Sharper Bounds in Lattice-Based Cryptography using the Rényi Divergence

Thomas Prest*

Thales Communications & Security

Abstract. The Rényi divergence is a measure of divergence between distributions. It has recently found several applications in lattice-based cryptography. The contribution of this paper is twofold.
First, we give theoretic results which renders it more efficient and easier to use. This is done by providing two lemmas, which give tight bounds in very common situations – for distributions that are tailcut or have a bounded relative error. We then connect the Rényi divergence to the max-log distance. This allows the Rényi divergence to indirectly benefit from all the advantages of a distance.
Second, we apply our new results to five practical usecases. It allows us to claim 256 bits of security for a floating-point precision of 53 bits, in cases that until now either required more than 150 bits of precision or were limited to 100 bits of security: rejection sampling, trapdoor sampling (61 bits in this case) and a new sampler by Micciancio and Walter. We also propose a new and compact approach for table-based sampling, and squeeze the standard deviation of trapdoor samplers by a factor that provides a gain of 30 bits of security in practice.

Keywords: Rényi Divergence, Security Proofs, Lattice-Based Cryptography, Gaussian Sampling.

1 Introduction

An essential tool in cryptography is the use of divergence measures to prove the security of cryptographic schemes. As an introductory example, we consider the statistical distance Δ. It verifies a probability preservation property, which states that for any two distributions $\mathcal{P}, \mathcal{Q}$ and any measureable event E over the support of $\mathcal{P}$ and $\mathcal{Q}$, we have

$$\mathcal{Q}(E) \geq \mathcal{P}(E) - \Delta(\mathcal{P}, \mathcal{Q}). \tag{1}$$

In a cryptographic context, a useful abstraction is to modelize a cryptographic scheme as relying on some ideal distribution $\mathcal{Q}$ and the success of an attacker against this scheme as an event E. If $\Delta(\mathcal{P}, \mathcal{Q})$ is negligible, the equation 1 will allow to say that a scheme secure with $\mathcal{Q}$ will stay secure if one replaces $\mathcal{Q}$ by an "imperfect" distribution $\mathcal{P}$. Many other measures can be used to provide security arguments in cryptography (see e.g. [Cac97]).

* Contact: https://www.di.ens.fr/~prest/.

The Rényi divergence. In the subfield of lattice-based crytography, the Rényi divergence [R61] has been used for cryptographic proofs in several recent works. Noted R_a, it is somewhat trickier to use than the statistical distance. First, it is parameterized by a value $a \in [0, +\infty]$, and has different properties depending on a. It is not a distance, as it is asymmetric and does not verify the triangle inequality; the lack of these two properties can be problematic in security proofs. Interestingly, it also verifies a probability preservation property. For any event $E \subseteq \mathrm{Supp}(\mathcal{Q})$ and $a \in (1, +\infty)$, we have

$$\mathcal{Q}(E) \geq \mathcal{P}(E)^{a/(a-1)} / R_a(\mathcal{P} \| \mathcal{Q}). \tag{2}$$

The equation 2 is not additive like equation 1, but rather multiplicative. We will later see that in the context of search problems, it allows to give tighter bounds in practice.

1.1 Floating-Point in Lattice-Based Cryptography

Lattice-based cryptography has proven to be a serious candidate for post-quantum cryptography. It is efficient and allows to instantiate a wide range of cryptographic primitives. Some lattice-based schemes [DDLL13,ADPS16] have even already been deployed in large-scale projects.[1]

A notable characteristic of lattice-based cryptography is that it often makes extensive use of floating-point arithmetic, for several reasons.

Gaussians. The first vector for the use of floating-point arithmetic in lattice-based cryptography is the widespread need to sample from discrete Gaussian distributions. When done by standard approaches like precomputed tables, [Pei10] the required precision is rather high and renders the use of these tables cumbersome if not impractical.

On the other hand, bitwise approaches [DDLL13] have been developed to circumvent these floating-point issues, but they can be somewhat tricky to implement.

Rejection sampling. In the early lattice-based signature schemes GGH [GGH97] and NTRUSign [HHGP+03], there existed a correlation between the secret key and the distribution of the signatures. This subsequently led to several key-recovery attacks [GJSS01,GS02,NR06,Wan10,DN12b] which broke the signature schemes and their evolutions.

A provably secure countermeasure was introduced by Lyubashevsky [Lyu09]. The idea is to use rejection sampling as a final step, in order to "factor out" the correlation between the key and the distribution of the signatures.

[1] [Str14] and https://www.imperialviolet.org/2016/11/28/cecpq1.html.

This paradigm was instantiated in [Lyu12,GLP12,DDLL13,PDG14,POG15]. Now, in the existing implementations [DDLL13], this step is *not* done in floating-point. Because of precision concerns, another approach based on combining Bernoulli samples was chosen. We will see in section 4.3 that this approach also has several drawbacks.

Trapdoor sampling. In lattice-based cryptography, the tool that makes the most intensive use of floating-point arithmetic is arguably trapdoor sampling. Introduced by Gentry et al. [GPV08], it is a cornerstone of lattice-based cryptography, as it has numerous applications such as hash-and-sign and identity-based encryption in the random oracle model [GPV08], signatures in the standard model [CHKP10,Boy10], hierarchical IBE [CHKP10,ABB10a,ABB10b], attribute-based encryption [Boy13,BGG+14], and much more.

The existing algorithms [Kle00,GPV08,Pei10,MP12] heavily rely on floating-point arithmetic and they perform between $O(n \log n)$ and $O(n^2)$ floating-point operations. However, the best available estimations require 150 bits of precision for a security of 256 bits, which is completely impractical.

