefore, attacks against cryptosystems with smaller parameters will not even result in a lattice algorithm with better provable performance than current algorithms. This shows that the parameters that are used for LWE based cryptosystems in practice are significantly smaller than what is needed to claim any concrete security from Regev's quantum reduction. Attacks against these cryptosystems could therefore improve significantly without necessarily implying any progress in algorithms for standard lattice problems.

7 Conclusion

The parameters we use for our cryptosystem are significantly larger than what is typically used for LWE-based cryptosystems. A large reason for this is the inefficiency of Regev's quantum reduction and further improving its efficiency could potentially allow it to support the security of cryptosystems that use smaller parameters. However, only optimizing the run time of the reduction is insufficient for it to support the security of typical LWE-based cryptosystems.

Security	OW-CPA	OW-CPA	IND-CCA	IND-CCA
Error distribution	Fixed	Variable	Fixed	Variable
n	5782	4701	17093	12767
$\lceil \log(q) \rceil$	30	30	34	33
$\lceil \log(T_R) \rceil$	590	470	1774	1293
$\lceil \log(\gamma_R) \rceil$	43	42	48	48
β	1155	915	3519	2559

Table 2: Equivalent parametrizations as in Table 1 but instead of assuming Conjecture 1 assumes that Remark 1 corresponds to an optimal algorithm.

To see this, we consider a parametrization where we completely disregard the run time of the reduction and thus assume that an adversary against the cryptosystem implies a solution to SIVP_{γ_R} in time 2^{128}. This approach still results in a parametrization using a dimension $n \approx 9400$ to argue for 128 bits of OW-CPA security. Thus, the large approximation factors for which the reduction solves approximate SIVP is an arguably bigger reason for the large parameters required in our cryptosystem.

Our primary parametrizations do not account for the fact that the reduction solves a worst-case instance of approximate SIVP. In practice, there does not seem to be anything that indicates that such a worst-case instance is significantly harder to solve than a typical problem instance. Accounting for the possibility of harder instances of approximate SIVP can be done by comparing the performance of the reduction and that of provably correct lattice algorithms. However, even such a comparison is unable to support the security of cryptosystems with parameters of similar size to those of typical LWE-based schemes.

As such, it seems hard to improve this reduction to such an extent that it actually supports parametrizations close to those of typical LWE-based schemes. Any such improvement would not only have to significantly improve the reductions efficiency, but also solve a harder lattice problem.

This does not directly indicate that typical LWE-based cryptosystems are insecure. However, it does mean that attacks against these systems could improve significantly, without necessarily implying any progress in algorithms for general lattice problems. As such, arguments about the concrete security of typical lattice-based cryptosystems can not reasonably be considered to be supported by similar worst-case to average-case reductions.

In contrast to this, our parametrizations are provably secure based on the assumed hardness of standard lattice problems. Thus, unless lattice algorithms improve significantly, these parametrizations are guaranteed to be secure. While far too inefficient for most use-cases, our parametrizations actually provide the first concrete systems with provable security based only on the hardness of

standard lattice problems.

7.0.1 Acknowledgments

This research has been supported in part by the Swedish Armed Forces and was conducted at KTH Center for Cyber Defense and Information Security (CDIS). The author would like to thank Johan Håstad for his helpful input.

References

[1] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time via discrete gaussian sampling, 2015.

[2] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-svp and $\sqrt{n}$-hermite svp, and an improved time-approximation tradeoff for (h)svp, 2020.

[3] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! Cryptology ePrint Archive, Report 2018/331, 2018. `https://eprint.iacr.org/2018/331`.

[4] Erdem Alkim, Joppe W. Bos, Léo Ducas, Patrick Longa, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, Ilya Mironov, and Douglas Stebila. FrodoKEM: Learning with errors key encapsulation. `https://frodokem.org/`, June 2021. Submission to the NIST Post-Quantum Cryptography standardization project, Round 3.

[5] L. Babai. On lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[6] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.

[7] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 10–24, 2016.

[8] Xavier Bonnetain, André Chailloux, André Schrottenloher, and Yixin Shen. Finding many collisions via reusable quantum walks: Application to lattice sieving. pages 221–251, 2023.

[9] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness ii: Practical issues in cryptography. Cryptology ePrint Archive, Report 2016/360, 2016. https://eprint.iacr.org/2016/360.

[10] Yuanmi Chen. *Réduction de réseau et sécurité concrète du chiffrement complètement homomorphe*. PhD thesis, Université Paris Diderot, 2013. 2013PA077242.

[11] M. Chiani, D. Dardari, and M.K. Simon. New exponential bounds and approximations for the computation of error probability in fading channels. *IEEE Transactions on Wireless Communications*, 2(4):840–845, 2003.

[12] J.H. Conway, N.J.A. Sloane, E. Bannai, R.E. Borcherds, J. Leech, S.P. Norton, A.M. Odlyzko, R.A. Parker, L. Queen, and B.B. Venkov. *Sphere Packings, Lattices and Groups*. Grundlehren der mathematischen Wissenschaften. Springer New York, 2013.

[13] Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within mordell's inequality. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 207–216, New York, NY, USA, 2008. Association for Computing Machinery.

[14] Fletcher Gates. Reduction-respecting parameters for lattice-based cryptosystems. Master's thesis, McMaster University, 2018.

[15] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. pages 197–206, 2008.

[16] Lov K. Grover and T. Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv: Quantum Physics*, 2002.

[17] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[18] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography*, pages 341–371, Cham, 2017. Springer International Publishing.

[19] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. Cryptology ePrint Archive, Report 2017/1096, 2017. https://eprint.iacr.org/2017/1096.

[20] Neal Koblitz, Subhabrata Samajder, Palash Sarkar, and Subhadip Singha. Concrete analysis of approximate ideal-sivp to decision ring-lwe reduction. Cryptology ePrint Archive, Report 2022/275, 2022. https://ia.cr/2022/275.

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

[22] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[23] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 319–339, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[24] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. pages 1–23, 2010.

[25] Daniele Micciancio and Oded Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[26] Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`.

[27] Michael A. Nielsen and Isaac Chuang. Quantum computation and quantum information. *American Journal of Physics*, 70(5):558–559, 2002.

[28] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 461–473, New York, NY, USA, 2017. Association for Computing Machinery.

[29] Chris Pekert. What does gchq's "cautionary tale" mean for lattice cryptography?

[30] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, New York, NY, USA, 2005. Association for Computing Machinery.

[31] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009.

[32] C. A. Rogers. The number of lattice points in a set. *Proceedings of the London Mathematical Society*, s3-6(2):305–320, 1956.

[33] Palash Sarkar and Subhadip Singha. Verifying solutions to lwe with implications for concrete security. Cryptology ePrint Archive, Report 2019/728, 2019. https://eprint.iacr.org/2019/728.

[34] Claus Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 08 1994.

[35] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[36] Noah Stephens-Davidowitz. Dimension-preserving reductions between lattice problems. *http://www.noahsd.com/latticeproblems.pdf*, 2016.

[37] Anders Södergren. On the poisson distribution of lengths of lattice vectors in a random lattice, 2010.

A Lemmas Used for Proof of Theorem 4.3

Here we present the remaining parts of the full proof from [31] while we keep track of the reduction tightness and the number of LWE samples it requires.

To begin with, we introduce some background lemmas that are used in the proofs of this section. First is this following version of the Poisson summation formula.

Lemma A.1 (Poisson summation formula)**.** *For any lattice L and any function $f : \mathbb{R}^n \to \mathbb{C}$*

$$f(L) = \det(L^*)\hat{f}(L^*)\,.$$

Next, this following lemma is used to bound the difference in value on a Gaussian function between two points that are at most a distance l from each other.

Claim A.2 (Claim 2.1 from [31])**.** *For all $s, t, l > 0$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\| \leq t$ and $\|\boldsymbol{x} - \boldsymbol{y}\| \leq l$*

$$\rho_s(\boldsymbol{y}) \geq (1 - \pi(2lt + l^2)/s^2)\rho_s(\boldsymbol{x})$$

Finally the following bound on the value of ν_s on a lattice follows from Lemma A.1.

