

# Verifying Solutions to LWE with Implications for Concrete Security

Palash Sarkar and Subhadip Singha  
Applied Statistics Unit  
Indian Statistical Institute  
203, B.T.Road, Kolkata, India - 700108.  
{palash, subha.r}@isical.ac.in

November 8, 2019

## Abstract

A key step in Regev's (2009) reduction of the Discrete Gaussian Sampling (DGS) problem to that of solving the Learning With Errors (LWE) problem is a statistical test required for verifying possible solutions to the LWE problem. In this work, we work out a concrete lower bound on the success probability and its effect in determining an upper bound on the tightness gap of the reduction. The success probability is determined by the value of the rejection threshold  $t$  of the statistical test. Using a particular value of  $t$ , Regev showed that asymptotically, the success probability of the test is exponentially close to one for all values of the LWE error  $\alpha \in (0, 1)$ . From the concrete analysis point of view, the value of the rejection threshold used by Regev is sub-optimal. It leads to considering the lattice dimension to be as high as 400000 to obtain somewhat meaningful tightness gap. We show that by using a different value of the rejection threshold and considering  $\alpha$  to be at most  $1/\sqrt{n}$  results in the success probability going to 1 for small values of the lattice dimension. Consequently, our work shows that it may be required to modify values of parameters used in an asymptotic analysis to obtain much improved and meaningful concrete security.

**Keywords:** lattices, learning with errors, discrete gaussian sampling, statistical test, concrete analysis.

**Mathematics Subject Classification: Primary:** 94A60

# 1 Introduction

In a seminal work, Regev [13] introduced the learning with errors (LWE) problem and highlighted its role in lattice-based cryptography. The major achievement of the work was to show a reduction from a worst case lattice problem to the breaking of a cryptosystem. This worst-case to average-case reduction has been later claimed to be a major theoretical advantage of cryptosystems based on lattices. The entire analysis in [13] was done in an asymptotic setting where the lattice dimension  $n$  is allowed to go to infinity.

Suppose  $\mathcal{A}$  is an algorithm which given access to an oracle  $\mathcal{O}$  solves a problem  $\mathcal{P}$  in time  $T$  with success probability  $P_S$ . Further, suppose  $\mathcal{O}$  takes time  $T'$  and has success probability  $P'_S$ . Then the tightness gap of the algorithm  $\mathcal{A}$  is  $(T \cdot P'_S)/(T' \cdot P_S)$ . The reduction is said to be tight if the tightness gap is 1 (or small) and is said to be loose if the tightness gap is ‘large’.

A later work [8] performed a concrete analysis of the reduction in [13] to determine whether it can be used to choose parameters for practical cryptosystems. This required determining the tightness gap of the reduction in concrete terms as a function of  $n$ . It turned out that the tightness gap can indeed be very large casting doubt on the practical usefulness of the reduction in [13]. For example, for  $n = 1024$ , it was argued that the tightness gap is about  $2^{504}$  and so the worst-case to average-case reduction in [13] cannot be used to argue about the security of cryptosystems with lattice dimension  $n = 1024$ .

The reduction in [13] is a cascade of three smaller reductions. The first reduction is from the Smallest Independent Vector Problem (SIVP) to the problem of Discrete Gaussing Sampling (DGS). The second reduction is from DGS to (search) LWE while the third reduction is from search-LWE to average case decisional LWE (DLWE<sub>ac</sub>). There is a further reduction from DLWE<sub>ac</sub> to that of breaking the cryptosystem. We ignore this reduction, since this depends on the actual cryptosystem.

The second reduction, i.e., the one from DGS to LWE is the main contribution of [13]. A key step in this reduction consists of verifying solutions to LWE. This verification is done using a statistical test. It has been proved in [13] that asymptotically the success probability of the statistical test is exponentially close to one. The statistical test is used many times in the entire reduction and the success probability of the statistical test determines the overall success probability of the complete reduction. We take a close look at the success probability of the statistical test. Using the standard Hoeffding inequality, we determine an upper bound on the error of the statistical test. This in turn leads to a lower bound on the success probability of the test and then to a lower bound  $P_S$  on the success probability of the entire reduction.