As we can see, floating-point arithmetic can be found everywhere in lattice-based cryptography. However, if often comes with high precision, which makes it impractical as it stands.

1.2 Our Contributions

Theory. We provide theoretic tools related to the use of the Rényi divergence in cryptographic proofs. They make it not only simpler to use, but also very efficient in some easily-identifiable situations.

1. We establish two lemmas that bound the Rényi divergence of related distributions in two very common situations in lattice-based cryptography. The first lemma concerns tailcut distributions, and for this reason we call it the tailcut lemma. The second one involves distributions which relative error is bounded, so we call it the relative error lemma. The second lemma is particularly powerful in the sense that it often allows to take very aggressive parameters.

2. We show that taking $a = 2\lambda$ allows to have tight and efficient Rényi divergence-based security arguments for cryptographic schemes based on search problems. We also derive simple and explicit conditions on distributions that allow to easily replace a distribution by another in this context.

3. A simple and versatile distance of divergence was recently introduced by Micciancio and Walter [MW17], the max-log distance. We establish a "reverse Pinsker" inequality between it and the Rényi divergence. An immediate consequence is that we may benefit from the best of both worlds: the versatility of the max-log distance, and the power of the Rényi divergence.

Practice. Our results are not purely theoretic. In section 4, we present five applications of them in lattice-based cryptography.

1. We start by the study of a sampler recently introduced by Micciancio and Walter [MW17]. We show that for this sampler, the security analysis provided by [MW17] can be improved and we can claim a full security of 256 bits instead of the 100 bits claimed in [MW17].

2. We revisit the table-based approach (see e.g. [Pei10]) for sampling distributions such as discrete Gaussians. By a Rényi divergence-based analysis combined to a little tweak on the precomputed table, we reduce the storage size by an order of magnitude, both in theory and in practice (where we gain a factor 9). Our improvement seems highly composable with other techniques related to precomputed tables.

3. We analyze the rejection sampling step of BLISS [DDLL13]. We show that it can be done simply and efficiently in floating-point, simultaneously eliminating the issues – code complexity, side-channel attacks, table storage, etc. – that plagued the only previously existing approach.

4. We then study trapdoor samplers [Kle00,GPV08,Pei10]. We improve the usual bounds on the standard deviation σ by obtaining a new bound which is both smaller and essentially independent of the security level λ. In practice, we gain about 30 bits of security compared to a statistical distance-based analysis.

5. The last contribution is also related to trapdoor samplers. We show that a precision of 64 bits allows 256 bits of security, whereas previous estimations [LP15,Pre15] required a precision of 150 bits.

A word on the security parameter and number of queries. In order to make our results as simple as possible and to derive explicit bounds, we consider in this paper that the security level λ and the number of queries q_s verify $\lambda \leq 256$ and $q_s \leq 2^{64}$. The first choice is arguably standard.

For the bound on q_s, we consider that making more than 2^{64} signature queries would be extremely costly and, unlike queries to e.g. a hash function, require the presence of the target to attack. In addition, it would be easily detectable by the target and so we believe it to be impractical.

Finally, a more pragmatic reason comes from NIST's current call for proposals for post-quantum cryptography,[2] which explicitely assumes that an attacker can make no more than 2^{64} signatures queries (resp. decryption queries).

However, if one decides to take $q_s > 2^{64}$, our results could be easily adapted, but their efficiency would be impacted.

[2] http://csrc.nist.gov/groups/ST/post-quantum-crypto/

1.3 Related Works

In the context of lattice-based cryptography, Stehlé, Steinfeld and their coauthors [LSS14,LPSS14,BLL+15] have used the Rényi divergergence to derive better parameters for cryptographic schemes. The Rényi divergence has also been used by [BGM+16] to improve security proofs, and in [TT15], which aims to improve the proofs from [BLL+15].

A few papers [PDG14,DLP14] used a third metric, the Kullback-Leibler divergence – actually the Rényi divergence of order 1 –, but the Rényi divergence has since then given better results [BLL+15, this work].

Precision issues have been tackled by [DN12a], which resorted to lazy Gaussian sampling but still didn't eliminate high-precision. A precision analysis of trapdoor samplers by Prest [Pre15] gave about 120 bits of precision for $\lambda = 192$ – which we extrapolate to 150 for $\lambda = 256$. A recent work by Saarinen [Saa15] has also claimed that using p-bit fixed point approximation achieves $2p$ bits of security, but this was proven to be incorrect by [MW17], which also introduced the max-log distance.

1.4 Roadmap

Section 2 introduces the notations and tools that we will use throughout the paper, including the Rényi divergence.

Section 3 is dedicated to our theoretic results. We first present the tailcut and relative error lemmas, as well as typical usecases for their applications. We give a framework for using them in cryptographic proofs, along with explicit bounds. Finally, we establish a connection between the Rényi divergence and the max-log distance.

Section 4 presents five applications of our theoretic results. We first give a tighter analysis of a sampler from [MW17], then we revisit the standard table-based approach for sampling Discrete distributions. We then show that rejection sampling in BLISS can be done simply in floating-point arithmetic. To conclude, we study trapdoor samplers and provide improved bounds on the standard deviation and precision with which they can be used.

Section 5 concludes this article and presents related open problems.

2 Preliminaries

2.1 Notations

Cryptographic parameters. When clear from context, we note λ the security level of a scheme and q_s the number of public queries that an attacker can make. In this article, we consider that $\lambda \leq 256$ and $q_s \leq 2^{64}$.