Claim A.3. *For any $r \in \mathbb{R}$ and lattice $L \subseteq \mathbb{Z}^n$ we have that $\nu_r(L) \geq \det(L^*)$ and if $r > \lambda_1^*/\sqrt{n}$ we also have $\nu_r(L) \leq \det(L^*)(1 + 2^{-2n})$.*

Proof. By Lemma A.1 we have that $\rho_r(L) = \det(L^*) \cdot r^n \rho_{1/r}(L^*) \geq \det(L^*) \cdot r^n$ giving

$$\sum_{\boldsymbol{x} \in L} \nu_r(\boldsymbol{x}) = \rho_r(L)/r^n \geq \det(L^*)$$

Furthermore, if $r > \sqrt{n}/\lambda_1(L^*)$ we can apply Lemma 2.2 giving

$$\nu_r(L) = \det(L^*) \cdot \rho_{1/r}(L^*) \leq \det(L^*) \cdot (1 + 2^{-2n})$$

□

The iterative procedure used by Regev's reduction requires samples from $D_{L,r}$ in order to produce samples from $D_{L,r'}$ with $r' < r$. To start this process we must therefore have samples from $D_{L,r}$ for some r. Such samples are produced by using the following bootstrapping lemma. The proof is the same as in [31] but keeps track of the statistical distance from the target distribution.

Lemma A.4 (Bootstrapping, Lemma 3.2 from [31])**.** *There exists an efficient algorithm that, given any n-dimensional lattice L and $r > 2^{2n}\lambda_n(L)$, outputs a sample from a distribution that is within statistical distance $2^{-n/2}$ of $D_{L,r}$ if $n \geq 20$.*

Proof. The procedure begins by using LLL basis reduction algorithm [22] to obtain a basis $\boldsymbol{B}$ for L where the longest vector of B has length at most $2^n\lambda_n(L)$. Next, a vector $\boldsymbol{y} \in \mathbb{R}^n$ is sampled from ν_r, which can be done efficiently with high precision. The output of the procedure is given by $\boldsymbol{y} - (\boldsymbol{y} \mod \mathcal{P}(\boldsymbol{B})) \in L$, which is efficiently computable from $\boldsymbol{y}$ and $\boldsymbol{B}$.

To show its correctness, we first note that by Lemma 2.2 only $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| \leq r\sqrt{n}$ have to be considered. By definition, the target distribution $D_{L,r}$ is $\rho_r(\boldsymbol{x})/\rho_r(L)$ where Lemma A.1 gives the denominator as

$$\rho_r(L) = \det(L^*) \cdot r^n \cdot \rho_{1/r}(L^*) \geq \det(L^*) \cdot r^n \ .$$

As such, the probability of any $\boldsymbol{x}$ in the target distribution is at most

$$\rho_r(\boldsymbol{x})/(\det(L^*) \cdot r^n) = \det(L)\nu_r(\boldsymbol{x}) \, .$$

On the other hand, our procedure samples an $\boldsymbol{x} \in L$ with the probability

$$\begin{aligned}\int_{\boldsymbol{x}+\mathcal{P}(L)} \nu_r(\boldsymbol{y})d\boldsymbol{y} &\geq \int_{\boldsymbol{x}+\mathcal{P}(L)} \left(1 - \pi\frac{2\|\boldsymbol{x}\|\|\boldsymbol{y}-\boldsymbol{x}\| + \|\boldsymbol{y}-\boldsymbol{x}\|^2}{r^2}\right)\nu_r(\boldsymbol{x})d\boldsymbol{y} \\ &\geq (1 - 3\pi n^{1.5}2^{-n})\det(L)\nu_r(\boldsymbol{x})\end{aligned}$$

where the first inequality use claim A.2 and the second inequality use that $\|\boldsymbol{y} - \boldsymbol{x}\| \leq \text{diam}(\mathcal{P}(L)) \leq n \cdot 2^n \lambda_n(L)$, $\|\boldsymbol{x}\| \leq r\sqrt{n}$ and $r \geq 2^{2n}\lambda_n(L)$. This gives a statistical distance between the target and the produced distribution of no more than

$$\sum_{\boldsymbol{x} \in L} 3\pi n^{1.5} 2^{-n} \cdot \det(L) \nu_r(\boldsymbol{x}) \leq 3\pi n^{1.5} 2^{-n}(1 + 2^{-2n}) \leq 2^{-n/2}$$

where the first inequality follows from claim A.3 as $\lambda_1(L^*) \geq 1/\lambda_n(L) > \sqrt{n}/r$. The final inequality holds for $n \geq 20$ and is not very tight for larger n. However, as the statistical distance of $2^{-n/2}$ is sufficiently small this does not impact the results. □

There are other algorithms that can sample from discrete Gaussian distributions with smaller standard deviation, such as the sampler from [15]. However, there are no efficient algorithms that produce samples from $D_{L,r}$ for r that is some subexponential factor larger than λ_n. As such, using a better discrete Gaussian samplers would only have a minor impact on the reduction and we would still require $\mathcal{O}(n)$ iterative steps to produce samples from the target distribution $D_{L,\text{poly}(n)\cdot\lambda_n}$. Because of this, for simplicity we choose to use the same discrete Gaussian sampler as in [31].

The bootstrapping procedure is used to produce input samples to the first iterative step, where these samples together with the LWE oracle are used in order to solve a BDD instance on the dual of the lattice. This process that, with the help of discrete Gaussian samples and an LWE oracle, solves BDD on the dual of the lattice is detailed in the following lemma.

Lemma A.5 (Lemma 3.4 from [31])**.** *Let $\varepsilon = \varepsilon(n)$ be a negligible function, $q = q(n) \geq 2$ be an integer and $\alpha = \alpha(n) \in (0, 1)$ be a real number. Furthermore, let L be any n-dimensional lattice and r be a number such that $r > \sqrt{2}q\eta_\varepsilon(L)$. Assume that we have access to an oracle W that can solve LWE(Ψ_β, M) for arbitrary $\beta \in [\alpha/\sqrt{2}, \alpha]$ with a failure probability of at most ε. Then, there exists an algorithm that, except for with probability at most $9Mn\varepsilon$, solves BDD$_{L^*,\alpha q/(\sqrt{2}r)}$ by using M samples from $D_{L,r}$ and by using W at most n times.*

The proof of this lemma consists of a combination of several other lemmas and we present these before actually proving Lemma A.5. First, the next lemma essentially says that, if we can solve a BDD$_{L,d}$ instance modulo q, then we can also solve the full problem. To formalize this, we define the following intermediate problem.

Definition 12 (BDD$^{(q)}_{L,d}$)**.** An instance of the BDD$^{(q)}_{L,d}$ problem is given by a vector $\boldsymbol{x}$ that is guaranteed to be at most a distance d from the lattice L. The problems solution is $\boldsymbol{B}^{-1}\kappa_L(\boldsymbol{x}) \mod q$ for an arbitrary basis $\boldsymbol{B}$ of the lattice L.

The idea behind the proof is to find the solution to the initial $\mathrm{BDD}_{L,d}$ instance modulo q by using the $\mathrm{BDD}^{(q)}_{L,d}$ oracle, remove this part from the initial problem instance and then divide by q. As the solution modulo q is removed, this result in a correctly constructed BDD instance, but where the distance from the lattice has decreased by a factor q. Repeating this process eventually results in a BDD instance that can be solved efficiently by using Babai's algorithm. The steps that led to this easy BDD instance are then performed backwards in order to provide a solution to the original BDD instance. The proof follows the same steps as the proof in [31] but keeps track of number of required calls and error probabilities.

Lemma A.6 (Lemma 3.5 from [31])**.** *Let L be a lattice, $d < \lambda_1(L)/2$ some number, $q \geq 2$ an integer and W be an oracle that solves $BDD^{(q)}_{L,d}$ with failure probability at most ε. Then there exists an algorithm that, except for with probability at most $n\varepsilon$, solves $BDD_{L,d}$ by using W a total of n times.*

Proof. Let $\boldsymbol{B}$ be any basis for the lattice L. The input to the problem is a point $\boldsymbol{x}$ within distance d of L. We define the sequence of points $\boldsymbol{x}_1 = \boldsymbol{x}, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots$ by letting $\boldsymbol{a}_i = \boldsymbol{B}^{-1}\kappa_L(\boldsymbol{x}_i) \in \mathbb{Z}^n$ and $\boldsymbol{x}_{i+1} = (\boldsymbol{x}_i - \boldsymbol{B}(\boldsymbol{a}_i \mod q))/q$. By using the oracle W we are able to calculate $\boldsymbol{a}_i \mod q = \boldsymbol{B}^{-1}\kappa_L(\boldsymbol{x}_i) \mod q$ if $\boldsymbol{x}_i$ is close enough to L. The distance between $\boldsymbol{x}_1$ to L is at most d and each further step in the series decreases the distance between $\boldsymbol{x}_i$ and the lattice by a factor q, meaning that $\boldsymbol{x}_{i+1}$ is at most a distance d/q^i from L. Therefore, the points $\boldsymbol{x}_i$ are all close enough to the lattice for the oracle W to be used to calculate the next point in the series.