The success probability of the test depends on the value of the rejection threshold in the statistical test. The proof of Lemma 3.6 of [13] sets the rejection threshold  $t$  to be equal to 0.02. This value of  $t$  is sufficient to show that asymptotically the success probability goes to one for all values of the LWE error  $\alpha \in (0, 1)$ . For concrete security, however,  $t = 0.02$  is sub-optimal. Using this value of  $t$  requires considering lattice dimensions as high as 400000 to ensure even somewhat meaningful tightness gap. In an alternative analysis, we consider the rejection threshold to be mid-way between the means of the two distributions under consideration. Using this value of the rejection threshold and considering  $\alpha$  to be at most  $1/\sqrt{n}$  we show that the success probability becomes close to one for reasonable value of  $n$ . Consequently, under this alternative analysis, the success probability of the statistical test has no effect on the overall tightness gap of the reduction.

The ring-LWE (RLWE) problem was later considered in [11]. This work showed a worst-case to average-case reduction which is analogous to the reduction in [13]. The reduction in [11] refers to the verification lemma used by Regev [13]. Regev’s reduction has a quantum component. A follow up work by Peikert [12] showed how to remove the quantum component. This was achieved at the cost of using an exponential sized modulus. Though not explicitly mentioned, Peikert’s work also requires the statistical test to verify LWE solutions. A later work by Brakerski et al. [7], built upon Peikert’s work to show classical hardness of LWE with polynomial sized modulus. Again, the verify LWE statistical test is implicitly used in this work. So, our concrete analysis of the success probability of the statistical test to verify LWE solutions should also apply to the reductions in [11, 12, 7].

The importance of LWE in the context of lattice based cryptography is underscored by the fact that a number

of submissions made to the ongoing NIST process for selecting a new public key standard base their security on the LWE problem and several of its variants. LWE based proposals which are in the second round of the NIST process are Frodo [2], Kyber [3], LAC [10], NewHope [1], Round5 [4] and Saber [9]. A recent work by Bernstein [5] performs a comparative study of the provable security of these and other lattice based proposals. While commenting on the tightness of reduction, Bernstein [5] comments that “the loss of tightness is gigantic” and credits [8] for pointing this out. For parameters of practical interest, our work slightly updates the tightness bound given in [8].

## 2 Preliminaries

A full rank lattice  $L$  in  $\mathbb{R}^n$  is the set of all integer linear combinations of  $n$  linearly independent vectors  $\mathbf{v}_1, \dots, \mathbf{v}_n$  in  $\mathbb{R}^n$ . Following Regev [13],  $L$  will also denote the  $n \times n$  matrix whose columns are  $\mathbf{v}_1, \dots, \mathbf{v}_n$ . So, given  $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{Z}^n$ ,  $L\mathbf{a}$  denotes the lattice point  $\mathbf{v} = a_1\mathbf{v}_1 + \dots + a_n\mathbf{v}_n$  and  $L^{-1}(\mathbf{v})$  denotes the integer coefficient vector  $\mathbf{a}$  corresponding to  $\mathbf{v}$ .

The length of a vector is its Euclidean norm. For  $i \in \{1, \dots, n\}$ ,  $\lambda_i(L)$  is the least real number  $r$  such that  $L$  has  $i$  linearly independent vectors with the longest having length  $r$ . The dual of a lattice  $L$  is denoted as  $L^*$  and is defined to be the set of all vectors  $\mathbf{y} \in \mathbb{R}^n$  such that  $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}$  for all  $\mathbf{x} \in L$ .

The normal distribution with mean  $\mu$  and standard deviation  $\sigma$  will be denoted as  $\mathcal{N}(\mu, \sigma)$ . For  $\alpha \in (0, 1)$ ,  $\Psi_\alpha$  is the probability distribution obtained by sampling from  $\mathcal{N}(0, \alpha/\sqrt{2\pi})$  and reducing the result modulo 1.

Let  $p \geq 2$  be an integer. Let  $\chi$  be a probability distribution on  $\mathbb{Z}_p$ . Let  $n$  be a positive integer and fix  $\mathbf{s} \in \mathbb{Z}_p^n$ . The distribution  $A_{\mathbf{s}, \chi}$  on  $\mathbb{Z}_p^n \times \mathbb{Z}_p$  is defined as follows. Choose  $\mathbf{a}$  uniformly at random from  $\mathbb{Z}_p^n$ ;  $e$  from  $\mathbb{Z}$  following  $\chi$  and output  $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ . Let  $\phi$  be a probability density function on  $\mathbb{T} = (0, 1)$ . The distribution  $A_{\mathbf{s}, \phi}$  is defined as follows. Choose  $\mathbf{a}$  uniformly at random from  $\mathbb{Z}_p^n$ ;  $e$  from  $\mathbb{T}$  following  $\phi$  and output  $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle / p + e)$ , where the addition is performed modulo 1.