Probabilities For any distribution $\mathcal{D}$, we note $\mathrm{Supp}(\mathcal{D})$ its support. We may abbreviate the statistical distance and Kullback-Leibler divergence by SD and KLD. As a mnemonic device, we will often refer to $\mathcal{D}$ as some perfect distribution, and to $\mathcal{D}_\delta$ as a distribution close to $\mathcal{D}$ in a sense parameterized by δ.

Matrices and vectors. Matrices will usually be in bold uppercase (e.g. $\mathbf{B}$), vectors in bold lowercase (e.g. $\mathbf{v}$) and scalars in italic (e.g. s). Vectors are represented as rows. The p-norm of an vector $\mathbf{v}$ is noted $\|\mathbf{v}\|_p$, and by convention $\|\mathbf{v}\| = \|\mathbf{v}\|_2$. We note $\|\mathbf{B}\|_2 = \max_{\mathbf{x}\neq 0} \|\mathbf{xB}\|_2/\|\mathbf{x}\|_2$ the spectral norm of a matrix, it is also the maximum of its singular values and is sometimes noted $s_1(\mathbf{B})$. For $\mathbf{B} = (b_{ij})_{i,j}$, we define the max norm of $\mathbf{B}$ as $\|\mathbf{B}\|_{\max} = \max_{i,j} |b_{ij}|$.

Gram-Schmidt orthogonalization. An important tool in lattice-based cryptography is the Gram-Schmidt orthogonalization of a full-rank matrix $\mathbf{B}$, which is the unique factorization $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$ such that $\mathbf{L}$ is lower triangular with 1's on the diagonal, and $\tilde{\mathbf{B}}$ is orthogonal. Noting $\tilde{\mathbf{B}} = (\tilde{\mathbf{b}}_i)_i$, it allows to define the Gram-Schmidt norm, defined as $\|\mathbf{B}\|_{\mathrm{GS}} = \max_i \|\tilde{\mathbf{b}}_i\|$.

Lattices and Gaussians. A lattice will be noted Λ. For a matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, we note $\Lambda(\mathbf{B})$ the lattice generated by $\mathbf{B}$: $\Lambda(\mathbf{B}) = \mathbb{Z}^n \cdot \mathbf{B}$. We define the Gaussian function $\rho_{\sigma,\mathbf{c}}$ as $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp(-\|\mathbf{x}-\mathbf{c}\|^2/2\sigma^2)$, and the Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$ over a lattice as

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{z}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{z})}$$

The parameter $\mathbf{c}$ may be omitted when it is equal to zero.

Smoothing parameter. For $\epsilon > 0$, we define the smoothing parameter $\eta_\epsilon(\Lambda)$ of a lattice as the smallest value $\sigma > 0$ such that $\rho_{1/\sigma}(\Lambda^\star\backslash\mathbf{0}) \leq \epsilon$. We carefully note that in the existing literature, some definitions take the smoothing parameter to be our definition multiplied by a factor $\sqrt{2\pi}$. A useful bound on the smoothing parameter is given by [MR07]:

$$\eta_\epsilon(\mathbb{Z}^n) \leq \frac{1}{\pi}\sqrt{\frac{1}{2}\log\left(2n\left(1+\frac{1}{\epsilon}\right)\right)}. \tag{3}$$

2.2 The Rényi Divergence

We define the Rényi divergence in the same way as [BLL+15].

Definition 1 (Rényi divergence). *Let $\mathcal{P}, \mathcal{Q}$ be two distributions such that $Supp(\mathcal{P}) \subseteq Supp(\mathcal{Q})$. For $a \in (1, +\infty)$, we define the Rényi divergence of order a by*

$$R_a(\mathcal{P}\|\mathcal{Q}) = \left(\sum_{x\in Supp(\mathcal{P})} \frac{\mathcal{P}(x)^a}{\mathcal{Q}(x)^{a-1}}\right)^{\frac{1}{a-1}}.$$

In addition, we define the Rényi divergence of order $+\infty$ by

$$R_\infty(\mathcal{P}\|\mathcal{Q}) = \max_{x\in Supp(\mathcal{P})} \frac{\mathcal{P}(x)}{\mathcal{Q}(x)}.$$

Again, this definition is slightly different from some other existing definitions, which take the log of ours. However, it is more convenient for our purposes. Generic (resp. cryptographic) properties of the Rényi divergence can be found in [vEH14] (resp. [BLL+15]). We recall the most important ones.

Lemma 1 ([BLL+15, Lemma 2.9]). *For two distributions $\mathcal{P}, \mathcal{Q}$ and two families of distributions $(\mathcal{P}_i)_i, (\mathcal{Q}_i)_i$, the Rényi divergence verifies the following properties:*

- ***Data processing inequality.*** *For any function f, $R_a(\mathcal{P}^f \| \mathcal{Q}^f) \leq R_a(\mathcal{P} \| \mathcal{Q})$.*
- ***Multiplicativity.*** $R_a(\prod_i \mathcal{P}_i \| \prod_i \mathcal{Q}_i) = \prod_i R_a(\mathcal{P}_i \| \mathcal{Q}_i)$.
- ***Probability preservation.*** *For any event $E \subseteq Supp(\mathcal{Q})$ and $a \in (1, +\infty)$,*

$$\mathcal{Q}(E) \geq \mathcal{P}(E)^{a/(a-1)} / R_a(\mathcal{P} \| \mathcal{Q}),$$
$$\mathcal{Q}(E) \geq \mathcal{P}(E) / R_\infty(\mathcal{P} \| \mathcal{Q}).$$

However, we note that the Rényi divergence is not a distance. In section 3.4, we will circumvent this issue by linking the Rényi divergence to the max-log distance.