Using the oracle n times thus allows us to calculate the whole sequence up to $\boldsymbol{x}_{n+1}$. We are guaranteed that $\boldsymbol{x}_{n+1}$ is within a distance d/q^n from the lattice. This allows us to use Babai's nearest plane algorithm [5] to efficiently find the lattice point closest to $\boldsymbol{x}_{n+1}$. The resulting lattice point $\boldsymbol{Ba}$ is at most a distance $d/q^n < \lambda_1(L)/2$ from $\boldsymbol{x}_{n+1}$ and is thus the unique lattice point closest to $\boldsymbol{x}_{n+1}$. As such, we have recovered $\boldsymbol{a}_{n+1} = \boldsymbol{a}$ from which it is easy to calculate all the $\boldsymbol{a}_i$ via

$$\boldsymbol{a}_i = q\boldsymbol{a}_{i+1} + (\boldsymbol{a}_i \mod q)$$

where $(\boldsymbol{a}_i \mod q)$ are the solutions to $\mathrm{BDD}^{(q)}_{L,d}$ that have already been calculated. This allows calculating the whole series $\boldsymbol{a}_{n+1}, \boldsymbol{a}_n \ldots, \boldsymbol{a}_1$ which gives the lattice point closest to $\boldsymbol{x} = \boldsymbol{x}_1$ via $\boldsymbol{Ba}_1$.

Next, we bound the failure probability of this procedure by noting that we require all n calls to the $\mathrm{BDD}^{(q)}$ oracle to succeed. This happens with probability at least

$$(1-\varepsilon)^n \geq 1 - n\varepsilon$$

which gives the claimed procedure failure probability of at most $n\varepsilon$. □

This lemma could be somewhat optimized by applying Babai's algorithm on $\boldsymbol{x}_i$ for some $i < n$ instead of $i = n + 1$. For example, $i = n/\log(q)$ steps should be sufficient as then $d/q^i = d \cdot 2^{-n} \leq \lambda_1 \cdot 2^{-n}$ is small enough that Babai's algorithm can find the lattice point closest to $\boldsymbol{x}_i$. However, this is only a minor factor in the performance of the full reduction and does not significantly alter the end result.

The next lemma was the focus of the work of [33] where it was noted that, if used exactly as stated by Regev in [31], it would require n to be around 400000 in order for it to have a reasonable failure probability. However, the work also noted that for parametrizations of LWE based schemes that are used in practice, a minor tweak of the lemma would be sufficient for it to have a reasonable failure probability. The lemma is used to verify whether or not a candidate solution to an LWE instance actually is the solution.

Lemma A.7 (Verifying solution of LWE, Lemma 3.6 of [31])**.** *Let $q = q(n) \geq 1$ be some integer. There exists an efficient algorithm that, given $\boldsymbol{s}'$ and m samples from $A_{\boldsymbol{s},\Psi_\alpha}$ for some (unknown) $\boldsymbol{s} \in \mathbb{Z}_q^n$ and $\alpha < 1/\sqrt{n}$, outputs whether $\boldsymbol{s} = \boldsymbol{s}'$ and is correct except with probability at most $\exp(-9m/80)$ assuming $n > 60$.*

Proof. The idea is to perform a statistical test on the samples from $A_{\boldsymbol{s},\Psi_\alpha}$ that checks if $\boldsymbol{s}' = \boldsymbol{s}$. Let ξ be the distribution obtained by taking samples $(\boldsymbol{a}, x)$ from $A_{\boldsymbol{s},\Psi_\alpha}$ and outputting $y = x - \langle \boldsymbol{a}, \boldsymbol{s}' \rangle / p \mod 1$. The test consists of taking m samples y_i from ξ, calculating

$$z = \frac{1}{m} \sum_{i=1}^{m} \cos(2\pi y_i)$$

and accepting that $\boldsymbol{s}' = \boldsymbol{s}$ if $z > t$. In Regev's analysis [31] of this procedure, $t = 0.02$ and $m = n$ was used to get a proof that asymptotically gives the correct answer with overwhelming probability. Later Sarkar and Singha showed [33] that this would require a very large n in practice. However, in the same paper they also note that, with a different value for t and with values of α used in practice, the same idea works for reasonably small values of n.

To show that the test correctly distinguishes between $\boldsymbol{s}' = \boldsymbol{s}$ and $\boldsymbol{s}' \neq \boldsymbol{s}$ we let

$$(\boldsymbol{a}, x = \boldsymbol{a}\boldsymbol{s} + e) \leftarrow A_{\boldsymbol{s},\Psi_\alpha}$$

with $e \leftarrow \Psi_\alpha$. This gives that a sample from ξ is given by

$$x - \langle \boldsymbol{a}, \boldsymbol{s}' \rangle / q \mod 1 = \langle \boldsymbol{a}, \boldsymbol{s} - \boldsymbol{s}' \rangle / q + e \mod 1$$

and thus if $\boldsymbol{s} = \boldsymbol{s}'$, this sample from ξ equals e and thus $\xi = \Psi_\alpha$. Furthermore, if $\boldsymbol{s} \neq \boldsymbol{s}'$ then ξ will have some periodicity $1/k$. To see this, consider some coordinate j such that $s'_j \neq s_j$. For this j we can see that $a_j(s_j - s'_j) \mod q$ is periodic with period $\gcd(q, s_j - s'_j) < q$. This implies that $a_j(s_j - s'_j)/q$

mod 1 has some period $1/k$ with $2 \leq k \leq q$. Since a sample from ξ is obtained by adding $a_j(s_j - s'_j)/q \mod 1$ and an independent sample from some other distribution, this shows that ξ also has the same period $1/k$.

Next, the expectation $\tilde{z}$ of $\cos(2\pi y)$ with $y \leftarrow \xi$ is calculated as

$$\tilde{z} = \operatorname*{Exp}_{y \sim \xi}[\cos(2\pi y)] = \int_0^1 \cos(2\pi y)\xi(y)dy = \operatorname{Re}\left[\int_0^1 \exp(2\pi i y)\,\xi(y)dy\right]$$

where a calculation shows that $\xi = \Psi_\alpha$ gives $\tilde{z} = \exp\left(-\pi\alpha^2\right)$. Furthermore, if ξ has a period $1/k$ then

$$\begin{aligned}\tilde{z} = \int_0^1 \exp(2\pi i y)\,\xi(y)dy &= \int_0^1 \exp\left(2\pi i\left(y + \frac{1}{k}\right)\right)\xi(y)dy \\ &= \exp(2\pi i/k)\int_0^1 \exp(2\pi i y)\,\xi(y)dy\end{aligned}$$

which is possible only if $\tilde{z} = 0$ as $k \geq 2$.

Now the probability of getting the incorrect result is bounded by using the Hoeffding inequality, in the same way as done in [33]. Let $\xi_0 = \Psi_\alpha$ be the distribution of y if $\boldsymbol{s} = \boldsymbol{s}'$ and ξ_1 be the distribution of y if $\boldsymbol{s} \neq \boldsymbol{s}'$. The Hoeffding inequality gives that the probability to incorrectly claim that $\boldsymbol{s}$ does not equal $\boldsymbol{s}'$ when $\boldsymbol{s} = \boldsymbol{s}'$ is

$$\Pr_{y \sim \xi_0}[z \leq t] = \Pr_{y \sim \xi_0}[z - \mu_0 \leq -(\mu_0 - t)] \leq \exp\left(-m(\mu_0 - t)^2/2\right)$$

where $\mu_0 = \exp(-\pi\alpha^2)$ is the expected value of Ψ_α. Meanwhile, the probability to incorrectly claim that $\boldsymbol{s}$ equals $\boldsymbol{s}'$ when $\boldsymbol{s} \neq \boldsymbol{s}'$ is bounded by

$$\Pr_{y \sim \xi_1}[z > t] \leq \exp\left(-mt^2/2\right)$$

as the expected value of ξ_1 is 0.