Fix a positive integer  $n$  and an integer  $p \geq 2$ . The learning with errors problem  $\text{LWE}_{p, \chi}$  is the following. For any  $\mathbf{s} \in \mathbb{Z}_p^n$ , given samples from  $A_{\mathbf{s}, \chi}$ , it is required to output  $\mathbf{s}$ . Similarly, for a probability density function  $\phi$  on  $\mathbb{T}$ , the  $\text{LWE}_{p, \phi}$  problem is the following. For any  $\mathbf{s} \in \mathbb{Z}_p^n$ , given samples from  $A_{\mathbf{s}, \phi}$ , it is required to output  $\mathbf{s}$ .

For  $\mathbf{x} \in \mathbb{R}^n$  and  $s > 0$ , define  $\rho_s(\mathbf{x}) = \exp(-\pi\|\mathbf{x}\|^2/s^2)$ . For a lattice  $L$ , define  $\rho_s(L) = \sum_{\mathbf{x} \in L} \rho_s(\mathbf{x})$ . The discrete Gaussian distribution  $D_{L, s}$  on a lattice  $L$  assigns to a vector  $\mathbf{v} \in L$  the probability  $D_{L, s}(\mathbf{v}) = \rho_s(\mathbf{v})/\rho_s(L)$ . Given a lattice  $L$  and a real number  $r$ , the discrete Gaussian sampling problem  $\text{DGS}_{L, r}$  is to obtain a sample from  $D_{L, r}$ .

For a lattice  $L$  and a real number  $\epsilon > 0$ , the smoothing parameter  $\eta_\epsilon(L)$  is the smallest  $s$  such that  $\rho_{1/s}(L^* \setminus \{0\}) \leq \epsilon$ .

## 3 From DGS to LWE

The following is a restatement of Theorem 3.1 of [13] which is the main result of [13].

**Theorem 1.** *Let  $L$  be a lattice of dimension  $n$ ,  $p \geq 2$  be an integer and  $\alpha \in (0, 1)$  be a real number. Given an oracle for  $\text{LWE}_{p, \Psi_\alpha}$ , it is possible to sample from  $D_{L, r}$  where  $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$ ,  $\alpha p > 2\sqrt{n}$ .*

For  $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$ , define  $r_i = r \cdot (\alpha p/\sqrt{n})^i$  for  $i = 1, \dots, 3n$ .

The proof of the theorem is provided in [13] as a sequence of nested oracle calls. In the following, we rewrite the oracle calls and the other computations required for the proof in [13] in an algorithmic form. The required subroutines and data structures are as follows. The quantity  $c$  is a constant such that the LWE oracle is provided with  $n^c$  samples.

**solveLWE $_{p,\Psi_\alpha}(\mathcal{I})$ :** This is the oracle to solve  $\text{LWE}_{p,\Psi_\alpha}$ . The list  $\mathcal{I}$  consists of  $n^c$  samples from  $A_{\mathbf{s},\Psi_\beta}$  for some  $0 < \beta \leq \alpha$ . Note that the oracle is guaranteed to work correctly if  $\beta = \alpha$ , otherwise it might return an incorrect result.

**verifyLWE( $\mathbf{s}', \mathcal{I}$ ):** The input  $\mathcal{I}$  contains  $n^c$  samples from  $A_{\mathbf{s},\Psi_\beta}$ . This algorithm returns **true** if  $\mathbf{s} = \mathbf{s}'$ , otherwise it returns **false**.

**solveCVP $^{(p)}(L^*, \mathcal{L}, \mathbf{z})$ :** Here  $L^*$  is the dual lattice of  $L$ ;  $\mathcal{L}$  contains  $n^c$  samples from  $D_{L,r_i}$  for some  $i \in \{1, \dots, 3n\}$ ;  $\mathbf{z}$  is within distance  $\lambda_1(L)/2$  of  $L^*$ . Returns the coefficient vector modulo  $p$  of the vector in  $L^*$  which is closest to  $\mathbf{z}$ .

**solveCVP( $L^*, \mathcal{L}, \mathbf{z}$ ):** The inputs  $L^*, \mathcal{L}$  and  $\mathbf{z}$  are as in the case of  $\text{solveCVP}^{(p)}$ . Returns a point of  $L^*$  which is closest to  $\mathbf{z}$ .

