Rounded Gaussians

Fast and Secure Constant-Time Sampling for Lattice-Based Crypto

Andreas Hülsing, Tanja Lange, Kit Smeets

Department of Mathematics and Computer Science
Technische Universiteit Eindhoven, P.O. Box 513, 5600 MB Eindhoven, NL
andreas@huelsing.net, tanja@hyperelliptic.org, c.j.c.smeets@xept.nl

Abstract. This paper suggests to use rounded Gaussians in place of discrete Gaussians in rejection-sampling-based lattice signature schemes like BLISS or Lyubashevsky's signature scheme. We show that this distribution can efficiently be sampled from while additionally making it easy to sample in constant time, systematically avoiding recent timing-based side-channel attacks on lattice-based signatures.
We show the effectiveness of the new sampler by applying it to BLISS, prove analogues of the security proofs for BLISS, and present an implementation that runs in constant time. Our implementation needs no precomputed tables and is twice as fast as the variable-time CDT sampler posted by the BLISS authors with precomputed tables.

Keywords: Post-quantum cryptography, lattice-based cryptography, signatures, Gaussian sampling, BLISS, constant-time implementations.

1 Introduction

Lattice-based cryptography is a promising candidate for post-quantum cryptography. A key reason for this – especially from an applied point of view – is that it is known how to construct efficient signature *and* encryption/key-exchange schemes from lattice assumptions. As both primitives are needed for many applications this is an advantage as it allows for code reuse and relying on one sort of cryptographic hardness assumptions instead of two. For all other well-established candidate areas of post-quantum cryptography we only know how to construct efficient and confidence-inspiring signatures *or* encryption/key-exchange schemes.

In this work we take a look at lattice-based signature schemes. The most efficient lattice-based signature scheme with a security reduction from standard lattice problems today is BLISS (Bimodal Lattice Signature Scheme) [11], designed by Ducas, Durmus, Lepoint and Lyubashevsky. BLISS is one of the few

This work was supported by the European Communities through the Horizon 2020 program under project number 645622 (PQCRYPTO) and project number 645421 (ECRYPT-CSA). Date: 2018.01.11. The full version is [14].

Permanent ID of this document: 7e89be5ae05d8b17a4d03e20a2c454e0.

post-quantum schemes of which there already exists a production-level implementation. BLISS (or rather its subsequent improvement BLISS-b [10] by Ducas) is available in the open-source IPsec Linux library strongSwan [24].

BLISS builds on Lyubashevsky's signature scheme [16] which initiated the use of rejection sampling to make the signature distribution independent of the used secret key. In the most basic version of these schemes, a discrete Gaussian vector is added to a vector that depends on the secret key. The resulting vector follows a discrete Gaussian distribution that is shifted by a vector that depends on the secret key. To avoid leaking the secret key, a rejection step is executed that ensures that the output distribution is independent of the secret key, i.e. the outputs follow again a centered discrete Gaussian distribution.

The use of discrete Gaussian vectors to blind secrets is a very common approach in lattice-based cryptography. However, it is not trivial to sample from a discrete Gaussian efficiently. Over the last few years, many works have been published that deal with efficient sampling routines for discrete Gaussians, see e.g. [8,21,11,12,19]. Despite the number of publications, none achieved constant-time sampling. At CHES 2016, Groot-Bruinderink, Hülsing, Lange, and Yarom [7] demonstrated that these sampling methods enable a cache attack on BLISS which recovers the secret key after less than 5000 signatures. While the attack is only implemented for two samplers, the appendix of the full version surveys other efficient samplers and shows for each of them that they have similar issues.

In [7], the authors already discuss straightforward approaches for achieving constant-time implementations of discrete Gaussian samplers, such as deterministically loading entire tables into cache or fixing the number of iterations for some functions by introducing dummy rounds. While such approaches might work for encryption schemes such as [5], signatures require much wider Gaussians to achieve security. Hence, the impact on efficiency of applying these countermeasures is larger, effectively rendering their use prohibitive.

A different way to deal with such attacks is to complicate the attack. Such a heuristic approach was proposed by Saarinen [22]. However, this approach does not fix the vulnerability, as shown by Pessl [18]; this only makes it harder to exploit it. In consequence, it starts a cat-and-mouse game of attack and fix.

Our contribution. To stop such a cat-and-mouse game before it fully starts, this work deals with ways to systematically fix the vulnerability. We propose to take a completely different approach by replacing discrete Gaussians by a different distribution, namely the *rounded Gaussian* distribution. This distribution shares the benefits of the discrete Gaussians that (slightly) shifted distributions are relatively close to centered distributions. However, the security analysis of using rounded Gaussians in Lyubashevsky's scheme and in BLISS is somewhat more involved than for discrete Gaussians as the probability density function of rounded Gaussians is the integral of a continuous Gaussian. Our main theoretical contribution is a proof that it is safe to replace the discrete Gaussian distribution by a rounded Gaussian distribution in these schemes, while the resulting rejection rates are identical.

As the name suggests, sampling from a rounded Gaussian is done by sampling from a continuous Gaussian and rounding the result to an integer. The Box-Muller method is an efficient way of computing samples of continuous Gaussians starting from uniformly random numbers, and all steps leading to rounded Gaussian samples are efficiently and easily computed in constant time. We present a constant-time implementation of rounded Gaussians suitable for the BLISS-I parameter set and show that it is more than twice as fast as a sampler based on cumulative distribution tables (CDT) as implemented by the authors of BLISS. The CDT sampler uses large precomputed tables to speed up sampling. Note that the CDT sampler is exactly the one that [7] broke at CHES 2016. Using rounded Gaussians brings better speed and better security. Another benefit of rounded Gaussians is that they can use the Box-Muller sampler (see Section 4.1) which naturally does not require any precomputed tables, hence can work with a small code base, and furthermore is extremely easy to implement.

We conclude our work with our second theoretical contribution – a proof that using rounded Gaussians, sampled using our Box-Muller implementation is secure. For this we provide a detailed analysis of the new sampler. We study the difference between (perfect) rounded Gaussians and implementations with finite precision p using statistical distance and Rényi divergence. We also compare the asymptotic results using these different measures. We instantiate the calculation for BLISS parameters and the precision achieved by our Box-Muller implementation to derive bounds on the allowable number of signatures per key pair.

Related work. Rounded Gaussians are not a new distribution, in fact they have been used in the initial proposals for learning with errors, but were replaced by discrete Gaussians to make proofs and protocols easier (the sum of two discrete Gaussians is a discrete Gaussian). See, e.g. Regev [20, p.90] for an overview of distributions and [9] for an analysis of wrapped rounded Gaussians. Encryption schemes can be secure with narrow non-Gaussian distributions (NTRU/Frodo/New Hope) but signatures are much harder to protect, need much wider distributions (larger parameter σ), seemed to need discrete Gaussians, and so far were analyzed only for discrete Gaussians.

The work in this paper is based on Smeets masters thesis [23]. After a first version of our paper was circulated we became aware of a recent paper by Micciancio and Walter [17] which has a new proposal to perform sampling of discrete Gaussians in constant time. The target of our paper is very different: Showing that rounded Gaussians can efficiently be sampled in constant time and that their use in signature schemes is safe.

Acknowledgements. The authors would like to thank Jacob Appelbaum for discussions about the implementation; Daniel J. Bernstein for help with the implementation, benchmarking, and many useful discussions; Leo Ducas for discussions about replacing discrete Gaussians in lattice-based signatures; and Marko Boon for discussions about the standard deviation of the rounded Gaussian distribution.

2 Preliminaries

Vectors, considered as column vectors, will be written in bold lower case letters; matrices will be written in upper case bold letters. For a vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$, the Euclidean norm is defined by $\|\mathbf{a}\| = \sqrt{\sum_{i=1}^n a_i^2}$. The ∞-norm is defined by $\|\mathbf{a}\|_\infty = \max(|a_1|, |a_2|, \ldots, |a_n|)$. The Hamming weight $\mathrm{wt}(\mathbf{a})$ is the number of non-zero positions in $\mathbf{a}$. For two vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, both in $\mathbb{R}^n$, denote the inner product by $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i$.

In this paper we will be concerned with (discrete) probability functions. For a distribution h, we denote by $x \xleftarrow{\$} h$ that x is sampled according to h. For a set S we denote by $s \xleftarrow{\$} S$ that $s \in S$ is sampled uniformly at random from S.

We now cover some background on Gaussian distributions and signature schemes. We follow Lyubashevsky [16] closely and take definitions from there with minor modifications. Many lattice-based schemes use rejection sampling to massage one distribution to fit another. The following m-dimensional version which samples once from the provided distribution and outputs with a certain probability depending on both the target distribution and the sample is copied from Lemma 4.7 of the full ePrint version of [16].

Lemma 2.1 (Rejection Sampling). *Let V be an arbitrary set, and $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. If $g_{\mathbf{v}} : \mathbb{Z}^m \to \mathbb{R}$ is a family of probability distributions indexed by $\mathbf{v} \in V$ with the property*

$$\exists M \in \mathbb{R} : \forall \mathbf{v} \in V : \Pr[M g_{\mathbf{v}}(\mathbf{z}) \geq f(\mathbf{z}); \mathbf{z} \xleftarrow{\$} f] \geq 1 - \varepsilon,$$

then the distribution of the output of the following algorithm $\mathcal{A}$:

```
1: $\mathbf{v} \xleftarrow{\$} h$
2: $\mathbf{z} \xleftarrow{\$} g_{\mathbf{v}}$
3: output $(\mathbf{z}, \mathbf{v})$ with probability $\min\left(\frac{f(\mathbf{z})}{M g_{\mathbf{v}}(\mathbf{z})}, 1\right)$
```

is within statistical distance ε/M of the distribution of the following algorithm $\mathcal{F}$:

```
1: $\mathbf{v} \xleftarrow{\$} h$
2: $\mathbf{z} \xleftarrow{\$} f$
3: output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.
```

Moreover, the probability that $\mathcal{A}$ outputs something is at least $(1 - \varepsilon)/M$.

2.1 Discrete Gaussian Distribution

The discrete Gaussian distribution is based on the continuous Gaussian distribution. The definition of the continuous Gaussian distribution, also called the Normal distribution, is given by:

Definition 2.1. *The continuous Gaussian distribution over $\mathbb{R}^m$ centered at some $\mathbf{v} \in \mathbb{R}^m$ with standard deviation σ is defined for $\mathbf{x} \in \mathbb{R}^m$ as the (joint) density $\rho^m_{\mathbf{v},\sigma}(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$.*

When $\mathbf{v} = \mathbf{0}$, we simply write $\rho_\sigma^m(\mathbf{x})$. The definition of the discrete Gaussian distribution is given by:

Definition 2.2. *The discrete Gaussian distribution over $\mathbb{Z}^m$ centered at some $\mathbf{v} \in \mathbb{Z}^m$ with parameter σ is defined for $\mathbf{x} \in \mathbb{Z}^m$ as $D_{\mathbf{v},\sigma}^m(\mathbf{x}) = \rho_{\mathbf{v},\sigma}^m(\mathbf{x})/\rho_\sigma^m(\mathbb{Z}^m)$, where $\rho_\sigma^m(\mathbb{Z}^m) = \sum_{\mathbf{z}\in\mathbb{Z}^m} \rho_\sigma^m(\mathbf{z})$.*

Note that the discrete Gaussian distribution is defined over all length-m integer vectors in $\mathbb{Z}^m$. However, samples with large entries have negligible probability. Implementations need to provision for the maximal size of coefficients and table-based sampling schemes would require a lot of storage to cover rarely used values and still not cover all possibilities. Therefore, a *tail cut* τ is used, meaning that only integers in $[-\tau\sigma, \tau\sigma]^m$ are sampled. Results about the necessary size of the tail cut can be found in [16] and Lemma A.2. In practice, τ is often chosen as $\sqrt{2\lambda \ln 2}$, where λ is the security level because that ensures a negligible loss in values.

2.2 Lyubashevsky's Signature Scheme

In 2012 Lyubashevsky [16] designed a signature scheme that uses an $m \times n$ matrix $\mathbf{S}$ with small coefficients as secret key and the following two matrices as public key: a random matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$, $m = 2n$, and the $n \times n$ matrix $\mathbf{T} = \mathbf{AS} \bmod q$, where q is an integer. The matrix $\mathbf{A}$ can be shared among all users, but the matrix $\mathbf{T}$ is individual. To sign a message, the signer picks a vector $\mathbf{y}$ according to the m-dimensional discrete Gaussian. Then $\mathbf{c} = H(\mathbf{Ay} \mod q, \mu)$, where $H(\cdot)$ is a hash function, and the potential signature vector $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$ are computed.

The system then uses rejection sampling to shape the distribution of $\mathbf{z}$ to a centered discrete Gaussian, i.e., to decide whether to output the candidate signature $(\mathbf{z}, \mathbf{c})$. In terms of Lemma 2.1, h is the distribution of $\mathbf{Sc}$, $g_{\mathbf{v}}$ is the m-dimensional discrete Gaussian $D_{\mathbf{v},\sigma}(\mathbf{z})$ centered around $\mathbf{v} = \mathbf{Sc}$, and f is the m-dimensional centered discrete Gaussian $D_\sigma^m(\mathbf{z})$.