3 Main Results

In this section, we present our theoretic results: the tailcut lemma and relative error lemma for bounding the Rényi divergence between distributions, a generic framework for using these lemmas and a "reverse Pinsker" inequality that connects the Rényi divergence to the max-log distance.

3.1 The Tailcut Lemma

This first lemma may arguably be considered as folklore; it is already briefly mentioned in e.g. [BLL+15]. Here we explicit it, as applications of it arise naturally in lattice-based cryptography, especially whenever Gaussians distributions are used.

Lemma 2 (Tailcut). *Let $\mathcal{D}, \mathcal{D}_\delta$ be two distributions such that:*

- *$Supp(\mathcal{D}_\delta) \subseteq Supp(\mathcal{D})$*
- *$\exists \delta > 0$ such that $\frac{\mathcal{D}_\delta}{\mathcal{D}} \leq 1 + \delta$ over $Supp(\mathcal{D}_\delta)$*

Then for $a \in (1, +\infty]$:

$$R_a(\mathcal{D}_\delta \| \mathcal{D}) \leq 1 + \delta$$

Proof. We note $S = \mathrm{Supp}(\mathcal{D}_\delta)$. If $a \neq +\infty$:

$$R_a(\mathcal{D}_\delta \| \mathcal{D})^{a-1} = \sum_{x \in S} \frac{\mathcal{D}_\delta(x)^a}{\mathcal{D}(x)^{a-1}} \leq (1+\delta)^{a-1} \sum_{x \in S} \mathcal{D}_\delta(x) \leq (1+\delta)^{a-1},$$

which yields the result. If $a = +\infty$, the result is immediate. □

We may also refer to lemma 2 as the tailcut lemma.

Usecases. As its name implies, the tailcut lemma is adapted to situations where $\mathcal{D}_\delta$ is a "tailcut" of $\mathcal{D}$: we discard a set $T \subseteq \mathrm{Supp}(\mathcal{D})$ such that $\mathcal{D}(T) \leq \delta$. In order to still have a true measure of probability, the remaining probabilities are scaled by a factor $\frac{1}{1-\mathcal{D}(T)} \approx 1 + \mathcal{D}(T) \leq 1 + \delta$, and we note $\mathcal{D}_\delta$ the new distribution. Lemma 2 gives a relation of closeness between $\mathcal{D}$ and $\mathcal{D}_\delta$ in this case, which is illustrated by the figure 1.

3.2 The Relative Error Lemma

In our second lemma, the conditions are slightly stricter than for the tailcut lemma, but as a compensation the result is a much stronger closeness relation. It is somewhat similar to the [PDG14, Lemma 2] for the KLD, but allows tighter security arguments.

Lemma 3 (Relative error). *Let $\mathcal{D}, \mathcal{D}_\delta$ be two distributions such that:*

- *$Supp(\mathcal{D}_\delta) = Supp(\mathcal{D})$*
- *$\exists \delta > 0$ such that $1 - \delta \leq \frac{\mathcal{D}_\delta}{\mathcal{D}} \leq 1 + \delta$ over $Supp(\mathcal{D}_\delta)$*

Then, for $a \in (1, +\infty)$:

$$R_a(\mathcal{D}_\delta || \mathcal{D}) \leq \left(1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}}\right)^{\frac{1}{a-1}} \underset{\delta \to 0}{\sim} 1 + \frac{a\delta^2}{2}$$

Proof. Let $f_a : (x, y) \mapsto \frac{y^a}{(x+y)^{a-1}}$. We compute values of f_a and its derivatives around $(0, y)$:

$$\begin{aligned}
f_a(x,y) &= y && \text{for } x = 0 \\
\tfrac{\partial f_a}{\partial x}(x,y) &= 1 - a && \text{for } x = 0 \\
\tfrac{\partial^2 f_a}{\partial x^2}(x,y) &= a(a-1)y^a(x+y)^{-a-1} \\
&\leq \tfrac{a(a-1)}{(1-\delta)^{a+1}y} && \text{for } |x| \leq \delta \cdot y
\end{aligned}$$

We now use partial Taylor bounds. If $|x| \leq \delta \cdot y$, then:

$$f_a(x,y) \leq f_a(0,y) + \frac{\partial f_a}{\partial x}(0,y) \cdot x + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}} \cdot y$$

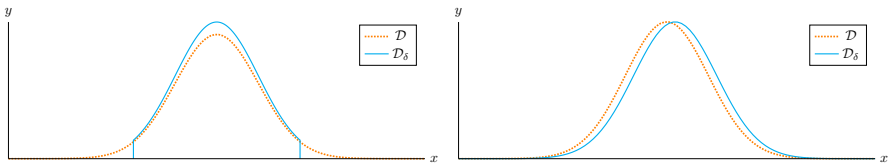
Let $S = \mathrm{Supp}(\mathcal{D}_\delta)$. Taking $y = \mathcal{D}_\delta(i)$, $x = \mathcal{D}(i) - \mathcal{D}_\delta(i)$, then summing i all over S and using the fact that $\sum_{i \in S} \mathcal{D}_\delta(i) = \sum_{i \in S} \mathcal{D}(i) = 1$ yields

$$\sum_{i \in S} \frac{\mathcal{D}_\delta(i)^a}{\mathcal{D}(i)^{a-1}} \leq 1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}}$$

From which we can conclude. □

We may also refer to lemma 3 as the relative error lemma.