**quantumSample():** Uses  $\text{solveCVP}(L^*, \mathcal{L}, \cdot)$  as an oracle and some quantum computation to return a sample from  $D_{L,r_{i-1}}$ . The list  $\mathcal{L}$  contains  $n^c$  samples from  $D_{L,r_i}$ .

**solveDGS( $p, \alpha, r$ ):** Uses the oracle  $\text{solveLWE}_{p,\Psi_\alpha}(\cdot)$  to return a sample from  $D_{L,r}$  where  $r \geq \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$ . Note that the description of the algorithm  $\text{solveDGS}$  provides the proof of Theorem 1.

In the algorithm descriptions, we will make use of the following two subroutines mentioned below. We will not be needing the details of these procedures and so we do not describe these details. They can be found in [13].

1. **bootstrap( $L, r$ ):** Here  $L$  is a lattice and  $r > \sqrt{2n} \cdot \eta_\epsilon(L)/\alpha$ . Returns a list  $\mathcal{L}$  containing  $n^c$  independent samples from  $D_{L,r_{3n}}$  where  $r_{3n} = r \cdot ((\alpha p)/(\sqrt{n}))^{3n}$ .
2. **reconstruct( $x$ ):** This is used in  $\text{solveCVP}$  to reconstruct the closest vector by first applying a nearest neighbour algorithm and then retracing through the results returned by the repeated calls to  $\text{solveCVP}^{(p)}$ .

---

**Algorithm 1** Algorithm to solve DGS using an LWE oracle.

---

```

1: function solveDGS( $p, \alpha$ )
2:    $\mathcal{L} \leftarrow \text{bootstrap}(L, r)$ ;
3:   for  $i \leftarrow 3n$  down to 1 do
4:      $\mathcal{L}' \leftarrow \{\}$ ;
5:     for  $j \leftarrow 1$  to  $n^c$  do
6:        $\mathbf{y} \leftarrow \text{quantumSample}()$  (using  $\text{solveCVP}(L^*, \mathcal{L}, \cdot)$  as an oracle);
7:        $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\mathbf{y}\}$ ;
8:     end for
9:      $\mathcal{L} \leftarrow \mathcal{L}'$ ;  $r_{i-1} = r_i \cdot (\sqrt{n})/(\alpha p)$ ;
10:  end for
11:  return one element from  $\mathcal{L}$ .
12: end function.
```

---

## 4 Concrete Analysis

The number of times the oracle  $\text{solveLWE}$  is called is determined by the following factors.

---

**Algorithm 2** Algorithm to solve CVP.

---

```

1: function solveCVP( $L^*, \mathcal{L}, z$ )
2:    $\mathbf{z}_1 \leftarrow \mathbf{z}$ ;
3:   for  $k \leftarrow 1$  to  $n$  do
4:      $\mathbf{a}_k \leftarrow \text{solveCVP}^{(p)}(L^*, \mathcal{L}, \mathbf{z}_k)$ ;
5:      $\mathbf{z}_{k+1} \leftarrow (\mathbf{z}_k - L^* \mathbf{a}_k)/p$ ;
6:   end for
7:    $\mathbf{s} \leftarrow \text{reconstruct}(\mathbf{z}_{n+1})$ ;
8:   return  $\mathbf{s}$ .
9: end function.

```

---

**Algorithm 3** Algorithm to solve  $\text{CVP}^{(p)}$ .

---

```

1: function solveCVP(p)( $L^*, \mathcal{L}, \mathbf{z}$ )
2:    $Z \leftarrow$  set of all integer multiples of  $n^{-2c}\alpha^2$  in the range  $(0, \alpha^2]$ ;
3:    $\mathcal{I} \leftarrow \{\}$ ;
4:   for  $\mathbf{v}$  in  $\mathcal{L}$  do
5:      $\mathbf{a} \leftarrow L^{-1}\mathbf{v} \bmod p$ ;
6:      $e \xleftarrow{\$} \mathcal{N}(0, \alpha/(2\sqrt{\pi}))$ ;
7:      $\mathcal{I} \leftarrow \mathcal{I} \cup \{(\mathbf{a}, \langle \mathbf{z}, \mathbf{v} \rangle/p + e \bmod 1)\}$ ;
8:   end for
9:   for  $\gamma$  in  $Z$  do
10:     $\mathcal{I}' \leftarrow \{\}$ ;
11:    for  $i \leftarrow 1$  to  $n$  do
12:      for  $(\mathbf{a}, e) \in \mathcal{I}$  do
13:         $\varepsilon \xleftarrow{\$} \Psi_{\sqrt{\gamma}}$ ;
14:         $\mathcal{I}' \leftarrow \mathcal{I}' \cup \{(\mathbf{a}, e + \varepsilon)\}$ ;
15:      end for
16:       $\mathbf{s}' \leftarrow \text{solveLWE}_{p, \Psi_\alpha}(\mathcal{I}')$ ;
17:      if  $\text{verifyLWE}(\mathbf{s}', \mathcal{I}')$  returns true then
18:        return  $\mathbf{s}'$ ;
19:      end if
20:    end for
21:   end for
22: end function.