The specific choice of t can be tweaked somewhat in order to change which types of failure are more common, incorrectly accepting an $\boldsymbol{s}' \neq \boldsymbol{s}$ or rejecting the correct $\boldsymbol{s}' = \boldsymbol{s}$. However, this has a negligible impact on the end result and t is instead chosen so that both types of failures have the same probability. This means that we choose $\mu_0 - t = t$ and thus

$$t = \mu_0/2 = \exp\left(-\pi\alpha^2\right)/2$$

which gives that the probability is $\exp(-m\exp(-2\pi\alpha^2)/8)$ for both types of errors. With $\alpha < 1/\sqrt{n}$ this probability is at most $\exp(-m\exp(-2\pi/n)/8)$ and with $n > 60$ we have $\exp(-2\pi/n) > 0.9$ which gives the claimed probability of less than $\exp(-9m/80))$ of incorrectly verifying a solution. □

Next is a lemma that is used in upcoming proofs in order to translate an $\mathrm{BDD}^{(q)}_{L^*,d}$ instance into samples from an LWE distribution. The lemma is taken directly from a corollary in [31] that already includes the statistical distance between the two distributions being compared. Because of this, there is no need for further analysis of this lemma and it is included here without a proof.

Lemma A.8 (Corollary 3.10 from [31])**.** *Let L be a lattice, let $\boldsymbol{z}, \boldsymbol{u} \in \mathbb{R}^n$ be vectors, and let $r, \alpha > 0$ be two reals. Assume that*

$$1/\sqrt{1/r^2 + (\|\boldsymbol{z}\|/\alpha)^2} \geq \eta_\varepsilon(L)$$

for some $\varepsilon < \frac{1}{2}$. Then the distribution of $\langle \boldsymbol{z}, \boldsymbol{v} \rangle + e$ where $\boldsymbol{v}$ is distributed according to $D_{L+\boldsymbol{u},r}$ and e is a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, is within statistical distance 4ε of a normal variable with mean 0 and standard deviation $\sqrt{(r\,\|\boldsymbol{z}\|)^2 + \alpha^2}/\sqrt{2\pi}$. In particular, since statistical distance cannot increase by applying a function, the distribution of $\langle \boldsymbol{z}, \boldsymbol{v} \rangle + e \mod 1$ is within statistical distance 4ε of $\Psi_{\sqrt{(r\|\boldsymbol{z}\|)^2+\alpha^2}}$.

Using this lemma, we are able to transform an $\mathrm{BDD}^{(q)}_{L^*,\alpha q/(r\sqrt{2})}$ instance together with samples from $D_{L,r}$ into samples from an LWE distribution, as described in the following lemma. Our proof is the same as in [31] but keeps track of some additional details.

Lemma A.9 (Lemma 3.11 from [31])**.** *Let $\varepsilon = \varepsilon(n) < 1/2$ be a function, $q = q(n) \geq 2$ be an integer, and $\alpha = \alpha(n) \in (0,1)$ be a real number. Assume that we have access to an oracle W that solves $\mathrm{LWE}(\Psi_\beta, M)$ for arbitrary $\beta \in [\alpha/\sqrt{2}, \alpha]$ with a failure probability of at most $M\varepsilon$. Then there exists an algorithm that, given an n-dimensional lattice L, a number $r > \sqrt{2}q\eta_\varepsilon(L)$, and M samples from $D_{L,r}$, solves $\mathrm{BDD}^{(q)}_{L^*,\alpha q/(r\sqrt{2})}$ except for with probability at most $9M\varepsilon$ by using W once.*

Proof. Let $\boldsymbol{B}$ be a basis for the lattice L and $\boldsymbol{B}^* = (\boldsymbol{B}^{-1})^T$ be the corresponding basis for the dual lattice L^*. Let $\boldsymbol{x}$ be the input to the $\mathrm{BDD}^{(q)}_{L^*,\alpha q/(r\sqrt{2})}$ instance, and thus $\boldsymbol{x}$ is at most a distance $\alpha q/(r\sqrt{2})$ from L^*. By using a sample from $D_{L,r}$ this input $\boldsymbol{x}$ is transformed into a sample from $A_{\boldsymbol{s},\Psi_\beta}$ for some $\beta \leq \alpha$ and $\boldsymbol{s} = (\boldsymbol{B}^*)^{-1}\kappa_{L^*}(\boldsymbol{x}) \mod q$. Repeating this procedure M times with M different samples from $D_{L,r}$ creates M samples from $A_{\boldsymbol{s},\Psi_\beta}$ and thus allows W to be used to recover $\boldsymbol{s}$, which is the solution to the input $\mathrm{BDD}^{(q)}_{L^*,\alpha q/(\sqrt{2}r)}$ instance.

To produce one of these samples from $A_{\boldsymbol{s},\Psi_\beta}$, a vector $\boldsymbol{v} \in L$ is sampled from $D_{L,r}$ and we let $\boldsymbol{a} = B^{-1}\boldsymbol{v} \mod q$. From this, the output sample is given by

$$(\boldsymbol{a}, \langle \boldsymbol{x}, \boldsymbol{v} \rangle/q + e \mod 1) \tag{11}$$

where $e \in \mathbb{R}$ is sampled from a normal distribution with standard deviation $\alpha/(2\sqrt{\pi})$.

We now show that these samples are of statistical distance at most 8ε from the distribution $A_{\boldsymbol{s},\Psi_\beta}$ for some $\beta \leq \alpha$. First, note that the probability of $\boldsymbol{a}$ is proportional to $\rho_r(qL + \boldsymbol{B}\boldsymbol{a})$ and, with $\eta_\varepsilon(qL) = q\eta_\varepsilon(L) < r$, Claim 2.6 shows that the probability of a specific $\boldsymbol{a}$ lies within the range $k(1 \pm \varepsilon)$ for some constant k. This shows that the distribution of $\boldsymbol{a}$ is a statistical distance at most 4ε from the uniform distribution since

$$q^n k(1-\varepsilon) \leq 1 \leq q^n k(1+\varepsilon)$$

and thus

$$\frac{1}{1+\varepsilon} \leq q^n k \leq \frac{1}{1-\varepsilon}$$

giving the statistical distance

$$\sum_{\boldsymbol{a}\in\mathbb{Z}_q^n} \left| \frac{1}{q^n} - p(\boldsymbol{a}) \right| \leq |1 - q^n k| + q^n k\varepsilon \leq \left| \frac{1}{1-\varepsilon} - 1 \right| + \frac{\varepsilon}{1-\varepsilon} \leq \frac{2\varepsilon}{1-\varepsilon} \leq 4\varepsilon\,.$$

Next, we consider the second part of (11) conditioned on a fixed value of $\boldsymbol{a}$. We define $\boldsymbol{x}' = \boldsymbol{x} - \kappa_{L^*}(\boldsymbol{x})$, note that $\|\boldsymbol{x}'\| \leq \alpha q/(\sqrt{2}r)$, and see that

$$\langle \boldsymbol{x}, \boldsymbol{v}\rangle/q + e \mod 1 = \langle \kappa_{L^*}(\boldsymbol{x}), \boldsymbol{v}\rangle/q + \langle \boldsymbol{x}', \boldsymbol{v}\rangle/q + e \mod 1\ .$$

Now we note that

$$\langle \kappa_{L^*}(\boldsymbol{x}), \boldsymbol{v}\rangle = \langle (\boldsymbol{B}^*)^{-1}\kappa_{L^*}(\boldsymbol{x}), \boldsymbol{B}^{-1}\boldsymbol{v}\rangle$$

since $\boldsymbol{B}^{-1} = (\boldsymbol{B}^*)^T$. Thus the inner product between $\kappa_{L^*}(\boldsymbol{x}) \in L^*$ and $\boldsymbol{v} \in L$ is the same as the inner product between their coefficient vectors in basis $\boldsymbol{B}$ and $\boldsymbol{B}^*$. This implies that

$$\langle \kappa_{L^*}(\boldsymbol{x}), \boldsymbol{v}\rangle \mod q = \langle \boldsymbol{s}, \boldsymbol{a}\rangle \mod q$$

and it follows that

$$\langle \kappa_{L^*}(\boldsymbol{x}), \boldsymbol{v}\rangle/q \mod 1 = \langle \boldsymbol{s}, \boldsymbol{a}\rangle/q \mod 1$$

which is the desired dependence on the secret for LWE samples.