Usecases. The relative error lemma can be used when the relative error between $\mathcal{D}_\delta$ and $\mathcal{D}$ is bounded. This may typically happen when the probabilities of $\mathcal{D}$ are stored in floating-point with a precision $\log_2 \delta$ – though we will see that it is not limited to this situation. Again, this is illustrated by figure 1.



Tailcut lemma Relative error lemma

Fig. 1. Typical usecases for the tailcut lemma and the relative error lemma

3.3 Security Arguments using the Rényi Divergence

We consider a cryptographic scheme making q_s queries to either a perfect distribution $\mathcal{D}$ or an imperfect distribution $\mathcal{D}_\delta$. Let E be an event breaking the scheme, and ε (resp. ε_δ) the probability that this event occurs under the use of $\mathcal{D}$ (resp. $\mathcal{D}_\delta$). We suppose that $\varepsilon_\delta \geq 2^{-\lambda}$. By the data processing and probability preservation inequalities:

$$\begin{aligned}\varepsilon &\geq \varepsilon_\delta^{a/(a-1)} / R_a(\mathcal{D}_\delta^{q_s} \| \mathcal{D}^{q_s}) \\ &\geq \varepsilon_\delta^{a/(a-1)} / R_a(\mathcal{D}_\delta \| \mathcal{D})^{q_s}\end{aligned}$$

We can choose any value in $(1, +\infty)$ for a, but small values for a impact the tightness of the reduction and large values impact its efficiency. Setting $a = 2\lambda$ seems to be a good compromise. Indeed, we then have $\varepsilon_\delta^{a/(a-1)} \geq \varepsilon_\delta / \sqrt{2}$, so we lose at most half a bit of security in the process.

Our goal is now to have $R_a(\mathcal{D}_\delta \| \mathcal{D})^{q_s} = \Omega(1)$, so that we have an almost tight security reduction. In this regard, having $R_a(\mathcal{D}_\delta \| \mathcal{D}) \leq 1 + \frac{1}{4q_s}$ is enough, since it yields $R_a(\mathcal{D}_\delta \| \mathcal{D})^{q_s} \leq e^{1/4} \leq \sqrt{2}$ by a classic inequality.[3]

This yields $\epsilon \geq 2^{-\lambda-1}$. By contraposition, a $(\lambda+1)$-bit secure scheme with $\mathcal{D}$ will be at least λ-bit secure when replacing $\mathcal{D}$ by $\mathcal{D}_\delta$ if the following condition is met:

$$R_a(\mathcal{D}_\delta \| \mathcal{D}) \leq 1 + \frac{1}{4q_s} \qquad \text{for } a = 2\lambda \tag{4}$$

We make two important remarks: first, this analysis is valid only for cryptographic schemes relying on search problems. This will be the case for all the applications that we consider in this paper, but for cryptographic schemes relying on decision problems, one may rather rely on SD-based, KLD-based analyses, or on a specific Rényi divergence-based analysis such as the one given in [BLL+15, Section 4].

[3] $(1 + x/n)^n \leq e^x$ for $x, n > 0$.

Second, the savings provided by our analysis heavily rely on the fact that the number of queries is limited. This was already observed in [BLL+15].

Practical Implications. We consider a cryptographic scheme of $\lambda \leq 256$ bits of security making $q_s \leq 2^{64}$ queries to a distribution $\mathcal{D}$. Replacing $\mathcal{D}$ by another distribution $\mathcal{D}_\delta$ will make the scheme lose at most one bit of security, provided that one of these conditions is verified:

$$\frac{\mathcal{D}_\delta}{\mathcal{D}} \leq 1 + \delta \text{ for } \delta = 2^{-66} \tag{5}$$

$$\mathrm{Supp}(\mathcal{D}_\delta) = \mathrm{Supp}(\mathcal{D}) \text{ and } 1 - \delta \leq \frac{\mathcal{D}_\delta}{\mathcal{D}} \leq 1 + \delta \text{ for } \delta = 2^{-37} \tag{6}$$

Where the condition 5 comes from using the tailcut lemma with equation 4, and the condition 6 from the relative error lemma with equation 4.

3.4 Relation to the max-log Distance

In [MW17], Micciancio and Walter introduced a new metric, the max-log distance. They argue that this metric is both easy to use and allows to have sharp bounds in cryptographic proofs.

In lemma 4, we show that the log of the Rényi divergence is bounded (up to a constant) by the square of the max-log distance. It can be seen as a "reverse" analogous of Pinsker inequality for the SD and KLD, so we call it the reverse Pinsker inequality.

Definition 2 (max-log distance [MW17]). *The max-log distance between two distributions $\mathcal{P}$ and $\mathcal{Q}$ over the same support S is*

$$\Delta_{\mathrm{ML}}(\mathcal{P}, \mathcal{Q}) = \max_{x \in S} |\log \mathcal{P}(x) - \log \mathcal{Q}(x)|$$

Lemma 4 (Reverse Pinsker inequality). *For two distributions $\mathcal{P}, \mathcal{Q}$ of common support, we have:*

$$R_a(\mathcal{P}||\mathcal{Q}) \leq \left(1 + \frac{a(a-1)(e^{\Delta_{\mathrm{ML}}(\mathcal{P},\mathcal{Q})} - 1)^2}{2(1 - \Delta_{\mathrm{ML}}(\mathcal{P}, \mathcal{Q}))^{a+1}}\right)^{\frac{1}{a-1}} \underset{\Delta_{\mathrm{ML}} \to 0}{\sim} 1 + \frac{a\Delta_{\mathrm{ML}}(\mathcal{P}, \mathcal{Q})^2}{2}$$