```

---

1. The loop in solveDGS has  $3n$  iterations. In the  $i$ -th iteration  $n^c$  samples of  $D_{L, r_i}$  are used to generate  $n^c$  samples of  $D_{L, r_{i-1}}$ . Generating each sample of  $D_{L, r_{i-1}}$  requires a call to `quantumSample` which in turn generates a call to `solveCVP`. So, the subroutines `quantumSample` and `solveCVP` are both called a total of  $3n \cdot n^c = 3n^{c+1}$  times.
2. The loop in `solveCVP` has  $n$  iterations and in each iteration, a call to `solveCVP(p)` is made. So, each call to `solveCVP` generates  $n$  calls to `solveCVP(p)`.
3. In `solveCVP(p)`, the set  $Z$  contains about  $n^{2c}$  values. So, the loop from Steps 9 to 21 makes about  $n^{2c+1}$  calls to `solveLWE` and to `verifyLWE`. So, each call to `solveCVP(p)` generates  $n^{2c+1}$  calls to `solveLWE` and to `verifyLWE`.

---

**Algorithm 4** Algorithm to verify an LWE solution.

---

```

1: function verifyLWE( $s', \mathcal{I}'$ )
2:   Let  $m \leq n^c$  be a positive integer;
3:   Choose  $m$  pairs  $(\mathbf{a}_1, x_1), \dots, (\mathbf{a}_m, x_m)$  from  $\mathcal{I}'$ ;
4:    $w \leftarrow 0$ ;
5:   for  $i \leftarrow 1$  to  $m$  do
6:      $y_i \leftarrow x_i - \langle \mathbf{a}_i, s' \rangle / p$ ;
7:      $w \leftarrow w + \cos(2\pi y_i)$ ;
8:   end for
9:    $z \leftarrow w/m$ ;
10:  Let  $t \in (0, 1)$ ;
11:  if  $z > t$  then
12:    return true
13:  else
14:    return false
15:  end if
16: end function.
```

---

**Proposition 2.** *Algorithm solveDGS has the following properties.*

1. *The solveLWE oracle is called  $T = 3n^{3c+3}$  times.*
2. *Algorithm verifyLWE is also called  $T$  times.*
3. *A total of  $3n^{c+1}$  quantum computations are required.*

**Remark:** Regev [13] has shown that each quantum computation is on a state of  $n \log R$  qubits where  $R \geq 2^{3n} \lambda_n(L^*)$  is an integer. For  $c = 1$  the number of quantum computations required is  $3n^2$  where each computation is on at least  $(3n^2 + n \log \lambda_n(L^*))$  bits. The cost of quantum computation increases quadratically with  $n$ .

The VerifyLWE algorithm is based on the proof of Lemma 3.6 of [13]. We highlight two aspects of VerifyLWE that are not present in this proof.

1. The parameter  $m$  is not present in the proof. The proof starts by considering  $n$  samples. This is achieved by setting  $m = n$  in VerifyLWE. Note that the set  $\mathcal{I}'$  has cardinality  $n^c$  and so  $m$  can be at most  $n^c$ .
2. The parameter  $t$  is not present in the proof. The proof considers the rejection threshold to be 0.02. This is achieved by setting  $t = 0.02$  in VerifyLWE.

The choices of  $m = n$  and  $t = 0.02$  are sufficient for asymptotic analysis. We show later that these choices are sub-optimal for concrete analysis.

Algorithm verifyLWE is essentially a test of hypothesis. In verifyLWE, the pairs in  $\mathcal{I}'$  are of the form  $(a, \langle a, s \rangle / p + e)$  where  $e$  follows  $\Psi_\beta$ . The test statistic is the variable  $z$ . Let  $\xi_0$  be the distribution of  $z$  when  $s = s'$  and let  $\mu_0$  be the corresponding mean of  $z$ ; let  $\xi_1$  be the distribution of  $z$  when  $s \neq s'$  and let  $\mu_1$  be the corresponding mean of  $z$ . The following have been proved by Regev [13].