Because $D_\sigma^m(\mathbf{z})$ is independent of $\mathbf{S}$ the signatures do not leak information about the private key.

2.3 Bimodal Lattice Signature Scheme: BLISS

Ducas, Durmus, Lepoint and Lyubashevsky introduced the Bimodal Lattice Signature Scheme (BLISS) in [11]. BLISS is an improvement of Lyubashevsky's signature scheme described above in that signatures are smaller and generated faster. We only cover signature generation here as we are focusing on the use of the discrete Gaussian distribution. For a full description of BLISS, see [11]. BLISS uses a special hash function H mapping to $\{\mathbf{c} \in \{0,1\}^n | \mathrm{wt}(\mathbf{c}) = \kappa\}$ for κ some small constant. A simplified version of the BLISS signature algorithm is given in Algorithm 2.1.

Given a message μ, the signing algorithm first samples a vector $\mathbf{y}$ from the m-dimensional discrete Gaussian distribution D_σ^m. Then it computes the hash $\mathbf{c} \leftarrow$

Algorithm 2.1 Simplified BLISS Signature Algorithm using matrices

```
Input: Message $\mu$, public key $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$ and secret key $\mathbf{S} \in \mathbb{Z}_{2q}^{m\times n}$
Output: A signature $(\mathbf{z}, \mathbf{c})$ of the message $\mu$
1: $\mathbf{y} \leftarrow D_\sigma^m$
2: $\mathbf{c} \leftarrow H(\mathbf{Ay} \bmod 2q, \mu)$        // $\mathbf{c} \in \{0,1\}^n$, $\mathrm{wt}(\mathbf{c}) = \kappa$, $\kappa$ small constant
3: Choose a random bit $b \in \{0,1\}$
4: $\mathbf{z} \leftarrow \mathbf{y} + (-1)^b \mathbf{Sc}$
5: Output $(\mathbf{z}, \mathbf{c})$ with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc}\rangle}{\sigma^2}\right)\right)$
```

$H(\mathbf{Ay} \bmod 2q, \mu)$. It samples a random bit $b \in \{0,1\}$ and computes the potential signature $\mathbf{z} \leftarrow \mathbf{y} + (-1)^b\mathbf{Sc}$. Now that the signing algorithm has $\mathbf{z}$, it performs rejection sampling according to Lemma 2.1, i.e., it outputs the signature $(\mathbf{z}, \mathbf{c})$ with probability $1 \Big/ \left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc}\rangle}{\sigma^2}\right)\right)$, where M is some fixed positive real constant that is set large enough to ensure that this probability is at most 1 for all choices of $\mathbf{c}$. If the signature algorithm is unsuccessful, it restarts with a fresh $\mathbf{y}$ and continues until a signature is output.

Again, rejection sampling is used to force the distribution of the output $\mathbf{z}$ to be that of a centered Gaussian distribution (i.e., to be independent of $\mathbf{Sc}$).

The bulk of the time in one round of the signing algorithm using BLISS is spent in the first step in generating m samples from the one-dimensional Gaussian. The number of repetitions depends on M and the size of $\mathbf{Sc}$.

Bound on $\|\mathbf{Sc}\|$. The parameter σ of the discrete Gaussian distribution, the size of $\mathbf{Sc}$, and the rejection rate M control how much the distributions of the target distribution D_σ^m and the input distribution overlap, i.e., how small ε can be achieved. For BLISS the input distribution is a bimodal Gaussian distribution $0.5(D^m_{-\mathbf{Sc},\sigma} + D^m_{\mathbf{Sc},\sigma})$. BLISS' authors show that rejection sampling can be used without error, i.e., $\varepsilon = 0$ is possible in Lemma 2.1 with resonable choices of σ and M. In later sections we require an upper bound on $\|\mathbf{Sc}\|$ for proofs. In [11] a new measure $N_\kappa(\mathbf{X})$ of $\mathbf{S}$, adapted to the form of $\mathbf{c}$, is presented.

Definition 2.3. *For any integer κ, $N_\kappa : \mathbb{R}^{m\times n} \to \mathbb{R}$ is defined as:*

$$N_\kappa(\mathbf{X}) = \max_{I\subset\{1,\ldots,n\},\#I=\kappa} \sum_{i\in I} \left(\max_{J\subset\{1,\ldots,n\},\#J=\kappa} \sum_{j\in J} W_{i,j} \right),$$

where $\mathbf{W} = \mathbf{X}^T \cdot \mathbf{X} \in \mathbb{R}^{n\times n}$.

With this definition, the authors of [11] show that for any $\mathbf{c} \in \{0,1\}^n$ with $\mathrm{wt}(\mathbf{c}) \le \kappa$, we have $\|\mathbf{Sc}\|^2 \le N_\kappa(\mathbf{S})$ [11, Proposition 3.2]. In addition to the use of bimodal Gaussians, this upper bound lowers the parameter σ by a factor $\approx \sqrt{\kappa}/2$ compared to [16].

3 Rounded Gaussian Rejection Sampling

In this section we discuss the applicability of the *rounded Gaussian distribution* in rejection-sampling-based signature schemes. After giving a formal definition of the rounded Gaussian distribution, we provide proofs showing that it can be used to replace the discrete Gaussian distribution in Lyubashevsky's signature scheme and in BLISS. We show the analogies between the rounded Gaussian distribution and the discrete Gaussian distribution and we point out where the security reductions differ when rounded Gaussians are used in place of discrete Gaussians. In practice, the most important question is how the probability in Step 5 in Algorithm 2.1 (and the equivalent on in Lyubashevsky's scheme) needs to change if $\mathbf{y}$ is sampled according to the rounded Gaussian distribution instead of the discrete Gaussian distribution. Note, again, that this step determines the rejection rate, i.e. how many times the algorithm needs to restart sampling fresh randomness.

To simplify comparisons and show that rounded Gaussians can be used in place of discrete Gaussians we follow the presentation and structure from [16] and [11] very closely. The main difference is that the definition of rounded Gaussians requires integrals over an interval of length 1, while the definition of discrete Gaussians requires a division by the probability mass at all integers. We essentially have to prove the same lemmas that were shown for discrete Gaussians in [16] and [11] for rounded Gaussians. In the end closely analogous results hold but the analysis turns out far more complicated than in the discrete Gaussian setting because we have to deal with bounding integrals.

3.1 Rounded Gaussian Distribution

We now formally define the rounded Gaussian distribution. Intuitively, the rounded Gaussian distribution is obtained by rounding samples from a continuous Gaussian distribution to the nearest integer x_i. To compute the probability at an integer x_i, we compute the integral over the interval $(x_i - \frac{1}{2}, x_i + \frac{1}{2}]$.

Definition 3.1. *The* rounded Gaussian distribution *over $\mathbb{Z}^m$ centered at some $\mathbf{v} \in \mathbb{Z}^m$ with parameter σ is defined for $\mathbf{x} \in \mathbb{Z}^m$ as*

$$R^m_{\mathbf{v},\sigma}(\mathbf{x}) = \int_{A_\mathbf{x}} \rho^m_{\mathbf{v},\sigma}(\mathbf{s}) d\mathbf{s} = \int_{A_\mathbf{x}} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(\frac{-\|\mathbf{s}-\mathbf{v}\|^2}{2\sigma^2}\right) d\mathbf{s},$$

where $A_\mathbf{x}$ denotes the area defined by $\left[x_1 - \frac{1}{2}; x_1 + \frac{1}{2}\right) \times \cdots \times \left[x_m - \frac{1}{2}; x_m + \frac{1}{2}\right)$.

We point out that this gives us $\mathrm{vol}(A_\mathbf{x}) = 1$, since the volume of this area is equal to $|(x_1 + \frac{1}{2}) - (x_1 - \frac{1}{2})| \cdots |(x_m + \frac{1}{2}) - (x_m - \frac{1}{2})|$. Note that the parameter σ in the definition above is the standard deviation of the underlying continuous Gaussian and not the standard deviation σ' of the rounded Gaussian distribution, which is given by $\sigma' = \sqrt{\sigma^2 + \frac{1}{12}} + \epsilon(\alpha)$, where $\epsilon(\alpha)$ is some function of small value with mean 0.

3.2 Using Rounded Gaussians in Lyubashevsky's Scheme

The proofs by Lyubashevsky [16] for the discrete Gaussian distribution rely on several lemmas for which we prove analogous statements in Appendix A. The following lemma states that the centered rounded Gaussian $R_\sigma^m(\mathbf{z})$ and the shifted rounded Gaussian $R_{\mathbf{v},\sigma}(\mathbf{z})$ are almost always close, and Theorem 3.1 applies it to the rejection-sampling Lemma 2.1.

Lemma 3.1. *For any $\mathbf{v} \in \mathbb{Z}^m$, if $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$, then*

$$\Pr\left[R_\sigma^m(\mathbf{z})/R_{\mathbf{v},\sigma}^m(\mathbf{z}) = O(1); \mathbf{z} \overset{\$}{\leftarrow} R_\sigma^m\right] = 1 - 2^{-\omega(\|\mathbf{v}\|\sqrt{\log m})}.$$

This is proven in Appendix A.

Theorem 3.1. *Let V be a subset of $\mathbb{Z}^m$ in which all elements have norms less than T, σ be some element in $\mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists a constant $M = O(1)$ such that the distribution of the following algorithm $\mathcal{A}$:*

```
1: $\mathbf{v} \overset{\$}{\leftarrow} h$
2: $\mathbf{z} \overset{\$}{\leftarrow} R_{\mathbf{v},\sigma}^m$
3: output $(\mathbf{z}, \mathbf{v})$ with probability $\min\left(\frac{R_\sigma^m(\mathbf{z})}{M R_{\mathbf{v},\sigma}^m(\mathbf{z})}, 1\right)$
```

is within statistical distance $2^{-\omega(\log m)}/M$ of the distribution of the following algorithm $\mathcal{F}$:

```
1: $\mathbf{v} \overset{\$}{\leftarrow} h$
2: $\mathbf{z} \overset{\$}{\leftarrow} R_\sigma^m$
3: output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.
```

Moreover, the probability that $\mathcal{A}$ outputs something is at least $(1-2^{-\omega(\log m)})/M$.

Proof. The proof of this theorem follows immediately from Lemma 3.1 and the general "rejection sampling" Lemma 2.1. □

This theorem looks the same for rounded Gaussians and for discrete Gaussians; see Appendix A.1 for a detailed comparison of the results.

3.3 Using Rounded Gaussians in BLISS

In Section 3.2 we have shown that we can use the rounded Gaussian distribution in the rejection sampling scheme by Lyubashevsky [16]. In this section we show how to apply the rounded Gaussian distribution to BLISS and that the same constant as in BLISS can be for rejection sampling.

BLISS randomly flips a bit to decide on adding or subtracting $\mathbf{Sc}$, i.e., for fixed $\mathbf{Sc}$, $\mathbf{z}^*$ is distributed according to the bimodal rounded Gaussian distribution $g_{\mathbf{Sc}}(\mathbf{z}^*) = \frac{1}{2}R_{\mathbf{Sc},\sigma}^m(\mathbf{z}^*) + \frac{1}{2}R_{-\mathbf{Sc},\sigma}^m(\mathbf{z}^*)$. To avoid leaking any information on the secret key $\mathbf{S}$ the scheme requires rejection sampling to change the bimodal Gaussian to a centered Gaussian $f(\mathbf{z}^*) = R_\sigma^m(\mathbf{z}^*)$. The probability to accept is given by $p_{\mathbf{z}^*} = f(\mathbf{z}^*)/Mg_{\mathbf{Sc}}(\mathbf{z}^*)$, where again M is chosen minimal such that this probability is ≤ 1 for all $\mathbf{z}^*$.

The results of this section are completely analogous to those in [11].