Lemma A.8 shows that the statistical distance from Ψ_β for the remaining part, $\langle \boldsymbol{x}', \boldsymbol{v}\rangle/q + e \mod 1$, is at most 4ε with $\beta = \sqrt{(r\,\|\boldsymbol{x}'\|/q)^2 + \alpha^2/2}$. This β is always in the range $[\alpha/\sqrt{2}, \alpha]$ where we know that the provided LWE oracle works. We also see that Lemma A.8 is applicable as, condition on a fixed value for $\boldsymbol{a}$, the distribution of $\boldsymbol{v}$ is $D_{qL+\boldsymbol{B}\boldsymbol{a},r}$, the distribution of e is normal with standard deviation $(\alpha/\sqrt{2})/\sqrt{2\pi}$ and mean 0 and

$$\sqrt{\frac{1}{\frac{1}{r^2} + \left(\frac{\sqrt{2}\|\boldsymbol{x}'\|}{\alpha q}\right)^2}} \geq \frac{r}{\sqrt{2}} > \eta_\varepsilon(qL)\ .$$

In total this gives that a single sample from this produced distribution is at most a statistical distance of 8ε from the desired distribution, meaning that all M samples provided to W are at most a distance $8M\varepsilon$ from the correct input distribution to W. This implies that W gives the correct answer to the LWE instance except for with probability at most $8M\varepsilon + M\varepsilon = 9M\varepsilon$. If this single call to W is successful, it recovers $\boldsymbol{s} = (\boldsymbol{B}^*)^{-1}\kappa_{L^*}(\boldsymbol{x}) \mod q$ which solves the $\mathrm{BDD}^{(q)}_{L^*,\alpha q/(\sqrt{2}r)}$ instance. As such a single call to W is sufficient. □

To actually prove Lemma A.5 we combine Lemmas A.9 and Lemma A.6 in order to actually solve a $\mathrm{BDD}_{L^*,d}$ instance.

Proof of Lemma A.5. We let $d = \alpha q/(\sqrt{2}r)$ and use Lemma A.9 with W in order to solve $\mathrm{BDD}^{(q)}_{L^*,d}$ instances. Using W once, this results in a solutions to $\mathrm{BDD}^{(q)}_{L^*,d}$ that is incorrect with probability at most $9M\varepsilon$. The solution to the input $\mathrm{BDD}_{L^*,d}$ instance is then given by using Lemma A.6, resulting in an error probability of at most

$$9Mn\varepsilon$$

while requiring $9Mn$ calls to W. The algorithm must be successful for every $\mathrm{BDD}^{(q)}_{L^*,d}$ instance produced by Lemma A.6 in order for it to correctly solve the input $\mathrm{BDD}_{L^*,d}$ instance. Therefore, the same M samples from $D_{L,r}$ can be reused every time we use Lemma A.9. As such, M samples from $D_{L,r}$ is sufficient to solve the input $\mathrm{BDD}_{L^*,d}$ instance. □

This concludes the first part of the iterative step, where an LWE oracle is used to solve an instance of $\mathrm{BDD}_{L^*,d}$. In the second part of the iterative step, we construct a quantum state that corresponds to the desired output distribution. To construct this output distribution, we must solve a single $\mathrm{BDD}_{L^*,d}$ instance, which we accomplish by using the first part of the iterative step.

Before presenting the proof of this second part, we introduce the following two lemmas. These are equivalent to Lemmas 3.12 and 3.13 from [31] where the ℓ_2 distances between constructed and desired states were given as $2^{-\Omega(n)}$. The following versions give a concrete bound on the trace distance between the states instead of asymptotically bounding the ℓ_2 distance.

Lemma A.10 (Variant of Lemma 3.12 from [31])**.** *There exists an efficient quantum algorithm that, given an n-dimensional lattice $L \subseteq \mathbb{Z}^n$ and a number $r > 2^{2n}\lambda_n(L)$, outputs a state that is within trace distance $2n \cdot 2^{-n/2}$ of the normalized state corresponding to*

$$\sum_{\boldsymbol{x}\in L} \sqrt{\rho_r(\boldsymbol{x})}|\boldsymbol{x}\rangle = \sum_{\boldsymbol{x}\in L} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle \tag{12}$$

Proof. A quantum state proportional to

$$\sum_{x=-r\sqrt{n}}^{r\sqrt{n}} e^{-\pi(x/(\sqrt{2}r))^2}|x\rangle \tag{13}$$

is created by a technique of Grover and Rudolph [16] which can be done with good precision. This is repeated n times, creating the n-fold tensor product of the state in (13). The resulting state is proportional to

$$\sum_{\boldsymbol{x}\in(-r\sqrt{n},\ldots,r\sqrt{n})^n} \rho_{\sqrt{2}r}(\boldsymbol{x})|x\rangle \tag{14}$$

which Lemma 2.10 shows is at most a trace distance 2^{-n} from the state

$$\sum_{\boldsymbol{x}\in\mathbb{Z}^n} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle \ . \tag{15}$$

Next, using the LLL basis reduction algorithm [22], we efficiently find a basis $\boldsymbol{B}$ for L where the length of its longest vector is at most $2^n\lambda_n$. We let $\mathcal{P}(B)$ denote the fundamental parallelepiped of this basis. Given the state in (15) we compute $\boldsymbol{x} \mod \mathcal{P}(B)$ in a new register and measure this new register to get the result $\boldsymbol{y} \in \mathcal{P}(B)$. This collapses the state to

$$\sum_{\boldsymbol{x}\in L+\boldsymbol{y}} \rho_{\sqrt{2}r}(\boldsymbol{x})|\boldsymbol{x}\rangle$$

and subtracting $\boldsymbol{y}$ from the register gives

$$\sum_{\boldsymbol{x}\in L} \rho_{\sqrt{2}r}(\boldsymbol{x}+\boldsymbol{y})|\boldsymbol{x}\rangle \ . \tag{16}$$

We now show that this state is only a small trace distance from the desired state in equation (12). First, Lemma 2.10 allows us to consider only $\boldsymbol{x}$ with $\|\boldsymbol{x}\| \le \sqrt{n}\cdot r$ in (12) as these states differ by a trace distance of at most 2^{-n}. We denote the weight of $|x\rangle$ by $p_1(\boldsymbol{x}) = \rho_{\sqrt{2}r}(\boldsymbol{x})/\sqrt{\rho_r(L)}$ in the desired state and $p_2(\boldsymbol{x}) = \rho_{\sqrt{2}r}(\boldsymbol{x}+\boldsymbol{y})/\sqrt{\rho_r(L+\boldsymbol{y})}$ in the constructed state. Since the compared states are pure states, the squared trace distance is given by

$$1 - \sum_{x\in L\cap r\sqrt{n}B_n} p_1^2(\boldsymbol{x})p_2^2(\boldsymbol{x})$$

and as such a lower bound on $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ is sufficient to upper bound the trace distance.