Proof. We note $\Delta_{\mathrm{ML}}(\mathcal{P}, \mathcal{Q}) = \delta$ for some $\delta \geq 0$. We have:

$$\begin{aligned} \Delta_{\mathrm{ML}}(\mathcal{P}, \mathcal{Q}) = \delta &\Rightarrow \forall x \in S, |\log \mathcal{P}(x) - \log \mathcal{Q}(x)| \leq \delta \\ &\Rightarrow \forall x \in S, e^{-\delta} \leq \tfrac{\mathcal{P}(x)}{\mathcal{Q}(x)} \leq e^{\delta} \\ &\Rightarrow R_a(\mathcal{P}||\mathcal{Q}) \leq \left(1 + \tfrac{a(a-1)(e^{\delta}-1)^2}{2(1-\delta)^{a+1}}\right)^{\frac{1}{a-1}} \end{aligned}$$

The first implication applies the definition of the max-log distance, the second one passes to the exponential, the third one applies the relative error lemma. □

There are two implications from lemma 4. First, we can add the max-log distance to our tools. Unlike the Rényi divergence, it is actually a distance, which is often useful when performing security analyses.

Second, the lemma 4 provides evidence that the Rényi divergence gives sharper bounds than the max-log distance, as the log of the former is essentially bounded by the square of the second.

In addition, we point out that the max-log distance is defined only for distributions with a common support. Per example, it cannot be applied to tailcut distributions. It is nevertheless a useful measure. One may per example use it if a true distance is needed, and then fall back to the Rényi divergence using lemma 4.

4 Applications

In this section we provide five applications of our results. In all the cases studied, we manage to claim 256 bits of security while lowering the precision requirements to be less than 53 bits (or 61 bits for the last application).

This bound of 53 bits is important. Floating-point with 53 bits of precision corresponds to the double precision type in the IEEE 754 standard, and is is very often available in software – see e.g. the type `double` in C. In many cases, it can also be simulated using fixed-point numbers of 64 bits of precision, which can be done easily and efficiently, in particular over 64-bit architectures.

4.1 Tighter Analysis of the Micciancio-Walter Sampler

The first application of our results is also arguably the simplest. A new Gaussiam sampler over $\mathbb{Z}$ was recently introduced by Micciancio and Walter [MW17]. They provide a security analysis using the max-log distance [MW17, Lemma 5.5].

Later, at the end of [MW17, Section 5.3], this lemma is used to argue that for a given set of parameters, if we note $\mathcal{Q}$ a perfect Gaussian distribution and $\mathcal{P}$ the output of the new sampler, we have $\Delta_{\mathrm{ML}}(\mathcal{P}\|\mathcal{Q}) \leq 2^{-52}$. This in turn allows them to claim about 100 bits of security.

A tighter analysis. We now prove that a Rényi divergence-based analysis gives tighter bounds than the max-log distance-based analysis from [MW17]. This analysis is done completely in black box, as we do not need to know anything about the sampler, except the fact that $\Delta_{\mathrm{ML}}(\mathcal{P}\|\mathcal{Q}) \leq 2^{-52}$. Applying the reverse Pinsker inequality (lemma 4) yields $R_a(\mathcal{P}\|\mathcal{Q}) \leq 1 + 2^{-96}$ for any $a \leq 512$.

Following the security argument of section 3.3 and in particular equation 4, this allows us to claim that the use of this sampler is secure for 256 bits of security and $q_s = 2^{64}$ queries. This remains the case even if we ask up to 2^{94} queries, which we believe is more than enough for any practical application.

4.2 Revisiting the Table Approach

We now study a more generic problem, namely sampling distributions over $\mathbb{Z}$. We consider situations where the use of precomputed tables is practical: this includes but is not limited to (pseudo-)Gaussians with parameters known in advance.

We revisit the table-based approach. First, we show that the standard approach based on the cumulative distribution function (see e.g. [Pei10]) suffers from precision issues for a large class of distributions: light-tailed distributions. Informally, these are distributions which tails have a negligible weight (like Gaussians). They also happen to be widespread in lattice-based cryptography.

We then introduce a new approach based on the conditional density function. We show that for light-tailed distributions, it behaves in a much nicer way. To conclude, we take a real-life example and show that in terms of space, the new approach allows to gain an order of magnitude compared to the standard approach.

Definition 3. *For a distribution $\mathcal{D}$ over $S \subseteq \mathbb{Z}$, we call cumulative distribution function of $\mathcal{D}$ and note $\mathrm{CDF}_{\mathcal{D}}$ the function defined over S by*

$$\mathrm{CDF}_{\mathcal{D}}(z) = \sum_{i \leq z} \mathcal{D}(z)$$

Classical CDF sampling. To sample from $\mathcal{D}$, a standard approach is to store a precomputed table of $\mathrm{CDF}_{\mathcal{D}}$, draw a uniform deviate $u \leftarrow [0, 1]$ and output $z = \min\{i \in S | \mathrm{CDF}_{\mathcal{D}}(i) \geq u\}$. In practice, we will not store the complete CDF table. If $\mathcal{D} = D_{\mathbb{Z},c,\sigma}$ is a discrete Gaussian, then we store the values for $z \in (c - k_0\sigma, c + k_0\sigma) \cap \mathbb{Z}$ with a given precision p_0. We now estimate the requirements in the context of λ bits of security and $m \cdot q_s$ queries.[4]

SD-based analysis. Using [GPV08, Lemma 4.2], we have $k_0 = \sqrt{2(\lambda + \log_2 m)}$. Each $\mathcal{D}(z) = \mathrm{CDF}_{\mathcal{D}}(z) - \mathrm{CDF}_{\mathcal{D}}(z-1)$ should be known with absolute precision $\lambda + \log_2 m$, so we may take $p_0 = \lambda + \log_2 m$.