For any $\mathbf{z}^* \in \mathbb{Z}^m$, we have

$$\begin{aligned}\Pr[\mathbf{z} = \mathbf{z}^*] &= \tfrac{1}{2}R^m_{\mathbf{Sc},\sigma}(\mathbf{z}^*) + \tfrac{1}{2}R^m_{-\mathbf{Sc},\sigma}(\mathbf{z}^*)\\ &= \tfrac{1}{2}\left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_{\mathbf{z}^*}} \exp\left(-\tfrac{\|\mathbf{x}-\mathbf{Sc}\|^2}{2\sigma^2}\right) + \exp\left(-\tfrac{\|\mathbf{x}+\mathbf{Sc}\|^2}{2\sigma^2}\right) d\mathbf{x} \qquad (1)\\ &= \exp\left(-\tfrac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right)\left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_{\mathbf{z}^*}} \exp\left(-\tfrac{\|\mathbf{x}\|^2}{2\sigma^2}\right)\cosh\left(\tfrac{\langle\mathbf{x},\mathbf{Sc}\rangle}{\sigma^2}\right) d\mathbf{x}.\end{aligned}$$

The desired output is the centered rounded Gaussian distribution $f(\mathbf{z}^*)$, since we need the centered property to avoid leaking $\mathbf{S}$. Thus by Theorem 3.1, we should accept the sample $\mathbf{z}^*$ with probability:

$$\begin{aligned}&p_{\mathbf{z}^*} = f(\mathbf{z}^*)/(Mg_{\mathbf{Sc}}(\mathbf{z}^*))\\ &= \frac{\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right) d\mathbf{x}}{M\exp\left(-\|\mathbf{Sc}\|^2/(2\sigma^2)\right)\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right)\cosh\left(\langle\mathbf{x},\mathbf{Sc}\rangle/\sigma^2\right) d\mathbf{x}}.\end{aligned}$$

To compute a bound on M, we use Equation (1) and that $\cosh(x) > 0$ for any x. This leads to the following upper bound:

$$\begin{aligned}p_{\mathbf{z}^*} &= \frac{\int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right) d\mathbf{x}}{M\exp\left(-\|\mathbf{Sc}\|^2/(2\sigma^2)\right)\int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right)\cosh\left(\langle\mathbf{x},\mathbf{Sc}\rangle/\sigma^2\right) d\mathbf{x}}\\ &\leq \frac{\int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right) d\mathbf{x}}{M\exp\left(-\|\mathbf{Sc}\|^2/(2\sigma^2)\right)\int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right) d\mathbf{x}}\\ &= 1/\left(M\exp\left(-\|\mathbf{Sc}\|^2/(2\sigma^2)\right)\right).\end{aligned}$$

Now M needs to be chosen large enough such that $p_{\mathbf{z}^*} \leq 1$. Note that the last inequality can only be used to estimate M, and not to define the probability. It suffices that $M = \exp\left(1/(2\alpha^2)\right)$', where $\alpha > 0$ is such that $\sigma \geq \alpha\|\mathbf{Sc}\|$. We can use the upper bound $\|\mathbf{Sc}\|^2 \leq N_\kappa(\mathbf{S})$ as in Definition 2.3 to put $M = \exp(N_\kappa(\mathbf{S})/2\sigma^2)$; here κ denotes the sparsity of $\mathbf{c}$ in Algorithm 2.1. This is the same constant as in BLISS.

3.4 BLISS Security Reduction

The security proof as given in [11] works for the rounded Gaussian distribution with very little tweaking. This is due to the changes made in the proofs in Section 3.2 and Appendix A. All statements follow through when replacing the discrete Gaussian distribution with the rounded Gaussian distribution. We do not need to adjust the proofs for [11, Lemma 3.3, 3.5]. The proof for [11, Lemma 3.4] uses $\sigma \geq 3/\sqrt{2\pi}$ which comes from [16, Lemma 4.4]. Our corresponding result is Lemma A.2 which requires $\sigma \geq \sqrt{2/\pi}$. Next to that, we need to adjust the definitions of $f(\mathbf{z})$ and $g_{\mathbf{Sc}}(\mathbf{z})$ as above, such that these match the rounded Gaussian distribution.

4 Practical Instantiation

In this section we discuss how we can implement a sampler for the rounded Gaussian distribution. A very efficient and easy way to generate samples from the continuous Gaussian distribution is based on the Box-Muller transform. We state the algorithm and discuss an early rejection technique to prevent the computation of values which would later be rejected due to the tail cut. Finally, we analyze the output precision required for an implementation of the rounded Gaussian distribution.

4.1 Box-Muller Transform

We begin by reviewing the Box-Muller transform [6] which is used to create centered Gaussian distributed numbers with standard deviation $\sigma = 1$ from uniform random distributed numbers. The algorithm is given as Algorithm 4.1 below.

Algorithm 4.1 Box-Muller Sampling

```
Input: Two uniform numbers $u_1, u_2 \in (0, 1]$
Output: Two independent centered (continuous) Gaussian distributed numbers $x_1, x_2$
    with standard deviation $\sigma = 1$
1: $a \leftarrow \sqrt{-2 \ln u_1}$
2: $b \leftarrow 2\pi u_2$
3: $(x_1, x_2) \leftarrow (a \cos b, a \sin b)$
4: return $(x_1, x_2)$
```

4.2 Sampling Rounded Gaussians

We can now use the Box-Muller transform to create an algorithm for sampling according to the rounded Gaussian distribution. For applying rounded Gaussians to the signature scheme of BLISS, we need centered rounded Gaussians with parameter σ. This is done by scaling the output x_i for $i = 1, 2$ of the Box-Muller sampling scheme $z'_i = x_i \cdot \sigma$ and then rounding the nearest integer $z_i = \lfloor z'_i \rceil$.

4.3 Rejection Sampling of Signatures

At the end of Algorithm 2.1 we need to output $(\mathbf{z}, \mathbf{c})$ with probability

$$\frac{2 \int_{A_{\mathbf{z}^*}} \exp\left(-\|\mathbf{x}\|^2/(2\sigma^2)\right) d\mathbf{x}}{M \cdot \exp\left(-\|\mathbf{Sc}\|^2/(2\sigma^2)\right) \left(\int_{A_{\mathbf{z}^*}} \exp\left(-\frac{\|\mathbf{x}-\mathbf{Sc}\|^2}{2\sigma^2}\right) d\mathbf{x} + \int_{A_{\mathbf{z}^*}} \exp\left(-\frac{\|\mathbf{x}+\mathbf{Sc}\|^2}{2\sigma^2}\right) d\mathbf{x}\right)}$$

(see Section 3.2).

Each of the three integrals factors, i.e., can be computed as the product of one-dimensional integrals. Each one-dimensional integral is

$$\int_{z_i-1/2}^{z_i+1/2} \exp\left(\frac{-x_i^2}{2\sigma^2}\right) dx_i = \sigma\sqrt{\frac{\pi}{2}}\left(\operatorname{erf}\left(\frac{z_i+1/2}{\sqrt{2\sigma^2}}\right) - \operatorname{erf}\left(\frac{z_i-1/2}{\sqrt{2\sigma^2}}\right)\right),$$

i.e., a constant times a difference of two nearby values of the standard error function (erf).

5 Code Analysis and Benchmarks

This section provides details about our implementation. First we give a general overview over our implementation. Then we discuss the dependency between floating point precision and allowable number of signatures. We end with timings and a comparison to the BLISS CDT sampler.

5.1 Implementation Details

We have used the C++ vector class library VCL by Fog [13] for the implementation of the Box-Muller sampling and the rounded Gaussian sampling. This library offers optimized vector operations for integers, floating point numbers and booleans. We use `Vec8d`, which are vectors with 8 elements of double floating point precision. This means that we are only limited by the maximum size of the `double` type, i.e. values of at most 53 bits of precision.

According to [13], the trigonometric and logarithmic functions in VCL have constant runtime, i.e. there is no timing difference dependent on the input. This makes the library ideal for constant-time implementations. The square-root function `sqrt(·)` takes constant time, unless all 8 inputs are in $\{0, 1\}$, which can lead to a timing difference for the square root. However, this is unlikely to happen: the `sqrt` function is applied to $2 \ln u_1$ and the logarithm function is strictly positive and thus the case of input 0 cannot appear; the probability of sampling 8 consecutive values u_{1i} that all would evaluate $2 \ln u_{1i} = 1$ is negligible, since each u_{1i} is sampled from $(0, 1]$ with 53 bit precision, making this an event of probability at most $2^{-8\cdot 53}$. Therefore we have chosen not to circumvent this problem in the implementation, even though one could also sacrifice a vector entry and force it to have a nontrivial square root computation.

Computing with floating-point numbers causes a drop in precision. While Fog states that operations in VCL lose at most one bit of precision with exception of several explicitly mentioned functions that can lose up to two bits of precision such as the trigonometric functions, a more careful analysis of the code shows that other operations keep (close to) the exact precision.

Sampling rounded Gaussians on top of VCL is only a few lines of code and the data paths are short (see the code listing in Appendix F of the full version [14]). The input of the code has 53 bits of precision and we loose at most 5 bits of precision, i.e. the output of the code has at least $p = 48$ bits of precision.

Remark 1. We were asked how to round floating point numbers in constant time. While VCL almost trivially rounds the entire vector in constant time, a bit more care is necessary if one wants to implement this on single values. To round $|A| < 2^{51}$ compute

$$(A + (2^{52} + 2^{51})) - (2^{52} + 2^{51})$$

in two arithmetic instructions or use assembly instructions.

5.2 Considerations Regarding the Precision

Samplers for discrete Gaussians typically require tables precomputed at a certain precision. This raises the question of how much a low-precision table can skew the distribution and whether this can lead to attacks. Similarly, floating-point computations, such as in our sampler, can slowly degrade precision.

An error in the computation of y results in a value y' which might be slightly larger or smaller than y. The magnitude of the error depends on the size of the value, e.g., values close to 0 have higher precision than larger values; in general the error of y is bounded by $|y|2^{-p}$.

When computing rounded Gaussians, most errors are insignificant because most erroneous values still get rounded to the correct integer. However, errors occurring close to the boundaries of the intervals $[z - \frac{1}{2}, z + \frac{1}{2}]$ can lead to wrong outputs. The interval of values that can possibly round to z is given by $[z - \frac{1}{2} - e_l, z + \frac{1}{2} + e_r)$, where the left boundary error satisfies $|e_l| \leq 2^{-p} \left|z - \frac{1}{2}\right|$ and the right boundary error satisfies $|e_r| \leq 2^{-p} \left|z + \frac{1}{2}\right|$.

We define success for the attacker to mean that he breaks the signature scheme or that he manages to distinguish between the implementation with precision p and a perfect implementation.

Most papers use the statistical distance (Definition B.1) to study the relative difference between two distributions. In [1] the authors showed that studying the Rényi divergence between the distributions can lead to better and tighter estimates.

In this section we work with the known precision $p = 48$ for our implementation and using the parameters for BLISS-I [11], we determine how many signatures an adversary $\mathcal{A}$ can observe before the Rényi divergence between the ideal implementation and the practical implementation becomes larger than some small constant c; this means, his chance of breaking the system is at most c times as high compared to the ideal implementation.

We also provide an analysis of the asymptotic behavior of the precision p compared to the standard deviation σ, the length m and the number of signatures q_s generated. The computations can be found in Appendix B. These results are naturally less tight because we prioritize readable formulas over best approximations. Accordingly, better results are obtained using numerical computations once one settles on concrete parameters. The asymptotic analysis is helpful in determining which distance or divergence to use.

To analyze the allowable number of signatures q_s before an attack could possibly distinguish the distributions, we look at the *Rényi divergence of order* ∞ as given in [1]:

Definition 5.1. *For any two discrete probability distributions P and Q, such that* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$*, the* Rényi divergence of order ∞ *is defined by*

$$\mathrm{RD}_\infty(P \mid\mid Q) = \max_{x \in \mathrm{Supp}(P)} \frac{P(x)}{Q(x)}.$$

In BLISS using rounded Gaussians we publish m independently sampled integers distributed according to the 1-dimensional rounded Gaussian distribution R^1_σ to obtain an m-dimensional vector in R^m_σ. Next to that we assume q_s signing queries. This means a potential attacker can learn a vector of length mq_s with entries from the (imprecise) real-world sampler R'^1_σ. We want to determine the probability that an attacker can distinguish between a vector sampled from $R^{mq_s}_\sigma$ and $R'^{mq_s}_\sigma$.

By the probability preservation property (Lemma B.2) of the Rényi divergence, any adversary $\mathcal{A}$ having success probability ϵ on the scheme implemented with imprecise rounded Gaussian sampling has a success probability $\delta \geq \epsilon / \mathrm{RD}_\infty(R'^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma)$ on the scheme implemented with the perfect rounded Gaussian. For a target success probability ϵ we have to choose $\delta \leq \epsilon / \exp(1)$ to have only a small, constant loss in tightness.

We need mq_s samples to create q_s signatures. By the multiplicative property of the Rényi divergence (Lemma B.1), we have $\mathrm{RD}_\infty(R'^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma) \leq \mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma)^{mq_s}$, so we can relate the divergence of the one-dimensional distributions to the mq_s dimensional one. The formula becomes

$$\mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma) = \\ \max_{z \in \mathrm{Supp}(R'^1_\sigma)} \left\{ \int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx \Bigg/ \int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx \right\}.$$

The BLISS-I parameters are $\sigma = 215$, $m = 2n = 1024$, and $\epsilon = 2^{-128}$, giving $\tau = \sqrt{2 \cdot 128 \ln(2)} = 13.32$, and we work with floating point precision $p = 48$. We compute RD_∞ numerically for the 1-dimensional case with Pari-GP with precision 200 digits, giving $\mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma) \approx 1.0000000000203563$. Recall we want $\mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma)^{mq_s} \leq \exp(1)$. For $m = 1024$ we get that $q_s = 2^{25}$ gives $2.01262 < \exp(1)$. This means that we can create 2^{25} signatures, i.e., 1 signature/min for over 60 years, securely with one key pair. Note also that the choice of $\exp(1)$ is kind of arbitrary and other constants would be suitable as well. Moreover, provable security continues to degrade slowly after these 2^{25} signatures. As far as we know, no attack is known that would use the distinguishability of the distributions.

Several papers, starting with [1], use Rényi divergence RD_a of order a to get much better results regarding the precision. We caution the reader that the relation $\delta > \epsilon^{a/(a-1)}/RD_a$, for $a = 2$, $\delta = 2^{-128}$ and constant $\mathrm{RD}_a = 2$, means $\epsilon = 2^{-64}$, which is loose to the point of being meaningless. For the same looseness we could use constant 2^{64} in place of $\exp(1)$ in RD_∞ and sign 2^{88} times.