A lower bound on $p_2(\boldsymbol{x}) = \rho_{\sqrt{2}r}(\boldsymbol{x}+\boldsymbol{y})/\sqrt{\rho_r(L+\boldsymbol{y})}$ is found by first bounding the square of the denominator. By using Lemma A.1 we have

$$\begin{aligned}\rho_r(L+\boldsymbol{y}) &= \det(L^*)\cdot r^n \sum_{\boldsymbol{z}\in L^*} e^{2\pi i\langle \boldsymbol{z},\boldsymbol{y}\rangle}\rho_{1/r}(\boldsymbol{z}) \\ &\le (1+2^{-2n})\det(L^*)\cdot r^n\end{aligned}$$

where the inequality is given by Lemma 2.2 which is applicable as

$$\lambda_1(L^*) \geq 1/\lambda_n(L) > \sqrt{n}/r \ .$$

Next, we use Claim A.2 to show that the numerator $\rho_{\sqrt{2}r}(\boldsymbol{x}+\boldsymbol{y})$ is at least $(1-2^{-\Omega(n)})\det(L)\rho_{\sqrt{2}r}(x)$. This is the case as $\|\boldsymbol{y}\| \leq \mathrm{diam}(\mathcal{P}(L)) \leq n2^n\lambda_n(L)$ which allows Claim A.2 to be used with $s = \sqrt{2}r$, $t = \|\boldsymbol{x}\| \leq \sqrt{n}r$ and $l = \|\boldsymbol{y}\| \leq n2^n\lambda_n$ in order to give

$$\begin{aligned}\rho_{\sqrt{2}r}(\boldsymbol{x}+\boldsymbol{y}) &\geq \frac{1-\pi(2\cdot n2^n\cdot\sqrt{n}r + n^2 2^{2n})}{(\sqrt{2}r)^2}\rho_{\sqrt{2}r}(\boldsymbol{x}) \\ &\geq (1-\pi n^2 2^{-n})\rho_{\sqrt{2}r}(\boldsymbol{x}) \ .\end{aligned}$$

In total this gives that $p_2^2(\boldsymbol{x})$ is at least

$$(1-\pi n^2 2^{-n})^2(1+2^{-2n})^{-1}\det(L)\nu_r(\boldsymbol{x}) \geq (1-2^{-2n}-2\pi n^2 2^{-n})\det(L)\nu_r(\boldsymbol{x})$$

The same lower bound on the denominator of $p_1(\boldsymbol{x})$ can be can be applied with $\boldsymbol{y}=\boldsymbol{0}$ which gives $p_1^2(\boldsymbol{x}) \geq (1-2^{-2n})\det(L)\nu_r(\boldsymbol{x})$. This gives that

$$\begin{aligned}\sum_{\boldsymbol{x}\in L} p_1^2(\boldsymbol{x})p_2^2(\boldsymbol{x}) &\geq (\det(L)\nu_{\sqrt{2}r}(L))^2(1-2\cdot 2^{-2n}-\pi n^2 2^{-n}) \\ &\geq (1-2\cdot 2^{-2n}-\pi n^2 2^{-n})\end{aligned}$$

where final inequality uses Claim A.3. This gives that the squared trace distance between the desired and constructed distribution is at most

$$2\cdot 2^{-2n} + 2\cdot 2^{-n} + \pi n^2 2^{-n} \leq 4n^2 2^{-n}$$

where $2\cdot 2^{-n}$ comes from applications of Lemma 2.10. Finally the square root of this gives the claimed bound on the trace distance of $2n\cdot 2^{-n/2}$. □

The next statement and its proof is similar to the one in [31] but with an explicit bound on trace distance instead of showing an ℓ_2 distance of $2^{-\Omega(n)}$ [2].

Claim A.11 (Variant of Claim 3.13 from [31])**.** *Let $R \geq 1$ be an integer, L be an n-dimensional lattice satisfying $\lambda_1(L) > 2\sqrt{n}$ and let $\mathcal{P}(L)$ be some basic parallelepiped of L. Then, the trace distance between the normalized quantum states corresponding to*

$$|\vartheta_1\rangle = \sum_{\substack{\boldsymbol{x}\in L/R \\ \|x\|<\sqrt{n}}} \rho(\boldsymbol{x})|\boldsymbol{x} \mod \mathcal{P}(L)\rangle$$

[2]The proof of [31] seems to incorrectly claim that the ℓ_2 norm of $|\vartheta_1\rangle$ is Z, while it actually calculates the squared norm to equal Z. This means that the proof actually shows that $\||\vartheta_1\rangle - |\vartheta_2\rangle\| \leq 2^{-\Omega(n)}\||\vartheta_1\rangle\|^2$ which is not what it claims to show. However, the proof is easily fixed which leads to some differences in the calculations, but without impacting the end result.

and,

$$|\vartheta_2\rangle = \sum_{\boldsymbol{x}\in L/R} \rho(\boldsymbol{x})|\boldsymbol{x} \mod \mathcal{P}(L)\rangle = \sum_{\boldsymbol{x}\in L/R\cap\mathcal{P}(L)} \sum_{\boldsymbol{y}\in L} \rho(\boldsymbol{x}-\boldsymbol{y})|\boldsymbol{x}\rangle$$

is no more than $2\cdot 2^{-n/2}$.

Proof. Let $|\psi\rangle = \alpha|\vartheta_1\rangle$ and $|\phi\rangle = \beta|\vartheta_2\rangle$ be the normalized states that we are interested in where α and β are some normalisation constants. As both $|\psi\rangle$ and $|\phi\rangle$ are pure states, the trace distance between them is simply

$$\sqrt{1-|\langle\psi|\phi\rangle|^2} = \sqrt{1-\alpha^2\beta^2\,|\langle\vartheta_1|\vartheta_2\rangle|^2}\ .$$

Each ket in $|\psi\rangle$ appear only once in the sum as $\lambda_1(L) > 2\sqrt{n}$. This means that the weight assigned to $|\boldsymbol{x} \mod \mathcal{P}(L)\rangle$ in $|\psi\rangle$ is $\alpha\rho(\boldsymbol{x})$. Because of this we see that

$$\frac{1}{\alpha^2} = \|\vartheta_1\|^2 = Z = \sum_{\boldsymbol{x}\in L/R,\|x\|<\sqrt{n}} \rho(\boldsymbol{x})^2 = \rho_{1/\sqrt{2}}(L/R\cap\sqrt{n}B_n)$$

where we use Lemma 2.2 to see that

$$(1-2^{-2n})\rho_{1/\sqrt{2}}(L/R) \le Z \le \rho_{1/\sqrt{2}}(L/R)\ .$$

Furthermore, we have that $|\langle\vartheta_1|\vartheta_2\rangle| = |\langle\vartheta_1|\vartheta_1\rangle| = Z$ as

$$\begin{aligned}|\langle\vartheta_1|\vartheta_2\rangle| &= \sum_{\boldsymbol{x}\in L/R,\|x\|<\sqrt{n}} \sum_{\boldsymbol{y}\in L/R} \rho(\boldsymbol{y})\rho(\boldsymbol{x})\langle\boldsymbol{x} \mod \mathcal{P}(L)|\boldsymbol{y} \mod \mathcal{P}(L)\rangle \\ &= \sum_{\boldsymbol{x}\in L/R,\|x\|<\sqrt{n}} \rho(\boldsymbol{x})^2\langle\boldsymbol{x} \mod \mathcal{P}(L)|\boldsymbol{x} \mod \mathcal{P}(L)\rangle = \langle\vartheta_1|\vartheta_1\rangle\end{aligned}$$

meaning that the trace distance of interest is at most

$$\sqrt{1-\alpha^2\beta^2\langle\vartheta_1|\vartheta_2\rangle} = \sqrt{1-\alpha^2\beta^2 Z^2} = \sqrt{1-\beta^2 Z}$$

and we must therefore bound β.

To bound β, we first bound $\||\vartheta_1\rangle - |\vartheta_2\rangle\|^2$ by

$$\begin{aligned}\||\vartheta_1\rangle - |\vartheta_2\rangle\|^2 &= \sum_{\boldsymbol{x}\in L/R,\|\boldsymbol{x}\|\ge\sqrt{n}} \rho(\boldsymbol{x})^2 = \sum_{\boldsymbol{x}\in L/R,\|\boldsymbol{x}\|\ge\sqrt{n}} \rho_{1/\sqrt{2}}(\boldsymbol{x}) \\ &\le 2^{-2n}\rho_{1/\sqrt{2}}(L/R) \qquad \text{(By Lemma 2.2)}\ .\end{aligned}$$

Combining this bound with $\||\vartheta_1\rangle\| = \sqrt{Z} \le \sqrt{\rho_{1/\sqrt{2}}(L/R)}$ gives

$$\begin{aligned}\frac{1}{\beta^2} = \||\vartheta_2\rangle\|^2 &= \||\vartheta_1\rangle + |\vartheta_2\rangle - |\vartheta_1\rangle\|^2 \le (\||\vartheta_1\rangle\| + \||\vartheta_2\rangle - |\vartheta_1\rangle\|)^2 \\ &\le \left((1+2^{-n})\sqrt{\rho_{1/\sqrt{2}}(L/R)}\right)^2 \le (1+2^{-n})^2\rho_{1/\sqrt{2}}(L/R) \\ &\le (1+3\cdot 2^{-n})\rho_{1/\sqrt{2}}(L/R)\end{aligned}$$

and thus

$$\beta^2 Z \geq \frac{1-2^{-2n}}{1+3\cdot 2^{-n}} \geq 1-3\cdot 2^{-n}-2^{-2n}$$

which finally gives the trace distance as

$$\sqrt{1-\beta^2 Z} \leq \sqrt{3\cdot 2^{-n}+2^{-2n}} \leq 2\cdot 2^{-n/2} \ .$$

□

By using these lemmas, we now show the second part of the iterative step. This consists of an algorithm that, by using an BDD oracle, creates samples from a distribution statistically close to a discrete Gaussian distribution on the lattice. This is detailed in the following lemma which is essentially the same as the one from [31] except for the addition of statistical distances from the actual desired distribution.