- $\xi_0 = \Psi_\alpha$  so that  $\mu_0 = \exp(-\pi\alpha^2) \geq 0.04$  for  $\alpha < 1$ . Note that  $\mu_0 > t = 0.02$ .
- $\mu_1 = 0$ .

The computation performed by `verifyLWE` is a test of hypothesis between  $H_0 : s = s'$  versus  $H_1 : s \neq s'$ .

Two types of errors are to be considered.

$$e_0 = \Pr[\text{Type-1 error}] = \Pr[\text{reject } H_0 \text{ when it is true}] = \Pr_{z \sim \xi_0} [z \leq t]; \quad (1)$$

$$e_1 = \Pr[\text{Type-2 error}] = \Pr[\text{accept } H_0 \text{ when it is false}] = \Pr_{z \sim \xi_1} [z > t]. \quad (2)$$

A Type-1 error will result in the correct value of  $s'$  being rejected and so the entire reduction will not succeed. A Type-2 error will result in an incorrect value of  $s'$  being accepted. This incorrect value of  $s'$  will be passed on to `verifyCVP(p)` and then on to `verifyCVP` resulting in an incorrect solution to the CVP problem. So, again, the entire reduction will fail. So, it is required to ensure that both Type-1 and Type-2 errors are small.

For  $i = 1, \dots, m$ , let  $v_i = \cos(2\pi y_i)$ . Then  $v_1, \dots, v_m$  take values in the interval  $[-1, 1]$ . Applying the Hoeffding inequality (see Appendix A) to  $v_1, \dots, v_m$  and  $z = (v_1 + \dots + v_m)/m$ , provides the following upper bounds on  $e_0$  and  $e_1$ .

$$e_0 = \Pr_{z \sim \xi_0} [z \leq t] = \Pr_{z \sim \xi_0} [z - \mu_0 \leq -(\mu_0 - t)] \leq \exp(-m(\mu_0 - t)^2/2); \quad (3)$$

$$e_1 = \Pr_{z \sim \xi_1} [z > t] = \Pr_{z \sim \xi_1} [z - \mu_1 > t - \mu_1] \leq \exp(-mt^2/2). \quad (4)$$

If  $s' = s$ , then the probability that `verifyLWE` makes an error is at most  $e_0$ ; if  $s' \neq s$ , then the probability that `verifyLWE` makes an error is at most  $e_1$ . So, the probability that `verifyLWE` makes an error is at most  $\max(e_0, e_1)$  and so the probability that `verifyLWE` is successful is at least  $(1 - \max(e_0, e_1)) = (1 - \max(\exp(-m(\mu_0 - t)^2/2), \exp(-mt^2/2)))$ .

Proposition 2 shows that `verifyLWE` is called a total of  $3n^{3c+3}$  times by `solveDGS`. The probability that all of these calls are successful is at least

$$P_S = (1 - \max(\exp(-m(\mu_0 - t)^2/2), \exp(-mt^2/2)))^{3n^{3c+3}}. \quad (5)$$

Again from Proposition 2, the number of calls to `solveLWE` made by `solveDGS` is  $T = 3n^{3c+3}$  and so the tightness gap of the reduction from DGS to LWE is at most

$$G = T/P_S = 3n^{3c+3} \cdot (1 - \max(\exp(-m(\mu_0 - t)^2/2), \exp(-mt^2/2)))^{-3n^{3c+3}}. \quad (6)$$

## 4.1 Numerical Results

To compute numerical values, we need to specify the values of the parameters  $m$  and  $t$ . We consider two scenarios. The first scenario corresponds to the values actually used in the proof of Lemma 3.6 of [13]. The second scenario corresponds to an alternative analysis where we change the value of the rejection threshold and consider values of  $\alpha$  which occur in practice.