5.3 Implementation of Rejection Sampling of Signatures

There are many standard numerical techniques and libraries to efficiently compute the complementary error function $1 - \mathrm{erf}$ to high precision. We use the following constant-time mixture of standard techniques: for fixed s, the integral of e^{-x^2} for x ranging from $t-s/2$ to $t+s/2$ is e^{-t^2} (which we compute in constant time using VCL) times a quickly converging power series in t^2. For the constants $s = 1/\sqrt{2\sigma^2}$ relevant to BLISS-I through BLISS-IV, and for the entire range of t allowed by our tail cut, the truncation error after five terms of this power series is below the rounding error of double-precision floating-point computation.

Each of the three 1024-dimensional integrals is computed by the `bigintegral` function shown in Appendix G of the full version [14], which is implemented in just four lines of code, on top of the `erfdiff` function, which is implemented in just two lines of code, plus a few constants precomputed from σ. The rest of the code in Appendix G in [14] is for speeding up VCL's `exp` by replacing it with a streamlined `fastexp`; running a Monte-Carlo sanity check on `bigintegral`; and benchmarking `bigintegral`.

Each call to `bigintegral` takes just 7800 cycles on a Haswell CPU core using `g++ 4.8.4` with standard compiler options (`-O3 -fomit-frame-pointer -std=gnu++11 -march=native -mtune=native -fabi-version=6`), and there are three integrals in the computation of rejection probabilities. (Dividing the integrals and comparing to a random number is equivalent to multiplying the random number by the denominator and comparing to the numerator, which takes constant time.) We save a lot more than these $3 \cdot 7800 = 23400$ cycles in the sampling step (see Table 5.1). Furthermore, the main point of the approach is to produce constant-time implementations, and our code is constant time.

There are only a small number of possible inputs to `erfdiff`. Specifically, each y_i is an integer in $[-\tau\sigma, \tau\sigma]$, and each entry of $\mathbf{Sc}$ is an integer bounded in absolute value by 3κ for BLISS-I and II and 5κ for BLISS-III and IV, so each `erfdiff` input is an integer bounded in absolute value by $\tau\sigma + 3\kappa$ or $\tau\sigma + 5\kappa$ respectively.

To compute the effects of approximating erf and working with finite-precision floating-point numbers we calculated the ratio of the result from our calculation (for all possible `erfdiff` inputs) to the exact solution, where we used Sage's arbitrary-precision `error_fcn` with 1000 bits of precision to very precisely compute the exact solution. The one-dimensional Rényi divergence RD_∞ of these distributions is defined as the maximum of these fractions.

For example, in 17 seconds on a 3.5GHz Haswell core we calculate for BLISS-I that $\mathrm{RD}_\infty(\text{ approx calculation} \,||\, \text{exact calculation}) < 1 + 2^{-46}$.

Using that RD_∞ is multiplicative and $(1 + 2^{-46})^{2^{46}} < \exp(1)$ we get that for $m = 1024$ we can output 2^{36} signatures without the attacker gaining more than a factor of $\exp(1)$. This is more than the number in Section 5.2 so the approximation is sufficiently good.

5.4 Timings for Sampling Rounded Gaussians

Another property that needs to be compared between the rounded Gaussian distribution and the discrete Gaussian distribution is the time it takes to generate one signature. We compare our implementation to the CDT implementation from `http://bliss.di.ens.fr/` which is a proof-of-concept, variable-time sampler for discrete Gaussians.

Both the discrete Gaussian and the rounded Gaussian can be used in the BLISS signature scheme as we have shown earlier. We now compare the time that it takes to generate $m = 1024$ samples by the two sampling schemes. We note that in a full implementation there are more steps to generate a signature, e.g., the rejection step. However, as said before, these steps are not the bottle neck and take approximately equal time for either sampling scheme; thus we do not include them in the analysis.

Our implementation starts by drawing random bits from `/dev/urandom` and then expanding them using ChaCha20 [3] to 8192 bytes of data. From that 128 vectors of 8 53-bit floating-point variables are initialized with randomness, corresponding to the initial u_i values in Algorithm 4.1. The rest of the implementation follows closely the description of that algorithm.

Both implementations have been compiled using `gcc` with `-O3`. The benchmarks have been run on a Haswell Intel(R) chip, i.e. Intel(R) Xeon(R) CPU E3-1275 v3 3.50GHz. All values given in Table 5.1 are given in CPU cycles. We give the quartiles Q1 and Q3 and the median over 10 000 runs to show the statistical stability.

Name of the scheme	Q1	Median	Q3
Rounded Gaussians (including generating randomness)	47532	47576	47616
Rounded Gaussians (without generating randomness)	27608	27672	27848
Discrete Gaussians (including generating randomness)	115056	116272	127170
Discrete Gaussians (without generating randomness)	77424	78136	78876

Table 5.1. CPU cycles analysis for the rounded Gaussian sampling scheme and discrete Gaussian sampling scheme with $m = 1024$ run on Intel(R) Xeon(R) CPU E3-1275 v3 3.50GHz., stating median and quartiles for 10 000 runs.

In Table 5.1 we can clearly see that the rounded Gaussian implementation is significantly faster than the discrete Gaussian implementation; the rounded Gaussian implementation needs noticeably less than half the number of CPU cycles compared to the discrete Gaussian implementation. We can also see that generating the randomness takes a significant part of the total CPU cycle count.

While the difference in speed is significant we would like to point out that the implementation we used for the discrete Gaussians is not fully optimized. It is hard to predict how much faster a better implementation would be and how much worse the performance would drop if countermeasures to achieve constant-time behavior were implemented.

Our motivation after [7] was to find an alternative to hard-to-secure discrete Gaussians, even if it was *slower* than current implementations. Our implementation shows that with less than 40 lines of code rounded Gaussians are at least fully competitive.

References

1. Shi Bai, Adeline Langlois, Tancrède Lepoint, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In *ASIACRYPT (1)*, volume 9452 of *Lecture Notes in Computer Science*, pages 3–24. Springer, 2015.
2. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
3. Daniel J. Bernstein. The ChaCha family of stream ciphers. ChaCha, a variant of Salsa20 [4], https://cr.yp.to/chacha.html.
4. Daniel J. Bernstein. The Salsa20 family of stream ciphers. In *New stream cipher designs: the eSTREAM finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.
5. Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *IEEE Symposium on Security and Privacy*, pages 553–570. IEEE Computer Society, 2015.
6. George E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *Ann. Math. Statist.*, 29(2):610–611, 1958.
7. Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, Gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 323–345. Springer, 2016.
8. Johannes A. Buchmann, Daniel Cabarcas, Florian Göpfert, Andreas Hülsing, and Patrick Weiden. Discrete Ziggurat: A time-memory trade-off for sampling from a Gaussian distribution over the integers. In Lange et al. [15], pages 402–417.
9. Alexandre Duc, Florian Tramèr, and Serge Vaudenay. Better algorithms for LWE and LWR. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 173–202. Springer, 2015.
10. Léo Ducas. Accelerating BLISS: the geometry of ternary polynomials. IACR Cryptology ePrint Archive, Report 2014/874, 2014. https://eprint.iacr.org/2014/874.
11. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013. full version http://eprint.iacr.org/2013/383.

12. Nagarjun C. Dwarakanath and Steven D. Galbraith. Sampling from discrete Gaussians for lattice-based cryptography on a constrained device. *Appl. Algebra Eng. Commun. Comput.*, 25(3):159–180, 2014.
13. Agner Fog. VCL C++ vector class library, 2016. Code and documentation, www.agner.org/optimize.
14. Andreas Hülsing, Tanja Lange, and Kit Smeets. Rounded Gaussians – Fast and Secure Constant-Time Sampling for Lattice-Based Crypto. IACR Cryptology ePrint Archive, Report 2017/1025, 2017. https://eprint.iacr.org/2017/1025.
15. Tanja Lange, Kristin E. Lauter, and Petr Lisonek, editors. *SAC*, volume 8282 of *Lecture Notes in Computer Science*. Springer, 2014.
16. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.
17. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 455–485. Springer, 2017.
18. Peter Pessl. Analyzing the shuffling side-channel countermeasure for lattice-based signatures. In *INDOCRYPT*, volume 10095 of *Lecture Notes in Computer Science*, pages 153–170, 2016.
19. Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In *CHES*, volume 8731 of *Lecture Notes in Computer Science*, pages 353–370. Springer, 2014.
20. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *ACM Conference on Computer and Communications Security*, 56(6), 2009.
21. Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. High precision discrete Gaussian sampling on FPGAs. In Lange et al. [15], pages 383–401.
22. Markku-Juhani O. Saarinen. Gaussian sampling precision and information leakage in lattice cryptography. IACR Cryptology ePrint Archive, Report 2015/953, 2015. https://eprint.iacr.org/2015/953.
23. Kit Smeets. Securing BLISS-b Against Side-Channel Attacks Lattice-Based Cryptography in a Post-Quantum Setting. Master thesis at Technische Universiteit Eindhoven, 2017.
24. strongSwan. strongSwan 5.2.2 released, January 2015. https://www.strongswan.org/blog/2015/01/05/strongswan-5.2.2-released.html.
25. Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Information Theory*, 60(7):3797–3820, 2014.

A Proofs for Rounded Gaussian Rejection Sampling

In this section we provide the missing lemmas and proofs from Section 3. We follow the structure of the security proofs in [16] to show that we can use the rounded Gaussian distribution in Lyubashevsky's scheme. Similarly to [16], our main proof relies on several lemmas about the rounded Gaussian distribution over $\mathbb{Z}^m$. The statements in the lemmas proven here differ slightly from those in [16] but serve analogous purposes.

First we look at the inner product of a rounded Gaussian variable with any vector in $\mathbb{R}^m$.

Lemma A.1. *For any fixed vector* $\mathbf{u} \in \mathbb{R}^m$ *and any* $\sigma, r > 0$, *we have*

$$\Pr[|\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle| > r; \mathbf{z} \xleftarrow{\$} R_\sigma^m] \leq 2e^{-\frac{r^2}{2\|\mathbf{u}\|^2\sigma^2}},$$

where $\mathbf{y} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ *minimizes* $\exp\left(\frac{1}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right)$.

Proof. Let $\mathbf{u} \in \mathbb{R}^m$ be fixed and let $\mathbf{y} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ be such that $\exp\left(\frac{1}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right)$ is minimized. For any $t > 0$, we have for the expectation of $\exp\left(\frac{t}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right)$, taken over all $\mathbf{z}$ sampled from R_σ^m:

$$\begin{aligned}
E\left[\exp\left(\tfrac{t}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right)\right] &= \exp\left(\tfrac{t}{\sigma^2}\langle \mathbf{y}, \mathbf{u}\rangle\right) E\left[\exp\left(\tfrac{t}{\sigma^2}\langle \mathbf{z}, \mathbf{u}\rangle\right)\right] \\
&= \exp\left(\tfrac{t}{\sigma^2}\langle \mathbf{y}, \mathbf{u}\rangle\right) \sum_{\mathbf{z}\in\mathbb{Z}^m} \Pr[\mathbf{z}] \exp\left(\tfrac{1}{\sigma^2}\langle \mathbf{z}, t\mathbf{u}\rangle\right) \\
&= \sum_{\mathbf{z}\in\mathbb{Z}^m} \int_{A_\mathbf{z}} \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(\tfrac{-\|\mathbf{x}\|^2}{2\sigma^2}\right) d\mathbf{x} \exp\left(\tfrac{1}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, t\mathbf{u}\rangle\right) \\
&\leq \sum_{\mathbf{z}\in\mathbb{Z}^m} \int_{A_\mathbf{z}} \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(\tfrac{-\|\mathbf{x}\|^2}{2\sigma^2}\right) \exp\left(\tfrac{1}{\sigma^2}\langle \mathbf{x}, t\mathbf{u}\rangle\right) d\mathbf{x} \\
&= \sum_{\mathbf{z}\in\mathbb{Z}^m} \int_{A_\mathbf{z}} \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(\tfrac{-\|\mathbf{x}-t\mathbf{u}\|^2}{2\sigma^2}\right) \exp\left(\tfrac{t^2\|\mathbf{u}\|^2}{2\sigma^2}\right) d\mathbf{x} \\
&= \sum_{\mathbf{z}\in\mathbb{Z}^m} R_{t\mathbf{u},\sigma}^m(\mathbf{z}) \exp\left(\tfrac{t^2\|\mathbf{u}\|^2}{2\sigma^2}\right) \\
&= \exp\left(\tfrac{t^2\|\mathbf{u}\|^2}{2\sigma^2}\right),
\end{aligned}$$

where the last equality follows from the fact that $\sum_{\mathbf{z}\in\mathbb{Z}^m} R_{t\mathbf{u},\sigma}^m(\mathbf{z}) = 1$ because it is the sum over the entire range of the probability density function. We proceed to prove the claim of the lemma by applying Markov's inequality first and then the above result. For any $t > 0$, we have:

$$\begin{aligned}
\Pr\left[\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle > r\right] &= \Pr\left[\exp\left(\tfrac{t}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right) > \exp\left(tr/\sigma^2\right)\right] \\
&\leq \left(E\left[\exp\left(t\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle/\sigma^2\right)\right]\right)/\left(\exp\left(tr/\sigma^2\right)\right) \\
&\leq \exp\left((t^2\|\mathbf{u}\|^2 - 2tr)/(2\sigma^2)\right).
\end{aligned}$$

The function on the right assumes its maximum at $t = r/\|\mathbf{u}\|^2$, so we get $\Pr\left[\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle > r\right] \leq \exp\left(-r^2/(2\|\mathbf{u}\|^2\sigma^2)\right)$. Because the distribution is symmetric around the origin we also know $\Pr[\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle < -r] \leq \exp\left(-r^2/(2\|\mathbf{u}\|^2\sigma^2)\right)$. By applying the union bound to the two inequalities, we get the probability for $|\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle| > r$, which results in the claim of the lemma. □

Lemma A.2. *Under the conditions of Lemma A.1 we have:*

1. *For any* $k\sigma > 1/4(\sigma+1), \sigma \geq 1, \Pr\left[|z| > k\sigma; z \xleftarrow{\$} R_\sigma^1\right] \leq 2e^{\frac{-\left(k-\frac{1}{2}\right)^2}{2}}$.
2. *For any* $\mathbf{z} \in \mathbb{Z}^m$ *and* $\sigma \geq \sqrt{2/\pi}, R_\sigma^m(\mathbf{z}) \leq 2^{-m}$.
3. *For any* $k > 1, \Pr\left[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} R_\sigma^m\right] < 2k^m e^{\frac{m}{2}(1-k^2)}$.