Lemma A.12 (Lemma 3.14 from [31])**.** *Let L be a n-dimensional lattice and let $d < \lambda_1(L^*)/2$, $\varepsilon > 0$ be some numbers. Furthermore, let W be an oracle that solves $BDD_{L^*,d}$ with failure probability at most ε. Then, there exists a quantum algorithm that, by using W once, outputs a sample from a distribution with statistical distance less than $\sqrt{2\varepsilon}+3n\cdot 2^{-n/2}$ from $D_{L,\sqrt{n}/(\sqrt{2}d)}$.*

Proof. We assume that $d=\sqrt{n}$ which, with scaling, is without loss of generality. Let $R \geq 2^{3n}\lambda_n(L^*)$ be some large enough integer. By using LLL to bound $\lambda_n(L^*)$ we can choose such an R while guaranteeing that $\log R$ is polynomial in the input size. To begin with, we create a state that is close to

$$\sum_{\boldsymbol{x}\in L^*/R\cap\mathcal{P}(L^*)}\sum_{\boldsymbol{y}\in L^*}\rho(\boldsymbol{x}-\boldsymbol{y})|\boldsymbol{x}\rangle \tag{17}$$

which is a state on $n\log R$ qubits and thus polynomial in the input size. To create this state, we first use Lemma A.10 with $r=1/\sqrt{2}$ to create a state of trace distance no more than $2n\cdot 2^{-n/2}$ from

$$\sum_{\boldsymbol{x}\in L^*/R}\rho(\boldsymbol{x})|\boldsymbol{x}\rangle \ .$$

This state is at most a trace distance 2^{-n} from

$$\sum_{\substack{\boldsymbol{x}\in L^*/R\\ \|\boldsymbol{x}\|<\sqrt{n}}}\rho(\boldsymbol{x})|\boldsymbol{x}\rangle$$

as shown by Lemma 2.10. Next we calculate $\boldsymbol{x} \mod \mathcal{P}(L)$ in a new register and thus have a state that is a trace distance of at most $2n\cdot 2^{-n/2}+2^{-n}$ from

$$\sum_{\substack{\boldsymbol{x}\in L^*/R\\ \|\boldsymbol{x}\|<\sqrt{n}}}\rho(\boldsymbol{x})|\boldsymbol{x}\rangle|\boldsymbol{x} \mod \mathcal{P}(L)\rangle$$

Using the BDD oracle we can recover $\boldsymbol{x}$ from $\boldsymbol{x} \mod \mathcal{P}(L)$, allowing us to reversibly erase the contents of the first register.

The provided oracle has a failure probability of ε, resulting in an additional trace distance from the desired state. This trace distance is bounded by using Lemma 2.11 and an additional register that is in an uniform superposition over all possible randomness states for the BDD oracle. Applying f corresponds to an always correct oracle while $g(x;r)$ corresponds to the provided oracle with failure probability ε. The additional randomness register can be discarded by measuring it, which do not increase the trace distance. As such, the trace distance between a state where we use our BDD oracle and one where an always correct BDD oracle is used is at most $\sqrt{2\varepsilon}$. We have thus constructed a state with trace distance at most $2n \cdot 2^{-n/2} + 2^{-n} + \sqrt{2\varepsilon}$ from

$$\sum_{\boldsymbol{x} \in L^*/R, \|\boldsymbol{x}\| < \sqrt{n}} \rho(\boldsymbol{x}) |\boldsymbol{x} \mod \mathcal{P}(L)\rangle \ . \tag{18}$$

By using Claim A.11 we see that the state in (18) is of trace distance at most $2^{-n/2}$ from

$$\sum_{\boldsymbol{x} \in L^*/R} \rho(\boldsymbol{x}) |\boldsymbol{x} \mod \mathcal{P}(L)\rangle = \sum_{\boldsymbol{x} \in L^*/R \cap \mathcal{P}(L^*)} \sum_{\boldsymbol{y} \in L^*} \rho(\boldsymbol{x} - \boldsymbol{y}) |\boldsymbol{x}\rangle$$

which is the desired state in (17). Thus we have constructed a state that is a trace distance at most $(2n+1)2^{-n/2} + 2^{-n} + \sqrt{2\varepsilon}$ from this desired state.

Next, we let $\boldsymbol{B}$ be a basis for L^* and rewrite the state in (17) as

$$\sum_{\boldsymbol{s} \in Z_R^n} \sum_{\boldsymbol{r} \in Z_n} \rho(\boldsymbol{B}\boldsymbol{s}/R - \boldsymbol{B}\boldsymbol{r}) |\boldsymbol{s}\rangle$$

by using the mapping between $L^*/R \cap \mathcal{P}(L^*)$ and Z_R^n. In this state we apply the quantum Fourier transform on $\mathbb{Z}_R^n$, giving a state proportional to

$$\begin{aligned}
&\sum_{\boldsymbol{t} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{s} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{r} \in \mathbb{Z}^n} \rho(\boldsymbol{B}\boldsymbol{s}/R - \boldsymbol{B}\boldsymbol{r}) \exp(2\pi i \langle \boldsymbol{s}, \boldsymbol{t} \rangle / R) |t\rangle \\
&\quad = \sum_{\boldsymbol{t} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{s} \in \mathbb{Z}^n} \rho(\boldsymbol{B}\boldsymbol{s}/R) \exp(2\pi i \langle \boldsymbol{s}, \boldsymbol{t} \rangle / R) |\boldsymbol{t}\rangle \\
&\quad = \sum_{\boldsymbol{t} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{x} \in L^*/R} \rho(\boldsymbol{x}) \exp(2\pi i \langle B^{-1}\boldsymbol{x}, \boldsymbol{t} \rangle) |\boldsymbol{t}\rangle \\
&\quad = \sum_{\boldsymbol{t} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{x} \in L^*/R} \rho(\boldsymbol{x}) \exp(2\pi i \langle \boldsymbol{x}, (\boldsymbol{B}^{-1})^T \boldsymbol{t} \rangle) |\boldsymbol{t}\rangle \\
&\quad = \det(RL) \sum_{\boldsymbol{t} \in \mathbb{Z}_R^n} \sum_{\boldsymbol{y} \in RL} \rho(\boldsymbol{y} - (\boldsymbol{B}^{-1})^T \boldsymbol{t}) |\boldsymbol{t}\rangle \ .
\end{aligned}$$

The final equality is given by Lemma A.1 and using that if $h(x) = e^{2\pi i \langle \boldsymbol{x}, \boldsymbol{v} \rangle} g(x)$ then its Fourier transform $\hat{h}(w)$ equals $\hat{g}(\boldsymbol{w} - \boldsymbol{v})$. By using that $\boldsymbol{B}$ is a basis for L^*, we identify $(\boldsymbol{B}^{-1})^T \mathbb{Z}_R^n$ with $L \cap \mathcal{P}(RL)$, giving that the state can be written as

$$\sum_{\boldsymbol{x} \in L \cap \mathcal{P}(RL)} \sum_{\boldsymbol{y} \in RL} \rho(\boldsymbol{y} - \boldsymbol{x}) |\boldsymbol{x}\rangle \ .$$