Rényi divergence-based analysis. From the tailcut lemma (see also equation 5), it is sufficient to take $k_0 = \sqrt{2(66 + \log_2 m)}$. From the relative error lemma (see also equation 6), each $\mathcal{D}(z)$ should be known with relative precision $37 + \log_2 m$.

For $\lambda = 256$, we divide the number of precomputed elements by about 1.87. A naive interpretation of the analyses above may also lead us to divide the precision p_0 by $(\lambda + \log_2 m)/(37 + \log_2 m) \approx 6.9$. However, the next paragraph will expose why we cannot simply do that.

[4] The call to a sampler over $\mathbb{Z}$ is often done several times per query. In the context of signatures, we typically have $m =$ the lattice dimension. Here we take $m = 2^{10}$.

Precision issues in the case of light-tailed distributions. In the previous paragraph, there is a slight but important difference between the SD and Rényi divergence analyses. The precision is given absolutely in the first case, and relatively in the second case. It is actually this relativity that allows us to use the relative error lemma in the second case, but it comes at a price: it is not efficient anymore to use the CDF table.

We present here an example explaining why this is the case: let $\mathcal{D}_2$ be the distribution defined over $\mathbb{N}^*$ by $\mathcal{D}_2(k) = 2^{-k}$. One can show that $\mathrm{CDF}_{\mathcal{D}_2}(k) = 1 - 2^{-k}$, so from a machine perspective, $\mathrm{CDF}_{\mathcal{D}_2}(k)$ will be rounded to 1 as soon as $k > p_0$. As a consequence, the probability output of the CDF table-based algorithm will be 0 for any $k > p_0 + 1$ and we will not be able to use the relative error lemma at all.

This problem is common to light-tailed distributions, including Gaussian-like distributions. As the CDF converges very fast to 1, we have to store it in high precision in order for it to be meaningful. This is not satisfactory from a practical viewpoint.

Conditional density sampling. A simple way around the aforementioned problem is to use the *conditional density function* instead of the CDF. First, we give its definition.

Definition 4. *For a distribution $\mathcal{D}$ over $\mathbb{N}$, we call conditional density function of $\mathcal{D}$ and note $\mathrm{CoDF}_{\mathcal{D}}$ the function defined by $\mathrm{CoDF}(z) = \mathcal{D}(z)/(\sum_{i \geq z} \mathcal{D}(i))$.*

In other words, $\mathrm{CoDF}(z)$ is the probability that a random variable X of distribution $\mathcal{D}$ takes the value z, *conditioned* to the fact that X is bigger or equal to z.[5] A way to use the CoDF to sample from $\mathcal{D}$ is given by algorithm 1, a variation of the CDF sampler.

Algorithm 1 CoDF sampler

Require: A precomputed table of $\mathrm{CoDF}_{\mathcal{D}}$
Ensure: $z \leftarrow \mathcal{D}$
1: $z \leftarrow 0$
2: $u \leftarrow [0, 1]$ uniformly
3: **while** $u \geq \mathrm{CoDF}_{\mathcal{D}}(z)$ **do**
4: $\quad z \leftarrow z + 1$
5: $\quad u \leftarrow [0, 1]$ uniformly
6: **Return** z

It is easy to show that the expected running time of algorithm 1 is the mean of $\mathcal{D}$. It outputs z with probability $\prod_{i<z} [1 - \mathrm{CoDF}_{\mathcal{D}}(i)] \cdot \mathrm{CoDF}_{\mathcal{D}}(z)$, which by

[5] We note that the support is now $S \subseteq \mathbb{N}$ instead of $S \subseteq \mathbb{Z}$, but switching between the two cases is algorithmically easy.

a telescopic product is equal to

$$\frac{\sum_{i>0}\mathcal{D}(i)}{\sum_{i\geq 0}\mathcal{D}(i)} \times \frac{\sum_{i>1}\mathcal{D}(i)}{\sum_{i\geq 1}\mathcal{D}(i)} \times \cdots \times \frac{\sum_{i>z-1}\mathcal{D}(i)}{\sum_{i\geq z-1}\mathcal{D}(i)} \times \frac{\mathcal{D}(z)}{\sum_{i\geq z}\mathcal{D}(i)} = \mathcal{D}(z) \quad (7)$$

and therefore, algorithm 1 is correct. However, in practice algorithm 1 will be used with precomputed values which are only correct up to a given precision. Lemma 5 provides an analysis of the algorithm is this case.