Proof. Item 1 follows from Lemma A.1 by substituting $m = 1, r = k\sigma - \frac{1}{2}$ and $u = 1$. This gives

$$|z + y| = |z| - \frac{1}{2} > r = k\sigma - \frac{1}{2}.$$

In other words, $|z| > k\sigma$. Then we have for the upper bound of the probability:

$$2\exp\left(-\frac{r^2}{2\|u\|^2\sigma^2}\right) = 2\exp\left(-\frac{\left(k\sigma - \frac{1}{2}\right)^2}{2\sigma^2}\right) \leq 2\exp\left(-\frac{\left(k - \frac{1}{2}\right)^2\sigma^2}{2\sigma^2}\right),$$

where we use $-\left(k\sigma - \frac{1}{2}\right)^2 \leq -\left(k - \frac{1}{2}\right)^2\sigma^2$ for $\sigma \geq 1$ in the inequality. Note that for $0.44 < k < 1.89$ item 3 actually provides a better bound.

To prove Item 2, we write

$$\begin{aligned} R_\sigma^m(\mathbf{z}) &= \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_\mathbf{z}} e^{-\|\mathbf{x}\|^2/(2\sigma^2)} d\mathbf{x} \\ &\leq \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \cdot \max_{\mathbf{x}\in A_\mathbf{z}} e^{-\|\mathbf{x}\|^2/(2\sigma^2)} \cdot \mathrm{vol}(A_\mathbf{z}) \leq \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m, \end{aligned}$$

where the first inequality follows from the fact that integrating a continuous function on a bounded area is bounded from above by the maximum of the function on the area times the volume of the area. The second inequality follows from the fact that the volume of the area $A_\mathbf{z}$ is equal to 1 and $e^{-\|\mathbf{x}\|^2/(2\sigma^2)} \leq 1$ for all $\mathbf{x} \in A_\mathbf{z}$ for all $\mathbf{z} \in \mathbb{Z}^m$. Thus if $\sigma \geq \sqrt{2/\pi}$, we have $R_\sigma^m \leq 2^{-m}$.

For Item 3, we write the following:

$$\begin{aligned} &\Pr\left[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} R_\sigma^m\right] \\ &\quad = \sum_{\mathbf{z}\in\mathbb{Z}^m, \|\mathbf{z}\|>k\sigma\sqrt{m}} \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \int_{A_\mathbf{z}} e^{-\|\mathbf{x}\|^2/(2\sigma^2)} d\mathbf{x} \\ &\quad \leq \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \sum_{\mathbf{z}\in\mathbb{Z}^m, \|\mathbf{z}\|>k\sigma\sqrt{m}} \left(\max_{\mathbf{x}\in A_\mathbf{z}} e^{-\|\mathbf{x}\|^2/(2\sigma^2)} \cdot \mathrm{vol}(A_\mathbf{z})\right) \\ &\quad \leq \left(\tfrac{1}{\sqrt{2\pi\sigma^2}}\right)^m \sum_{\mathbf{z}\in\mathbb{Z}^m, \|\mathbf{z}\|>k\sigma\sqrt{m}} e^{-\|\mathbf{z}+\mathbf{y}\|^2/(2\sigma^2)}, \end{aligned} \tag{2}$$

where $\mathbf{y} \in [-\frac{1}{2}, \frac{1}{2}]^m$ is chosen such that the maximum is attained, i.e. for each z_i we pick $y_i, i = 1, \ldots, m$ in the following way:

$$y_i = \begin{cases} -\frac{1}{2} & \text{if } z_i > 0, \\ 0 & \text{if } z_i = 0, \\ \frac{1}{2} & \text{if } z_i < 0. \end{cases} \tag{3}$$

We use the second part of a lemma by Banaszczyk [2, Lemma 1.5], saying that for each $c \geq 1/\sqrt{2\pi}$, lattice L of dimension m and $\mathbf{u} \in \mathbb{R}^m$, we have $\sum_{\mathbf{z}\in L, \|\mathbf{z}\|>c\sqrt{m}} e^{-\pi\|\mathbf{z}+\mathbf{u}\|^2} < 2\left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^n \sum_{\mathbf{z}\in L} e^{-\pi\|\mathbf{z}\|^2}$, and put $\mathbf{u} = \mathbf{y}$. If we scale the lattice L by a factor of $1/s$ for some constant s, we have that for all s,

$$\sum_{\mathbf{z}\in L, \|\mathbf{z}\|>cs\sqrt{m}} e^{-\pi\|\mathbf{z}+\mathbf{y}\|^2/s^2} < 2\left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^m \sum_{\mathbf{z}\in L} e^{-\pi\|\mathbf{z}\|^2/s^2}.$$

Setting $L = \mathbb{Z}^m$ and $s = \sqrt{2\pi}\sigma$, we obtain

$$\sum_{\mathbf{z}\in\mathbb{Z}^m, \|\mathbf{z}\|>c\sqrt{2\pi\sigma^2 m}} e^{-\|\mathbf{z}+\mathbf{y}\|^2/(2\sigma^2)} < 2\left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^m \sum_{\mathbf{z}\in\mathbb{Z}^m} e^{-\|\mathbf{z}\|^2/(2\sigma^2)}.$$

Finally, by setting $c = k/\sqrt{2\pi}$ in the upper bound for the probability and applying it to Equation (2), we get

$$\Pr\left[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} R_\sigma^m\right] < 2k^m e^{\frac{m}{2}(1-k^2)} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \sum_{\mathbf{z}\in\mathbb{Z}^m} e^{-\|\mathbf{z}\|^2/(2\sigma^2)}.$$

Note that $\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \sum\limits_{\mathbf{z}\in\mathbb{Z}^m} \exp(-\|\mathbf{z}\|^2/(2\sigma^2)) = 1$, since it is the probability density function $R_\sigma^m(\mathbf{z})$ summed over all possible values. Thus we have

$$\Pr\left[\|\mathbf{z}\| > k\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} R_\sigma^m\right] < 2k^m e^{\frac{m}{2}(1-k^2)}.$$

□

The following is the proof of Lemma 3.1 from Section 3.

Proof. By definition we have

$$\begin{aligned}\frac{R_\sigma^m(\mathbf{z})}{R_{\mathbf{v},\sigma}^m(\mathbf{z})} &= \frac{\int_{A_\mathbf{z}} \rho_\sigma^m(\mathbf{x})d\mathbf{x}}{\int_{A_\mathbf{z}} \rho_{\mathbf{v},\sigma}^m(\mathbf{x})d\mathbf{x}} = \frac{\int_{A_\mathbf{z}} \exp(-\|\mathbf{x}\|^2/(2\sigma^2))d\mathbf{x}}{\int_{A_\mathbf{z}} \exp(-\|\mathbf{x}-\mathbf{v}\|^2/(2\sigma^2))d\mathbf{x}} \\ &\le \frac{\max\limits_{\mathbf{x}\in A_\mathbf{z}} e^{-\|\mathbf{x}\|^2/(2\sigma^2)} \cdot \mathrm{vol}(A_\mathbf{z})}{\min\limits_{\mathbf{x}\in A_\mathbf{z}} e^{-\|\mathbf{x}-\mathbf{v}\|^2/(2\sigma^2)} \cdot \mathrm{vol}(A_\mathbf{z})} = \frac{\exp(-\|\mathbf{z}+\mathbf{y}_1\|^2/(2\sigma^2))}{\exp(-\|\mathbf{z}-\mathbf{v}+\mathbf{y}_2\|^2/(2\sigma^2))},\end{aligned}$$

where the inequality follows from the fact that integrating a continuous function on a bounded area is bounded from below by its minimum on the area times the volume of the area; $\mathbf{y}_1 \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ is chosen such that the maximum is achieved for $\|\mathbf{z}+\mathbf{y}_1\|^2$, and $\mathbf{y}_2 \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ is chosen such that the minimum is achieved for $\|\mathbf{z}-\mathbf{v}+\mathbf{y}_2\|^2$. In other words, $\mathbf{y}_1 \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ is defined as in Equation (3) and for $\mathbf{y}_2 \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ we have for each $z_i - v_i, i = 1, \dots, m$:

$$y_{2,i} = \begin{cases} -\frac{1}{2} & \text{if } z_i < v_i, \\ \frac{1}{2} & \text{if } z_i \ge v_i. \end{cases} \tag{4}$$

This results in the following formula:

$$\frac{e^{-\|\mathbf{z}+\mathbf{y}_1\|^2/(2\sigma^2)}}{e^{-\|\mathbf{z}-\mathbf{v}+\mathbf{y}_2\|^2/(2\sigma^2)}} \exp\left(\frac{(\|\mathbf{y}_2\|^2 - \|\mathbf{y}_1\|^2 + 2\langle\mathbf{z}, \mathbf{y}_2 - \mathbf{y}_1\rangle) - 2\langle\mathbf{z}+\mathbf{y}_2, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\sigma^2}\right).$$

We want to combine $\|\mathbf{y}_2\|^2 - \|\mathbf{y}_1\|^2 + 2\langle\mathbf{z}, \mathbf{y}_2 - \mathbf{y}_1\rangle$ with the inner product $\langle\mathbf{z}+\mathbf{y}_2, \mathbf{v}\rangle$ into an inner product of the form $\langle\mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle$ for some $\mathbf{a}$, where

$\mathbf{y} \in [-1/2, 1/2]^m$ minimizes $\langle \mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle$, such that we can apply Lemma A.1, where we set $\mathbf{u} = \mathbf{v}+\mathbf{a}$. We can write

$$\|\mathbf{y}_2\|^2 - \|\mathbf{y}_1\|^2 + 2\langle \mathbf{z}, \mathbf{y}_2 - \mathbf{y}_1\rangle = \sum_{i=1}^{m} \left(y_{2,i}^2 - y_{1,i}^2 + 2z_i\left(y_{2,i} - y_{1,i}\right)\right).$$

Using the definition of $y_{1,i}$ and $y_{2,i}$, for $i = 1, \ldots, m$ we get the following expression:

$$y_{2,i}^2 - y_{1,i}^2 + 2z_i\left(y_{2,i} - y_{1,i}\right) = \begin{cases} = -2z_i & \text{if } z_i < v_i \wedge z_i < 0, \\ = \frac{1}{4} & \text{if } z_i = 0, \\ = 2z_i & \text{if } z_i \geq v_i \wedge z_i > 0, \\ = 0 & \text{otherwise.} \end{cases} \tag{5}$$

To create an upper bound of the form $-2\langle \mathbf{z}+\mathbf{y}, \mathbf{a}\rangle$, where $\mathbf{y} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ minimizes $\langle \mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle$, we need to determine an expression for $\mathbf{a}$, i.e. we determine a_i such that it fits Equation (5). This gives us the following expressions for the coordinates $i = 1, \ldots, m$:

$$-2a_iz_i - 2a_iy_i = \begin{cases} -2a_iz_i + a_i & \text{if } z_i < 0, \\ -a_i & \text{if } z_i = 0, \\ -2a_iz_i - a_i & \text{if } z_i > 0. \end{cases} \quad \Rightarrow \quad a_i = \begin{cases} -\frac{2z_i}{-2z_i+1} & \text{if } z_i < 0, \\ -\frac{1}{4} & \text{if } z_i = 0, \\ -\frac{2z_i}{2z_i+1} & \text{if } z_i > 0. \end{cases}$$