**Concrete analysis of the proof Lemma 3.6 of [13].** As mentioned earlier, the proof of Lemma 3.6 of [13] corresponds to setting  $m = n$  and  $t = 0.02$ . Using  $m = n$  and  $t = 0.02$ , (6) simplifies to the following.

$$G = T/P_S = 3n^{3c+3} \cdot (1 - \exp(-n/5000))^{-3n^{3c+3}}. \quad (7)$$

Setting  $c = 1$  (which minimises the right hand side of (7)), we have evaluated  $T$ ,  $P_S$  and  $G$  for various values of  $n$ . It turns out that for  $n \leq 350000$ ,  $G$  is determined primarily by  $1/P_S$  while for  $n \geq 400000$ , the value of  $1/P_S$  becomes negligible. For  $2 \leq n \leq 350000$ , the value of  $G$  remains very high, for example for  $n = 350000$ ,  $\log_2(\log_2(G)) \approx 11.68$ . For  $n = 400000$ ,  $\log_2(\log_2(G)) \approx 6.83$ . The parameter  $n$  is the dimension of the underlying lattice. So, if the lattice dimension is to be chosen based on the proof of Lemma 3.6 of [13], then the value of  $n$  has to be at least 400000.

**Alternative concrete analysis.** The statistical test performed by `verifyLWE` is essentially a test for the means  $\mu_0$  and  $\mu_1 = 0$  of the distributions  $\xi_0$  and  $\xi_1$  respectively. A natural value of the rejection threshold  $t$  is the choice  $\mu_0/2 = \exp(-\pi\alpha^2)/2$ . This makes  $G$  depend on the value of  $\alpha$ . A higher value of  $\alpha$  makes the LWE problem more difficult but, also results in a worse tightness gap in the reduction from DGS to LWE. On the other hand, most practical cryptosystems<sup>1</sup> consider  $\alpha$  to be at most  $1/\sqrt{n}$ . To account for  $\alpha \in (0, 1/\sqrt{n})$ , one may set  $t = \exp(-\pi/n)/2$ . Using  $t = \exp(-\pi/n)/2$  gives the following expression for  $G$ .

$$G = T/P_S = 3n^{3c+3} \cdot (1 - \exp(-m \exp(-2\pi/n)/8))^{-3n^{3c+3}}. \quad (8)$$

As in the previous concrete analysis, setting  $c = 1$  and  $m = n$ , we have computed the values of  $T$ ,  $P_S$  and  $G$  for various values of  $n$ . In this case, we observe that the value of  $P_S$  becomes very close to 1 for values of  $n$  as small as 100. So, under this alternative concrete analysis, the value of  $G$  is determined entirely by  $T$  and is equal to  $3n^6$  for most practical values of  $n$ .

To summarise, the value of the rejection threshold  $t$  plays an important role in the concrete analysis. If the value of  $t$  is set to be equal to 0.02 as in the proof of Lemma 3.6 of [13], then extremely high lattice dimensions are required for even somewhat meaningful tightness gap. On the other hand, choosing the rejection threshold to be mid-way between the means of the two distribution and considering  $\alpha$  to be at most  $1/\sqrt{n}$ , there is no noticeable effect of  $P_S$  on the tightness gap for reasonable values of  $n$ . In this case, the tightness gap is given entirely by the number of oracle calls  $T$ .

## 4.2 Comparison to Previous Concrete Analysis

The complete reduction by Regev [13] is from worst case SIVP to average case decisional LWE problem ( $\text{DLWE}_{ac}$ ). This reduction consists of three parts.

- SIVP to DGS with tightness gap  $2n^3$  [8].
- DGS to LWE with tightness gap  $G$  given by (6). In the previous concrete analysis [8], the success probability of the statistical test was not considered and the value of  $T$  for the reduction from DGS to LWE was obtained as  $3n^{c+3}$ . The reason for obtaining a lower value of  $T$  is that the loop over  $\gamma$  from Steps 9 to 21 in Algorithm `solveCVP(p)` was missed. So, the number of oracle calls to `solveLWE` for solving `CVP(p)` was considered to be  $n$  instead of  $n^{2c+1}$ .
- LWE to  $\text{DLWE}_{ac}$  with tightness gap  $np \cdot n^{d_1+2d_2+2}$  [8] for positive integers  $d_1$  and  $d_2$ .

Our work does not change the tightness gaps of the first and the third reductions. The main part of the entire reduction is the second reduction. For this case, we have incorporated into the tightness gap the success probability of the statistical test required by the algorithm for verifying LWE solutions. This was not considered in [8]. The above concrete analysis shows that by appropriately setting the value of the rejection threshold and using values of  $\alpha$  that are used in practice, there is no noticeable effect of the success probability on the tightness gap. Ignoring the effect of success probability, the only change of the concrete analysis with respect to [8] is in revising the number of oracle calls as mentioned above. In [8], it was argued that for  $n = 1024$ , the tightness gap of the reduction from SIVP to  $\text{DLWE}_{ac}$  is about  $2^{504}$ . Incorporating the revised number of oracle calls mentioned above, the tightness gap becomes  $2^{524}$ .