As $\lambda_1(RL) = R\lambda_1(L) \geq R/\lambda_n(L^*) \geq 2^{3n}$ we can apply Claim A.11 showing that this state is of trace distance at most $2^{-n/2}$ from

$$\sum_{\boldsymbol{x} \in L, \|\boldsymbol{x}\| < \sqrt{n}} \rho(\boldsymbol{x}) |\boldsymbol{x} \mod \mathcal{P}(RL)\rangle \ . \tag{19}$$

Measuring this state results in $\boldsymbol{x} \mod \mathcal{P}(RL)$ for some $\boldsymbol{x}$ with $\|\boldsymbol{x}\| < \sqrt{n}$. From this measurement $\boldsymbol{x}$ is recovered efficiently by using Babai's nearest plane [5] as $\lambda_1(RL) \geq 2^{3n}$ and $\boldsymbol{x} \mod \mathcal{P}(RL)$ is at most $\sqrt{n}$ from the lattice RL. We claim that $\boldsymbol{x}$ recovered in this way from the state in (19) results in $\boldsymbol{x}$ that are distributed essentially as $D_{L,1/\sqrt{2}}$. To see this, note that the probability of any $\boldsymbol{x}$ with $\|\boldsymbol{x}\| < \sqrt{n}$ is proportional to $\rho(\boldsymbol{x})^2 = \rho_{1/\sqrt{2}}(\boldsymbol{x})$ as desired. This means that the distance we want to limit is the one between the distributions

$$p_1(\boldsymbol{x}) = \rho_{1/\sqrt{2}}(\boldsymbol{x}) / \rho_{1/\sqrt{2}}(L)$$

and

$$p_2(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\| < \sqrt{n} & \rho_{1/\sqrt{2}}(\boldsymbol{x}) / \rho_{1/\sqrt{2}}(L \cap \sqrt{n} B_n) \\ \|\boldsymbol{x}\| \geq \sqrt{n} & 0 \end{cases} \ .$$

Between these states the statistical distance is

$$\begin{aligned} \sum_{\boldsymbol{x} \in L, \|x\| < \sqrt{n}} \rho_{1/\sqrt{2}}(\boldsymbol{x}) \cdot & \left(\frac{1}{\rho_{1/\sqrt{2}}(L)} - \frac{1}{\rho_{1/\sqrt{2}}(L \cap \sqrt{n} B_n)} \right) \\ & + \sum_{\boldsymbol{x} \in L, \|x\| \geq \sqrt{n}} \rho_{1/\sqrt{2}}(\boldsymbol{x}) / \rho_{1/\sqrt{2}}(L) = \\ & \rho_{1/\sqrt{2}}(L \cap \sqrt{n} B_n) \left(\frac{1}{\rho_{1/\sqrt{2}}(L)} - \frac{1}{\rho_{1/\sqrt{2}}(L \cap \sqrt{n} B_n)} \right) \\ & + \frac{\rho_{1/\sqrt{2}}(L) - \rho_{1/\sqrt{2}}(L \cap \sqrt{n} B_n)}{\rho_{1/\sqrt{2}}(L)} \end{aligned}$$

where both terms are limited via Lemma 2.2 to be no more than 2^{-2n}, meaning that the statistical distance is at most $2 \cdot 2^{-2n}$. Finally, combining all trace and statistical distances via triangle inequality gives that the total statistical distance from the desired distribution is no more than

$$\sqrt{2\varepsilon} + 2 \cdot 2^{-2n} + 2^{-n} + (2n+2) \cdot 2^{-n/2} < 3n \cdot 2^{-n/2} + \sqrt{2\varepsilon}$$

with inequality holding given that $n \geq 3$. □

Now, the iterative step which is repeatedly used in the proof of Theorem 4.3 is stated and proven. Besides the additional tightness information, the lemma is the same as the one presented in [31] and essentially consists of combining Lemmas A.5 and A.12.

Lemma A.13 (Iterative step, Lemma 3.3 from [31])**.** *Let $\varepsilon = \varepsilon(n)$ be a function $\alpha = \alpha(n) \in (0,1)$ be a real number, τ an integer, and $q = q(n) \geq 2$ be an integer. Assume that we have access to an oracle W that solves LWE(Ψ_β, M) except for with probability at most ε for arbitrary $\beta \in [\alpha/\sqrt{2}, \alpha]$. Then there exists an efficient quantum algorithm that, given any n-dimensional lattice L, a number $r > \sqrt{2}q\eta_\varepsilon(L)$, and M samples from $D_{L,r}$, produces a sample of statistical distance at most*

$$\sqrt{18Mn\varepsilon} + 3n \cdot 2^{-n/2}$$

from $D_{L,r\sqrt{n}/(\alpha q)}$ while requiring n calls to W.

Proof. By using Lemma A.5 with W and the M samples that are provided from $D_{L,r}$, we are able to solve $\mathrm{BDD}_{L^*,\alpha q/(\sqrt{2}r)}$. This allows using Lemma A.12 to produce a sample from a distribution that is statistically close to $D_{L,r\sqrt{n}/(\alpha q)}$.

As Lemma A.5 is used as an BDD oracle for Lemma A.12 the output distribution is of statistical distance at most

$$\sqrt{18Mn\varepsilon} + 3n \cdot 2^{-n/2}$$

from $D_{L,r\sqrt{n}/(\alpha q)}$. This is given by the error probability from Lemma A.5 inserted into statistical distance for Lemma A.12. Producing this sample requires only the n calls to W that are required by Lemma A.5 as Lemma A.12 only requires a single call to the BDD oracle. Similarly the number of required samples from $D_{L,r}$ is M as Lemma A.5 is only used a single time. □

In order to use the reduction to solve GIVP, we require that our samples from $D_{L,r}$ contain n linearly independent vectors with high probability. A corollary in [31] shows exactly this, but only specifies that the probability is exponentially close to 1. We require a more precise bound on the probability and therefore redo the proof here. The corollary follows from the following lemma that already contain concrete probabilities in [31] and is therefore directly included here without a proof.

Lemma A.14 (Lemma 3.15 from [31])**.** *Let L be an n-dimensional lattice and let r be such that $r \geq 2\sqrt{n}\eta_\varepsilon(L)$ where $\varepsilon \leq \frac{1}{10}$. Then for any subspace H of dimension at most $n-1$ the probability that $\boldsymbol{x} \notin H$ where $\boldsymbol{x}$ is chosen from $D_{L,r}$ is at least $\frac{1}{10}$*

The corollary of interest easily follows from this lemma.

Corollary A.15 (Corollary 3.16 from [31]). *Let L be an n-dimensional lattice and let r be such that $r \geq \sqrt{2}\eta_\varepsilon(L)$ where $\varepsilon \leq \frac{1}{10}$. Then, the probability that a set of $N > n$ vectors chosen independently at random from $D_{L,r}$ contains no subset of n linearly independent vectors is at most $n(9/10)^{N/n}$.*

Proof. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ be N vectors chosen independently at random from $D_{L,r}$. For $i = 1, \ldots, n$, let B_i be the event that

$$\dim \operatorname{span}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{(i-1)N/n}) = \dim \operatorname{span}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{iN/n}) < n$$

Clearly, if none of the B_i's happen, then $\dim \operatorname{span}(\boldsymbol{x}_1, \ldots \boldsymbol{x}_N) = n$. Hence we are interested in the probability of B_i. For fixed i, we condition on fixed choices of $\boldsymbol{x}_1, \ldots \boldsymbol{x}_{(i-1)N/n}$ such that $\dim \operatorname{span}(\boldsymbol{x}_1, \ldots \boldsymbol{x}_{(i-1)N/n}) < n$. By Lemma A.14 the probability that

$$\boldsymbol{x}_{1+(i-1)N/n}, \ldots \boldsymbol{x}_{iN/n} \in \dim \operatorname{span}(\boldsymbol{x}_1, \ldots \boldsymbol{x}_{(i-1)N/n})$$

is at most $(9/10)^{N/n}$. As such, the probability that a specific B_i happens is at most $(9/10)^{N/n}$ and the probability that none of the B_i happens is therefore at least $1 - n(9/10)^{N/n}$. As such, except for with probability at most $n(9/10)^{N/n}$, there is a subset of n linearly independent vectors among the N sampled vectors. □