Now we can write $\sum_{i=1}^{m} \left(y_{2,i}^2 - y_{1,i}^2 + 2z_i\left(y_{2,i} - y_{1,i}\right)\right) \leq -2\langle \mathbf{z}+\mathbf{y}, \mathbf{a}\rangle$, where $\mathbf{a}$ is chosen as above such that $-z_ia_i \leq 0$ and $|a_i| \leq 1$ for $i = 1, \ldots, m$ and $\mathbf{y}$ minimizes $\langle \mathbf{z}+\mathbf{y}, \mathbf{a}\rangle$. Given $\mathbf{y}_2$ and $\mathbf{y}$, we can write $\mathbf{y}_2 = \mathbf{y}+\mathbf{b}$, where we pick $b_i \in \{-1, 0, 1\}$ for $i = 1, \ldots, m$ such that the equation holds. Then we can write $2\langle \mathbf{z}+\mathbf{y}_2, \mathbf{v}\rangle = 2\langle \mathbf{z}+\mathbf{y}, \mathbf{v}\rangle + 2\langle \mathbf{b}, \mathbf{v}\rangle$. We have $|2\langle \mathbf{b}, \mathbf{v}\rangle| = \left|\sum_{i=1}^{m} 2b_iv_i\right| \leq 2\|\mathbf{v}\|^2$, because $b_i \in \{-1, 0, 1\}$, dependent on the value of z_i and v_i. Combining these bounds and applying them to the previous result, gives us

$$\begin{aligned} &\exp\left(\left(\left(\|\mathbf{y}_2\|^2 - \|\mathbf{y}_1\|^2 + 2\langle \mathbf{z}, \mathbf{y}_2 - \mathbf{y}_1\rangle\right) - 2\langle \mathbf{z}+\mathbf{y}_2, \mathbf{v}\rangle + \|\mathbf{v}\|^2\right)/(2\sigma^2)\right) \\ &\leq \exp\left(\left(-2\langle \mathbf{z}+\mathbf{y}, \mathbf{a}\rangle - 2\langle \mathbf{z}+\mathbf{y}, \mathbf{v}\rangle - 2\langle \mathbf{b}, \mathbf{v}\rangle + \|\mathbf{v}\|^2\right)/(2\sigma^2)\right) \\ &\leq \exp\left(\left(-2\langle \mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle + 3\|\mathbf{v}\|^2\right)/(2\sigma^2)\right). \end{aligned}$$

Lemma A.1 tells us that $|\langle \mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle| \leq \sigma\sqrt{2\log m}\|\mathbf{v}+\mathbf{a}\|$ with probability at least $1 - 2^{-\log m}$ if $\mathbf{y}$ minimizes $\langle \mathbf{z}+\mathbf{y}, \mathbf{v}+\mathbf{a}\rangle$ and if $\mathbf{v}+\mathbf{a} \in \mathbb{Z}^m$. Since both conditions hold, we have

$$\begin{aligned} &\exp\left(\frac{-2\langle \mathbf{z}+\mathbf{y}_2, \mathbf{v}+\mathbf{a}\rangle + 3\|\mathbf{v}\|^2}{2\sigma^2}\right) < \exp\left(\frac{2\sqrt{2\log m}\|\mathbf{v}+\mathbf{a}\| + 3\|\mathbf{v}\|^2}{2\sigma^2}\right) \\ &\leq \exp\left(\frac{\sqrt{2\log m}\|\mathbf{v}+\mathbf{a}\|}{\sqrt{\log m}\|\mathbf{v}\|} + \frac{3\|\mathbf{v}\|^2}{2\log m\|\mathbf{v}\|^2}\right) = \exp\left(\frac{3\|\mathbf{v}\| + 2\sqrt{2}\log m\|\mathbf{v}+\mathbf{a}\|}{2\log m\|\mathbf{v}\|}\right) = O(1), \end{aligned}$$

where the second inequality uses $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$ and the final equality uses $\|\mathbf{a}\|^2$ being small. $\square$

A.1 Comparison of Proofs for Rounded Gaussians vs. Discrete Gaussians

As we have mentioned at the beginning of this section, the theorems and proofs follow the line of the theorems and proofs of Lyubashevsky [16] closely. Here we give a quick overview of the changes made in the lemmas and theorems next to replacing the discrete Gaussian with the rounded Gaussian. We do not state in detail where the proofs differ, since we require different techniques to end up with similar results.

In Lemma A.1 we use $\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle$ with $\mathbf{y} \in \left[-\frac{1}{2}, \frac{1}{2}\right]^m$ minimizing $\exp\left(\frac{1}{\sigma^2}\langle \mathbf{z}+\mathbf{y}, \mathbf{u}\rangle\right)$ instead of the $\langle \mathbf{z}, \mathbf{u}\rangle$ that is used in [16, Lemma 4.3].

In Lemma A.2 we require for Item 1 that $k\sigma > 1/4(\sigma+1)$ and $\sigma \geq 1$ instead of the $k > 0$ from [16, Lemma 4.4]. Next to that, we get that the probability $< \exp\left(\frac{-\left(k-\frac{1}{2}\right)^2}{2}\right)$ instead of the $< \exp\left(\frac{-k^2}{2}\right)$. For Item 2 we have $\sigma \geq \sqrt{2/\pi}$ instead of $\sigma \geq 3/\sqrt{2\pi}$. For Item 3 we have $2k^m e^{\frac{m}{2}\left(1-k^2\right)}$ instead of $k^m e^{\frac{m}{2}\left(1-k^2\right)}$.

Theorem 3.1 follows through directly based on the previous lemmas.

B Rényi Divergence

An adversary wins if within q_s signing queries he can distinguish the perfect scheme and an implementation thereof or if he breaks the scheme with the perfect implementation. We will upper bound the success probability of any such adversary dependent on the precision used in the computation.

First we analyze the statistical distance (SD) and then Rényi divergences (RD) of order 1 and ∞ (Definition 5.1). Based on [1] we expect a lower precision requirement from the RD analysis. We use the definition of Rényi divergence as given in [1] and copy the relevant properties of RD from there; see [25] for a proof of the following lemmas and note that the definitions agree up to taking logarithms. For completeness we include the statistical difference.

Definition B.1. *The* statistical distance $\Delta(P;Q)$ *between two discrete probability functions P and Q is defined by*

$$\Delta(P;Q) = \frac{1}{2}\sum_{x\in V} |P(x) - Q(x)|,$$

where $V = \operatorname{Supp}(P) \cup \operatorname{Supp}(Q)$ denotes the union of the support of P and the support of Q.

Definition B.2. *For any two discrete probability distributions P and Q, such that* $\operatorname{Supp}(P) \subseteq \operatorname{Supp}(Q)$ *the* Rényi divergences of order 1 *is defined by*

$$\mathrm{RD}_1(P \,||\, Q) = \exp\left(\sum_{x\in\operatorname{Supp}(P)} P(x)\log\frac{P(x)}{Q(x)}\right).$$

For RD the measures are related multiplicatively.

Lemma B.1 (Multiplicativity). *Let $a \in \{1, +\infty\}$. Let P and Q be two distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$ *of a pair of random variables (Y_1, Y_2) and let Y_1 and Y_2 be independent.*

Then we have: $\mathrm{RD}_a(P \,||\, Q) = \mathrm{RD}_a(P_1 \,||\, Q_1) \cdot \mathrm{RD}_a(P_2 \,||\, Q_2)$.

We will use the following *probability preservation* property to quantify the probability of distinguishing the perfect rounded Gaussian distribution from the one implemented with finite precision.

Lemma B.2 (Probability Preservation). *Let P and Q denote distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. *Let $A \subseteq \mathrm{Supp}(Q)$ be an arbitrary event. Then $Q(A) \geq P(A) / R_\infty(P \,||\, Q)$.*

B.1 Precision for Rounded Gaussians

We now give a formal analysis linking the precision p of the implementation to the security level of the signature scheme. Computing with floating-point precision p means that the intermediate value x will be output with a certain *error* η. We can write this as $x' = x + \eta$, with $|\eta| \leq 2^{-p}x$. After this, x' is rounded to the nearest integer, i.e. $z = \lfloor x' \rceil$. Note that this implies that for computing the probability of sampling z only the interval changes from $[z - \frac{1}{2}, z + \frac{1}{2})$ to $[z - \frac{1}{2} - e_l, z + \frac{1}{2} + e_r)$, with $|e_l| \leq 2^{-p}\left|z - \frac{1}{2}\right|$ and $|e_r| \leq 2^{-p}\left|z + \frac{1}{2}\right|$. The tail cut forces $|z| \leq \tau\sigma$ and for $\tau = O(\sqrt{\lambda})$ Lemma A.2 implies that $\exp\left(\frac{-(\tau - \frac{1}{2})^2}{2\sigma^2}\right) \approx 2^{-\lambda}$, i.e. with all but negligible probability the sampled value lies within the tail bound. For all practical values $\lambda \ll 2^p$.

First we analyze the SD to gain a basic understanding of the precision needed for our sampler in BLISS. After this we analyze two different kinds of RD, since we expect that the required floating point precision will be smaller, because the bounds are tighter for other samplers. At the end of this section, we compare all of these bounds on the precision.

SD-based analysis. We follow [1] in assuming that any forging adversary $\mathcal{A}$ with success probability $\leq \delta$ on the scheme implemented with the perfect rounded Gaussian sampling has a success probability $\epsilon \leq \delta + \Delta({R'}_\sigma^{mq_s}; R_\sigma^{mq_s})$ against the scheme implemented with the truncated rounded Gaussian sampling, with $R_\sigma^{mq_s}$, i.e. the success probability ϵ on the truncated scheme is upper bounded by the success probability on the perfect scheme δ and the extra information we gain by comparing the distributions ${R'}_\sigma^{mq_s}$ and $R_\sigma^{mq_s}$. For a target success probability ϵ we have to choose $\delta \leq \epsilon/2$ for the success probability on the perfect scheme and we want to determine the lower bound on p such that $\Delta({R'}_\sigma^{mq_s}; R_\sigma^{mq_s}) \leq \epsilon/2$.

By the union bound this means that we require $\Delta({R'}_\sigma; R_\sigma) \leq \epsilon/(mq_s)$. We only look at values between the tail bounds, i.e. $z \in [-\tau\sigma, \tau\sigma]$, since any element lying outside of the tail bounds is rejected and thus not in the support of R'_σ. Next to that, we assume that $e_r, e_l \leq 2^{-p}\tau\sigma$, which is the worst case setting.

$$\begin{aligned}
&\Delta(R'^1_\sigma(z); R^1_\sigma(z)) \\
&\quad = \tfrac{1}{2} \sum_{z=-\tau\sigma}^{\tau\sigma} \left| \int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx - \int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx \right| \\
&\quad \leq \tfrac{1}{2} \sum_{z=-\tau\sigma}^{\tau\sigma} \left| \int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx \right| \\
&\quad \leq \tfrac{1}{2} \tfrac{1}{\sqrt{2\pi\sigma^2}} \left(\sum_{z=-\tau\sigma}^{-1} \left| |e_l| \exp\left(\tfrac{-\left(z-\frac{1}{2}\right)^2}{2\sigma^2} \right) + |e_r| \exp\left(\tfrac{-\left(z+\frac{1}{2}+|e_r|\right)^2}{2\sigma^2} \right) \right| \right. \\
&\qquad\qquad \left. + |e_l| + |e_r| + \sum_{z=1}^{\tau\sigma} \left| |e_l| \exp\left(\tfrac{-\left(z-\frac{1}{2}-|e_l|\right)^2}{2\sigma^2} \right) + |e_r| \exp\left(\tfrac{-\left(z+\frac{1}{2}\right)^2}{2\sigma^2} \right) \right| \right) \\
&\quad \leq \tfrac{1}{2} \tfrac{2^{-p}\tau\sigma}{\sqrt{2\pi\sigma^2}} \left(\sum_{z=-\tau\sigma}^{-1} \left| \exp\left(\tfrac{-\left(z-\frac{1}{2}\right)^2}{2\sigma^2} \right) + \exp\left(\tfrac{-\left(z+\frac{1}{2}+2^{-p}\tau\sigma\right)^2}{2\sigma^2} \right) \right| + 2 \right. \\
&\qquad\qquad \left. + \sum_{z=1}^{\tau\sigma} \left| \exp\left(\tfrac{-\left(z-\frac{1}{2}-2^{-p}\tau\sigma\right)^2}{2\sigma^2} \right) + \exp\left(\tfrac{-\left(z+\frac{1}{2}\right)^2}{2\sigma^2} \right) \right| \right) \\
&\quad \leq \tfrac{2^{-p}\tau\sigma}{\sqrt{2\pi\sigma^2}} \left(1 + \sum_{z=1}^{\tau\sigma} \left(\exp\left(\tfrac{-\left(z-\frac{1}{2}-2^{-p}\tau\sigma\right)^2}{2\sigma^2} \right) + \exp\left(\tfrac{-\left(z+\frac{1}{2}\right)^2}{2\sigma^2} \right) \right) \right),
\end{aligned}$$

where we use in the second to last inequality the assumption that $|e_l|, |e_r| \leq 2^{-p}\tau\sigma$ and in the last inequality we note that for $z < 0$ we have $\exp\left(-\frac{\left(z-\frac{1}{2}\right)^2}{2\sigma^2}\right) = \exp\left(-\frac{\left(|z|+\frac{1}{2}\right)^2}{2\sigma^2}\right)$, which matches the term in the sum for $z > 0$. Similarly we have $\exp\left(-\frac{\left(z+\frac{1}{2}+2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right) = \exp\left(-\frac{\left(|z|-\frac{1}{2}-2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right)$. This means that we can group both sums under one sum running from 1 to $\tau\sigma$, which we need to multiply by 2 to compensate for having both distributions in one sum.