## 5 Conclusion

We have incorporated the success probability of the statistical test for verifying LWE solutions into an upper bound  $G$  on the tightness gap. If the rejection threshold is chosen as in the proof of Lemma 3.6 of [13], then  $G$

<sup>1</sup>This was mentioned by Chris Peikert in an email.



is very high for values of  $n$  as large as 350000. On the other hand, we show that by choosing a different value of the rejection threshold and considering  $\alpha$  to be at most  $1/\sqrt{n}$ , there is no noticeable effect of the success probability on the value of  $G$  for reasonable values of  $n$ . Consequently, our work highlights the sensitivity of concrete security analysis to values of underlying parameters.

## Acknowledgement

We are grateful to Alfred Menezes for providing comments on the work and in particular for pointing out an error in the computational results in a previous version of the paper. Thanks to Chris Peikert for mentioning that for practical LWE based cryptosystems  $\alpha$  is at most  $1/\sqrt{n}$ . He also suggested that for sensible values of parameters, the success probability of VerifyLWE can be made close to one.

## References

- [1] Erdem Alkim, Roberto Avanzi, Joppe Bos, Leo Ducas, Antonio de la Piedra, Thomas Poppelmann, Peter Schwabe, Douglas Stebila, Martin R. Albrecht, Emmanuela Orsini, Valery Osheter, Kenneth G. Paterson, Guy Peer, and Nigel P. Smart. NewHope: algorithm specifications and supporting documentation. <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2019.
- [2] Erdem Alkim, Joppe Bos, Leo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM: Learning With Errors key encapsulation. <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2019.
- [3] Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber: algorithm specifications and supporting documentation. <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2009.
- [4] Hayo Baan, Sauvik Bhattacharya, Scott Fluhrer, Oscar Garcia-Morchon, Thijs Laarhoven, Rachel Player, Ronald Rietman, Markku-Juhani O. Saarinen, Ludo Tolhuizen, Jose-Luis Torre-Arce, and Zhenfei Zhang. Round5: KEM and PKE based on (Ring) Learning With Rounding. <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2019.
- [5] Daniel J. Bernstein. Comparing proofs of security for lattice-based encryption. Cryptology ePrint Archive, Report 2019/691, 2019. <https://eprint.iacr.org/2019/691>.
- [6] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).
- [7] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC’13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.
- [8] Sanjit Chatterjee, Neal Koblitz, Alfred Menezes, and Palash Sarkar. Another look at tightness II: practical issues in cryptography. In Raphael C.-W. Phan and Moti Yung, editors, *Paradigms in Cryptology - Mycrypt 2016. Malicious and Exploratory Cryptology - Second International Conference, Mycrypt 2016, Kuala Lumpur, Malaysia, December 1-2, 2016, Revised Selected Papers*, volume 10311 of *Lecture Notes in Computer Science*, pages 21–55. Springer, 2016.

- [9] Jan-Pieter DANvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER: Mod-LWR based KEM (round 2 submission). <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2019.
- [10] Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, Zhenfei Zhang, Zhe Liu, Hao Yang, Bao Li, and Kunpeng Wang. LAC: Lattice-based Cryptosystems. <https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions>, 2019.
- [11] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013.
- [12] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009.
- [13] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
- [14] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 7.5.1)*, 2019. <https://www.sagemath.org>.

## A Hoeffding Inequality

We briefly recall Hoeffding’s inequality for the sum of independent random variables.

**Theorem 3** (Hoeffding Inequality). *Let,  $X_1, X_2, \dots, X_\lambda$  be a finite sequence of independent random variables, such that for all  $i = 1, \dots, \lambda$ , there exists real numbers  $a_i, b_i \in \mathbb{R}$ , with  $a_i < b_i$  and  $a_i \leq X_i \leq b_i$ . Let  $X = \sum_{i=1}^\lambda X_i$ . Then for any positive  $t > 0$ ,*

$$\Pr[X - E[X] \geq t] \leq \exp\left(-\frac{2t^2}{D_\lambda}\right) \quad \text{and} \quad \Pr[X - E[X] \leq -t] \leq \exp\left(-\frac{2t^2}{D_\lambda}\right); \quad (9)$$

where  $D_\lambda = \sum_{i=1}^\lambda (b_i - a_i)^2$ .