Note that this result looks like a rounded Gaussian centered around $\frac{1}{2}$ and a rounded Gaussian centered around $\frac{1}{2}+2^{-p}\tau\sigma$, except that all values for $z \leq 0$ are missing. Due to the symmetric property of the rounded Gaussian distribution, we know that both rounded Gaussians sum up to $\leq \frac{1}{2}$. This gives us:

$$\begin{aligned}
&\tfrac{2^{-p}\tau\sigma}{\sqrt{2\pi\sigma^2}} \left(1 + \sum_{z=1}^{\tau\sigma} \left(\exp\left(\tfrac{-\left(z-\frac{1}{2}-2^{-p}\tau\sigma\right)^2}{2\sigma^2} \right) + \exp\left(\tfrac{-\left(z+\frac{1}{2}\right)^2}{2\sigma^2} \right) \right) \right) \\
&\qquad \leq 2^{-p}\tau\sigma \left(\tfrac{1}{\sqrt{2\pi\sigma^2}} + \tfrac{1}{2} + \tfrac{1}{2} \right) = 2^{-p}\tau\sigma \left(\tfrac{1}{\sqrt{2\pi\sigma^2}} + 1 \right).
\end{aligned}$$

We require $2^{-p}\tau\sigma\left(\frac{1}{\sqrt{2\pi\sigma^2}} + 1\right) \leq (\epsilon/2)/(mq_s)$. Note that $0 < \epsilon < 1$ and thus that $\log \epsilon < 0$. This means that a smaller ϵ requires a higher level of floating point precision. This is what we expect; if we want an adversary $\mathcal{A}$ to be less likely to be successful, we need to be more precise in our computations.

If we use the common setting $\epsilon = 2^{-\lambda}$, we get the precision requirement

$$p \geq \log\left(mq_s\tau\sigma\left(\sqrt{2\pi\sigma^2}+1\right)\right) + \lambda - \log\left(\sqrt{2\pi\sigma^2}\right) + 1. \tag{6}$$

RD$_1$-based analysis. According to [1], if $a = 1$ we have for an arbitrary event $A \subseteq \mathrm{Supp}(Q)$ that $Q(A) \geq P(A) - \sqrt{\ln \mathrm{RD}_1(P \mid\mid Q)/2}$, which is the probability preservation property (Lemma B.2) for $a = 1$. This means that we have $\delta \geq \epsilon - \sqrt{\ln \mathrm{RD}_1\left({R'}_\sigma^{mq_s} \mid\mid R_\sigma^{mq_s}\right)/2}$. We follow [1] in bounding the right-hand side by $\epsilon/2$. By the multiplicative property of the RD over the mq_s independent samples needed for signing q_s times, we get $\mathrm{RD}_1\left({R'}_\sigma^{mq_s} \mid\mid R_\sigma^{mq_s}\right) \leq \left(\mathrm{RD}_1\left({R'}_\sigma^1 \mid\mid R_\sigma^1\right)\right)^{mq_s}$.

Recall that for the ln function we have $\ln(x) \leq x - 1$ for $x > 0$. Note that we are working with positive numbers, since probabilities lie between zero and one. If we only look at the elements between $-\tau\sigma$ and $\tau\sigma$, we know that they have a probability > 0. Now we compute the 1-dimensional case.

$$
\begin{aligned}
&\ln \mathrm{RD}_1\left({R'}_\sigma^1 \mid\mid R_\sigma^1\right)\\
&\quad= \sum_{z \in \mathrm{Supp}({R'}_\sigma^1)} {R'}_\sigma^1(z) \ln\left(\frac{{R'}_\sigma^1(z)}{R_\sigma^1(z)}\right) \leq \sum_{z \in \mathrm{Supp}({R'}_\sigma^1)} {R'}_\sigma^1(z)\left(\frac{{R'}_\sigma^1(z)}{R_\sigma^1(z)} - 1\right)\\
&\quad\leq \sum_{z \in \mathrm{Supp}({R'}_\sigma^1)} \frac{1}{\sqrt{2\pi\sigma^2}} \int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \left(\frac{\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx} - 1\right)\\
&\quad\leq \sum_{z \in \mathrm{Supp}({R'}_\sigma^1)} \frac{1}{\sqrt{2\pi\sigma^2}} \int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \cdot\\
&\quad\left(\frac{\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}\right).
\end{aligned}
\tag{7}
$$

We now want to bound this equation. We first look at a bound in the case $z > 0$ for the following part of the equation:

$$
\begin{aligned}
&\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \left(\frac{\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}\right)\\
&\quad\leq (1 + e_l + e_r) \exp\left(\frac{-\left(z-\frac{1}{2}-e_l\right)^2}{2\sigma^2}\right) \exp\left(\frac{(z+\frac{1}{2})^2}{2\sigma^2}\right) \cdot\\
&\qquad\qquad \left(|e_l| \exp\left(\frac{-\left(z-\frac{1}{2}-|e_l|\right)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-\left(z+\frac{1}{2}\right)^2}{2\sigma^2}\right)\right)\\
&\quad\leq (1 + e_r + e_l)\left(|e_l| \exp\left(\frac{-\left(z+\frac{1}{2}-2(1+|e_l|)\right)^2 + 2(1+|e_l|)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-\left(z-\frac{1}{2}-e_l\right)^2}{2\sigma^2}\right)\right).
\end{aligned}
$$

If we can find an equivalent bound like this for $z < 0$ and for $z = 0$, we can use the above formula to bound Equation (7). For $z < 0$, we have the following equation that gives an upper bound:

$$\begin{aligned}
&\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \left(\frac{\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}\right)\\
&\quad\leq (1+e_l+e_r) \exp\left(\frac{-\left(z+\frac{1}{2}+e_r\right)^2}{2\sigma^2}\right) \exp\left(\frac{(z-\frac{1}{2})^2}{2\sigma^2}\right) \cdot\\
&\qquad\qquad \left(|e_l| \exp\left(\frac{-\left(z-\frac{1}{2}\right)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-\left(z+\frac{1}{2}+|e_r|\right)^2}{2\sigma^2}\right)\right)\\
&\quad\leq (1+e_r+e_l) \left(|e_l| \exp\left(\frac{-\left(z+\frac{1}{2}+e_r\right)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-\left(z-\frac{1}{2}+2(1+|e_r|)\right)^2+2(1+|e_r|)^2}{2\sigma^2}\right)\right)\\
&\quad\leq (1+e_r+e_l) \left(|e_l| \exp\left(\frac{-\left(|z|-\frac{1}{2}-e_r\right)^2}{2\sigma^2}\right) + \right.\\
&\qquad\qquad \left. |e_r| \exp\left(\frac{-\left(|z|+\frac{1}{2}-2(1+|e_r|)\right)^2+2(1+|e_r|)^2}{2\sigma^2}\right)\right).
\end{aligned}$$

This means that we have the same result for $z > 0$ and $z < 0$, except that the e_l's change into e_r's and vice versa. Since $e_l, e_r \leq 2^{-p}\tau\sigma$, we end up with the following result for $z < 0$ and $z > 0$:

$$\begin{aligned}
&\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \left(\frac{\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}\right)\\
&\quad\leq \left(1+2^{-p+1}\tau\sigma\right) 2^{-p}\tau\sigma \left(\exp\left(\frac{-\left(|z|-\frac{1}{2}-2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right) + \right.\\
&\qquad\qquad \left. \exp\left(\frac{-\left(|z|+\frac{1}{2}-2\left(1+2^{-p}\tau\sigma\right)\right)^2+2\left(1+2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right)\right).
\end{aligned}$$

Now that we have found a bound for $z < 0$ and $z > 0$, we also need to find a bound for $z = 0$. If $z = 0$, we have

$$\begin{aligned}
&\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \left(\frac{\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}{\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx}\right)\\
&\quad\leq (1+e_l+e_r) \exp\left(\frac{1}{8\sigma^2}\right) \left(|e_l| \exp\left(-\frac{1}{8\sigma^2}\right) + |e_r| \exp\left(-\frac{1}{8\sigma^2}\right)\right)\\
&\quad= (1+e_l+e_r)\left(|e_l|+|e_r|\right) \leq \left(1+2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma,
\end{aligned}$$

where we use $e_l, e_r < 2^{-p}\tau\sigma$ in the second inequality. Combining the result for $z = 0$ with the results for $z < 0$ and $z > 0$ gives us:

$$\begin{aligned}
&\ln \mathrm{RD}_1\left({R'}^1_\sigma \mid\mid R^1_\sigma\right) \\
&\quad\leq \left(1 + 2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma \\
&\quad+ \sum_{z \in \mathrm{Supp}({R'}^1_\sigma), z>0} \tfrac{1}{\sqrt{2\pi\sigma^2}} \left(1 + 2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma \Bigg(\exp\left(\tfrac{-\left(|z| - \frac{1}{2} - 2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right) + \\
&\qquad \exp\left(\tfrac{-\left(|z| + \frac{1}{2} - 2\left(1 + 2^{-p}\tau\sigma\right)\right)^2 + 2\left(1 + 2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right)\Bigg) \\
&\quad= \left(1 + 2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma \Bigg(1 + \sum_{z=0}^{\infty} \tfrac{1}{\sqrt{2\pi\sigma^2}} \Bigg(\exp\left(\tfrac{-\left(z - \frac{1}{2} - 2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right) + \\
&\qquad \exp\left(\tfrac{-\left(z + \frac{1}{2} - 2\left(1 + 2^{-p}\tau\sigma\right)\right)^2 + 2\left(1 + 2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right)\Bigg)\Bigg) \\
&\quad\leq \left(1 + 2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma \left(2 + 2\exp\left(\tfrac{9}{4\sigma^2}\right)\right),
\end{aligned}$$

where we use in the last inequality that $\sum_{z=0}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\left(z - \frac{1}{2} - 2^{-p}\tau\sigma\right)^2}{2\sigma^2}\right) \leq 1$, as this sums over parts of a Gaussian centered at $-1/2 - 2^{-p}\tau\sigma$. Similarly, $\sum_{z=0}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-\left(z + \frac{1}{2} - 2\left(1 + 2^{-p}\tau\sigma\right)\right)^2}{2\sigma^2}\right) \leq 1$. and $1 < (1 + 2^{-p}\tau\sigma) < \frac{3}{2}$, since $0 < 2^{-p}\tau\sigma < \frac{1}{2}$. We note that we could use the stronger bound $\tau\sigma < 2^{-p/2+1}$ here, which implies that we can use a smaller number in the exp function. However, the goal is to get rid of p with this equation and for this the current estimate is sufficient. This means that we can use the equation above to compute the floating point precision needed in the RD_1 setting. First we look at $\ln \mathrm{RD}_1({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma)/2$, before we determine the precision p:

$$\begin{aligned}
\ln \mathrm{RD}_1({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma)/2 &\leq mq_s \ln \mathrm{RD}_1({R'}^1_\sigma \mid\mid R^1_\sigma)/2 \\
&\leq \tfrac{mq_s}{2} \left(1 + 2^{-p+1}\tau\sigma\right) 2^{-p+1}\tau\sigma \left(2 + 2\exp\left(\tfrac{9}{4\sigma^2}\right)\right) \\
&= mq_s \left(\left(2^{-p+1}\tau\sigma + \tfrac{1}{2}\right)^2 - \tfrac{1}{4}\right) \left(1 + \exp\left(\tfrac{9}{4\sigma^2}\right)\right).
\end{aligned}$$

If we now bound this expression by $\epsilon^2/4$ and determine p, we know that this p also holds in the setting $\sqrt{\ln \mathrm{RD}_1({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma)/2} \leq \epsilon/2$. This results in:

$$\begin{aligned}
& mq_s \left(\left(2^{-p+1}\tau\sigma + \tfrac{1}{2}\right)^2 - \tfrac{1}{4}\right) \left(1 + \exp\left(\tfrac{9}{4\sigma^2}\right)\right) \leq \tfrac{\epsilon^2}{4} \\
\Leftrightarrow\ & \left(2^{-p+1}\tau\sigma + \tfrac{1}{2}\right)^2 \leq \tfrac{\epsilon^2 + mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)}{4mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)} \\
\Leftrightarrow\ & 2^{-p+1} \leq \tfrac{\sqrt{\epsilon^2 + mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)} - \sqrt{mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)}}{2\tau\sigma\sqrt{mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)}}.
\end{aligned}$$

This means that we have as the floating point precision requirement

$$p \geq \log\left(\frac{\tau\sigma\sqrt{mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)}}{\sqrt{\epsilon^2 + mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)} - \sqrt{mq_s\left(1 + \exp\left(\frac{9}{4\sigma^2}\right)\right)}}\right) + 2. \tag{8}$$

RD$_\infty$-based analysis. For $a = +\infty$, we follow [1] such that we have that any forging adversary $\mathcal{A}$ having success probability ϵ on the scheme implemented with imperfect rounded Gaussian sampling has a success probability $\delta \geq \epsilon / \mathrm{RD}_\infty({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma)$ on the scheme implemented with the perfect rounded Gaussian, because of the multiplicative property of the RD, as given in Lemma B.1. If $\mathrm{RD}_\infty({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma) \leq O(1)$, then $\delta = \Omega(\epsilon)$.

We need mq_s samples to create q_s signatures. By the multiplicative property of the RD, we have $\mathrm{RD}_\infty({R'}^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma) \leq \mathrm{RD}_\infty({R'}^1_\sigma \mid\mid R^1_\sigma)^{mq_s}$. We target $\delta \geq \epsilon / \exp(1)$. We first compute ${R'}^1_\sigma(z)/R^1_\sigma(z)$ from which the maximum will automatically follow:

$$\begin{aligned}\frac{{R'}^1_\sigma(z)}{R^1_\sigma(z)} &= \left(\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx\right) \Big/ \left(\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/(2\sigma^2)} dx\right) \\ &\leq 1 + \left(\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} e^{-x^2/(2\sigma^2)} dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} e^{-x^2/(2\sigma^2)} dx\right) \Big/ \left(\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} e^{-x^2/(2\sigma^2)} dx\right) .\end{aligned} \tag{9}$$

Now we need to find a lower bound for the integral in the denominator. We start by looking into the case $z > 0$. We have the following bounds:

$$\begin{aligned}\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} e^{-x^2/(2\sigma^2)} dx &\geq \int_{z-\frac{1}{2}}^{z-\frac{1}{2}+\frac{1}{z}} e^{-x^2/(2\sigma^2)} dx \geq \frac{1}{z} \exp\left(\frac{-(z-\frac{1}{2}+\frac{1}{z})^2}{2\sigma^2}\right) \\ &= \frac{1}{z} \exp\left(\frac{-(z-\frac{1}{2})^2 - 2(z-\frac{1}{2})\frac{1}{z} - \frac{1}{z^2}}{2\sigma^2}\right) \geq \frac{1}{z} \exp\left(\frac{-(z-\frac{1}{2})^2}{2\sigma^2}\right) \exp\left(\frac{-1}{\sigma^2}\right) ,\end{aligned} \tag{10}$$

where we use that $\frac{2}{z}(z-\frac{1}{2}) + \frac{1}{z^2} \leq 2$ for $z \geq 1$ and $z \in \mathbb{Z}$. We bound the integrals in the numerator the same way as in the RD$_1$ analysis and combine this with the lower bound from Equation (9):

$$\begin{aligned}&1 + \left(\int_{z-\frac{1}{2}-|e_l|}^{z-\frac{1}{2}} e^{-x^2/(2\sigma^2)} dx + \int_{z+\frac{1}{2}}^{z+\frac{1}{2}+|e_r|} e^{-x^2/(2\sigma^2)} dx\right) \Big/ \left(\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} e^{-x^2/(2\sigma^2)} dx\right) \\ &\leq 1 + \left(|e_l| \exp\left(\frac{-(z-\frac{1}{2}-|e_l|)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-(z+\frac{1}{2})^2}{2\sigma^2}\right)\right) \Big/ \left(\frac{1}{z} \exp\left(\frac{-(z-\frac{1}{2})^2}{2\sigma^2}\right) \exp\left(\frac{-1}{\sigma^2}\right)\right) \\ &= 1 + z \exp\left(\frac{1}{\sigma^2}\right) \exp\left(\frac{(z-\frac{1}{2})^2}{2\sigma^2}\right) \left(|e_l| \exp\left(\frac{-(z-\frac{1}{2}-|e_l|)^2}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-(z+\frac{1}{2})^2}{2\sigma^2}\right)\right) \\ &\leq 1 + z \exp\left(\frac{1}{\sigma^2}\right) \left(|e_l| \exp\left(\frac{|e_l|(2z-1-|e_l|)}{2\sigma^2}\right) + |e_r| \exp\left(\frac{-z}{\sigma^2}\right)\right) \\ &\leq 1 + z \exp\left(\frac{1}{\sigma^2}\right) \left(|e_l| \exp\left(\frac{|e_l| z}{\sigma^2}\right) + |e_r|\right) \leq 1 + 2^{-p}(\tau\sigma)^2 \exp\left(\frac{1}{\sigma^2}\right) \left(\exp\left(\frac{2^{-p}(\tau\sigma)^2}{\sigma^2}\right) + 1\right) ,\end{aligned}$$

where we use in the last inequality that $|e_l|, |e_r| \leq 2^{-p}\tau\sigma$ and that $|z| \leq \tau\sigma$. We note that $2^{-p+1} \leq (\tau\sigma)^2$, which gives us

$$\begin{aligned}1 + 2^{-p}(\tau\sigma)^2 \exp\left(\frac{1}{\sigma^2}\right) \left(\exp\left(\frac{2^{-p}(\tau\sigma)^2}{\sigma^2}\right) + 1\right) &\leq 1 + 2^{-p}(\tau\sigma)^2 \exp\left(\frac{1}{2\sigma^2}\right) \left(\exp\left(\frac{1}{2\sigma^2}\right) + 1\right) \\ &\leq \exp\left(2^{-p}(\tau\sigma)^2 \exp\left(\frac{1}{\sigma^2}\right) \left(\exp\left(\frac{1}{2\sigma^2}\right) + 1\right)\right) .\end{aligned}$$

We have found an upper bound for ${R'}^{mq_s}_\sigma / R^{mq_s}_\sigma$ if $z > 0$. We need to check if this bound works for any value of $z \in \mathbb{Z}$. First we look into the case $z < 0$. We want to find a similar bound as in Equation (10). We have

$$\begin{aligned}\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} e^{-x^2/(2\sigma^2)}dx &\geq \int_{z+\frac{1}{2}+\frac{1}{z}}^{z+\frac{1}{2}} e^{-x^2/(2\sigma^2)}dx \geq \tfrac{1}{|z|}\exp\left(\tfrac{-\left(z+\frac{1}{2}+\frac{1}{z}\right)^2}{2\sigma^2}\right)\\ &= \tfrac{1}{|z|}\exp\left(\tfrac{-\left(z+\frac{1}{2}\right)^2-2\left(z+\frac{1}{2}\right)\cdot\frac{1}{z}-\frac{1}{z^2}}{2\sigma^2}\right) = \tfrac{1}{|z|}\exp\left(\tfrac{-\left(|z|-\frac{1}{2}\right)^2-2\left(|z|-\frac{1}{2}\right)\cdot\frac{1}{|z|}-\frac{1}{|z|^2}}{2\sigma^2}\right)\\ &\geq \tfrac{1}{|z|}\exp\left(\tfrac{-\left(|z|-\frac{1}{2}\right)^2}{2\sigma^2}\right)\exp\left(\tfrac{-1}{\sigma^2}\right),\end{aligned} \tag{11}$$

which is the same expression as we had for $z > 0$. We note that the only difference between $z < 0$ and $z > 0$ is the e_l and the e_r, which we already have seen in the case of RD_1. Since we use $|e_l|, |e_r| \leq 2^{-p}\tau\sigma$, we can use the bound found for $z > 0$ also in the case $z < 0$. Now we check if this maximum also works for $z = 0$:

$$\begin{aligned}\tfrac{R'^1_\sigma(z)}{R^1_\sigma(z)} &\leq \left(\int_{z-\frac{1}{2}-e_l}^{z+\frac{1}{2}+e_r} \tfrac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/(2\sigma^2)}dx\right)\Big/\left(\int_{z-\frac{1}{2}}^{z+\frac{1}{2}} \tfrac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/(2\sigma^2)}dx\right)\\ &\leq 1+|e_r|+|e_l| \leq 1+\tfrac{1}{2}\cdot 2^{-p}+\tfrac{1}{2}\cdot 2^{-p} = 1+2^{-p},\end{aligned}$$

as we have seen in the computations for RD_1. Since this is less than the maximum, we can use the upper bound $\exp\left(2^{-p}(\tau\sigma)^2\exp\left(\frac{1}{\sigma^2}\right)\left(\exp\left(\frac{1}{2\sigma^2}\right)+1\right)\right)$ to determine the floating point precision p needed.

We have $\mathrm{RD}_\infty(R'^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma) \leq \mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma)^{mq_s}$ and want to find an expression for p from this. This results in the following equations:

$$\begin{aligned}\mathrm{RD}_\infty(R'^{mq_s}_\sigma \mid\mid R^{mq_s}_\sigma) &\leq \mathrm{RD}_\infty(R'^1_\sigma \mid\mid R^1_\sigma)^{mq_s}\\ &\leq \exp\left(2^{-p}(\tau\sigma)^2\exp\left(\tfrac{1}{\sigma^2}\right)\left(\exp\left(\tfrac{1}{2\sigma^2}\right)+1\right)\right)^{mq_s}.\end{aligned}$$

We set the floating point precision p such that

$$\exp\left(mq_s 2^{-p}(\tau\sigma)^2\exp\left(\frac{1}{\sigma^2}\right)\left(\exp\left(\frac{1}{2\sigma^2}\right)+1\right)\right) \leq \exp(1).$$

This yields a precision argument

$$p \geq \log\left(mq_s(\tau\sigma)^2\exp\left(\frac{1}{\sigma^2}\right)\left(\exp\left(\frac{1}{2\sigma^2}\right)+1\right)\right). \tag{12}$$

Recall that we assumed that $\tau\sigma \ll 2^{-p/2}$, i.e. $p > 2\log(\tau\sigma)$. We need to check if this is true for the result we got. We see that indeed we get

$$\begin{aligned}p &\geq \log\left(mq_s(\tau\sigma)^2\exp\left(\tfrac{1}{\sigma^2}\right)\left(\exp\left(\tfrac{1}{2\sigma^2}\right)+1\right)\right)\\ &= 2\log(\tau\sigma)+\log\left(mq_s\exp\left(\tfrac{1}{\sigma^2}\right)\left(\exp\left(\tfrac{1}{2\sigma^2}\right)+1\right)\right) > 2\log(\tau\sigma),\end{aligned}$$

since all the logarithms give a positive result.

Note that, as in the analysis of the discrete Gaussian in [1], Equation (12) does not explicitly depend on ϵ. However, the dependency on ϵ is hidden in the security parameter λ, which is still dependent on ϵ.

	Lower bound on the precision p
SD (Equation (6))	$p \geq \log\left(mq_s\tau\sigma\left(\sqrt{2\pi\sigma^2}+1\right)\right)+\lambda-\log\left(\sqrt{2\pi\sigma^2}\right)+1$
RD_1 (Equation (8))	$p \geq \log\left(\frac{\tau\sigma\sqrt{mq_s\left(1+\exp\left(\frac{9}{4\sigma^2}\right)\right)}}{\sqrt{\epsilon^2+mq_s\left(1+\exp\left(\frac{9}{4\sigma^2}\right)\right)}-\sqrt{mq_s\left(1+\exp\left(\frac{9}{4\sigma^2}\right)\right)}}\right)+2$
RD_∞ (Equation (12))	$p \geq \log\left(mq_s(\tau\sigma)^2\exp\left(\frac{1}{\sigma^2}\right)\left(\exp\left(\frac{\tau}{2\sigma}\right)+1\right)\right)$

Table B.1. Comparison of the precision p to handle adversaries with success probability $\geq \epsilon$ making $\leq q_s$ signing queries to BLISS signature generation with Box-Muller transformation.

	Example p for rounded Gaussians	Example p for discrete Gaussians
SD	$p \geq 215$	$p \geq 207$
RD_1	$p \geq 346$	$p \geq 168$
RD_∞	$p \geq 98$	$p \geq 79$

Table B.2. Comparison of the precision p needed for BLISS-I implemented with rounded Gaussians and implemented with discrete Gaussians.

Equation (12) eliminates the term ϵ from the floating point precision p, which was needed for the SD-based and the RD_1-based analyses. However, m, q_s and ϵ are dependent on λ, i.e. the resulting floating point precision p is not independent of ϵ, since it is not independent of λ.

We summarize the results in Table B.1. Before we can numerically compute this p, we need to know the value of m and against how many signing queries q_s we want to be protected.

Note that the precision plays different roles per sampler and implementation. In our sampling approach, each computation step has the potential to decrease the precision, but all considerations are worst-case considerations. The CDT sampler that we considered for comparison has a stored table of fixed precision. To compare the precision bounds as described in Table B.1 to the precision bounds found in [1] for BLISS-I we use the same values for the variables, that is, we use $\epsilon = 2^{-128}$, dimension $m = 1024$, $q_s = 2^{64}$ sign queries, $\sigma = 215$ and tail bound $\tau = \sqrt{(2 \cdot 128 \cdot \log(2))} = 13.32087377852$. The results can be found in Table B.2. Here we can see that rounded Gaussians need more precision than discrete Gaussians, but rounded Gaussians come with the advantage that they can easily be implemented in constant time and without table look ups, which makes it suitable to use rounded Gaussians in practice for BLISS. Furthermore, the estimates are less tight because of the approximation of integrals and errors by their worst case value.

Note that the values in Table B.2 tell us the resulting precision needed. If we want to know the implementations precision, i.e. the precision before the implementation makes any changes, we need to compute how much precision is lost by the implementation. For our implementation of BLISS-I we have computed the loss of precision in Section 5.2.