Lemma 5. *For a distribution $\mathcal{D}$ of support $S \subseteq \mathbb{N}$, let $f = \mathrm{CoDF}_{\mathcal{D}}$ be the* CoDF *of $\mathcal{D}$, and f_δ be an approximation of f such that, over S:*

$$\begin{aligned} 1-\delta \leq \ \tfrac{f_\delta}{f} \ &\leq 1+\delta \\ 1-\delta \leq \tfrac{1-f_\delta}{1-f} &\leq 1+\delta \end{aligned} \quad (8)$$

Let $\mathcal{D}_\delta$ be the output distribution of the algorithm 1 using a precomputed table of f_δ instead of f. Then, for any $z \in S$:

$$1-\delta z \underset{0\leftarrow\delta}{\sim} (1-\delta)^z \leq \frac{\mathcal{D}_\delta(z)}{\mathcal{D}(z)} \leq (1+\delta)^z \underset{\delta\to 0}{\sim} 1+\delta z$$

Proof. We have

$$\begin{aligned} & & \mathcal{D}_\delta(z) &= \prod_{i<z}[1-f_\delta(i)]\cdot f_\delta(z) \\ &\Rightarrow (1-\delta)^z \prod_{i<z}[1-f(i)]\cdot f(z) \leq \mathcal{D}_\delta(z) &\leq (1+\delta)^z \prod_{i<z}[1-f(i)]\cdot f(z) \\ &\Rightarrow (1-\delta)^z \cdot \mathcal{D}(z) &\leq \mathcal{D}_\delta(z) \leq (1+\delta)^z \cdot \mathcal{D}(z) \end{aligned}$$

The first implication comes from equation 8, the second one from equation 7. □

Provided that the CoDF is stored with enough precision, lemma 5 gives us an inequality that allows to use the relative error lemma. Now, the interesting part is that for light-tailed distributions, the CoDF does not converge to 1 as fast as the CDF, which is important if we want the lower part of equation 8 to be true. Per example, if $\mathcal{D} = D_{\mathbb{Z},1}$, we have $\mathrm{CDF}_{\mathcal{D}}(z) - \mathrm{CDF}_{\mathcal{D}}(z-1) = O(e^{-z^2/2})$, whereas $1-\mathrm{CoDF}_{\mathcal{D}}(z) = O(e^{-z})$. This allows to store $\mathrm{CoDF}_{\mathcal{D}}$ in small precision and still remain able to use lemma 5.

Of course, one may argue that z can be arbitrarily big. However, in practice we will not sample from a distribution $\mathcal{D}$ of infinite support directly but rather from a tailcut distribution of $\mathcal{D}$, in the bounds provided by the tailcut lemma, so z will not take too large values and we will be able to store $\mathrm{CoDF}_{\mathcal{D}}$ efficiently.

Solving the precision issues. Going back to the example of the distribution $\mathcal{D}_2$, the table 4.2 shows how $\mathrm{CDF}_{\mathcal{D}_2}(k)$ and $\mathrm{CoDF}_{\mathcal{D}_2}(k)$ are stored in machine precision, and how it impacts the associated sampler.

For the CDF-based sampler, due to precision issues, it samples from a distribution $\mathcal{D}_2'$ which has a probability 0 for elements in the tail of $\mathcal{D}_2$. In contrast,

the CoDF-based sampler approximates $\mathcal{D}_2$ correctly even for elements in the tail of $\mathcal{D}_2$.

k	1	2	3	...	54	55	...
$\mathrm{CDF}_{\mathcal{D}_2}(k)$	1/2	3/4	7/8	...	1	1	...
$\mathcal{D}'_2(k)$	1/2	1/4	1/8	...	0	0	...
$\mathrm{CoDF}_{\mathcal{D}_2}(k)$	1/2	1/2	1/2	...	1/2	1/2	...
$\mathcal{D}_2(k)$	1/2	1/4	1/8	...	2^{-54}	2^{-55}	...

Table 1. Precomputed values of CDF and CoDF of $\mathcal{D}_2$ as stored in 53 bits precision. The stored value of $\mathrm{CDF}_{\mathcal{D}_2}(k)$ quickly becomes 1, leading to the associated algorithm sampling from some incorrect distribution $\mathcal{D}'_2$ instead of $\mathcal{D}_2$.

Application: sampling over $D^+_{\mathbb{Z},\boldsymbol{\sigma_2}}$ in BLISS. An important step of the signature scheme BLISS consists of sampling $z \leftarrow D^+_{\mathbb{Z},\sigma_2}$, where $\sigma_2 \approx 0.85$.

In BLISS, this is done in a bitwise rejection sampling fashion [DDLL13, algorithm 10], which is very efficient in hardware but not so much in software. In addition, the structure of the algorithm 10 from [DDLL13] exposes it to side-channel attacks in the lines of [EFGT17] (see also section 4.3).

Instead, one can sample efficiently from $D^+_{\mathbb{Z},\sigma_2}$ using a precomputed table T:

- With a CDF+SD approach, T must have 20 elements of 266 bits each, which amounts to about 5 300 bits.
- With a CoDF+Rényi divergence approach and using lemma 5, T must have 11 elements of 53 bits each, which amounts to about 600 bits.

Here, the CoDF+Rényi divergence approach makes us gain an order of magnitude in storage requirements. In addition, algorithm 1 is efficient and can easily be made constant-time and protected against side-channel attacks. Another notable advantage is that it is particularly fit to a fixed point implementation, which might make it easy to implement in hardware. Finally, it is generic in the sense that it can be applied to a large class of distributions over $\mathbb{N}$ (or $\mathbb{Z}$).

4.3 Simpler and More Secure Rejection Sampling in BLISS

We recall that the context and motivation of doing rejection sampling in lattice-based cryptography is exposed in section 1.1. We now focus our attention on the signature scheme BLISS [DDLL13]. In BLISS, the final step of the signature consists of this step:

$$\text{Accept with probability } p = 1/\left(M \exp\left(-\frac{\|\mathbf{Sc}\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{z}, \mathbf{Sc}\rangle}{\sigma^2}\right)\right) \tag{9}$$

where $\mathbf{S}$ is the secret key, σ, M are public parameters and $\mathbf{c}, \mathbf{z}$ are part of the signature. In the original scheme, as well as in all the subsequent implementations

that we are aware of [LD13,Pop14,Str14], this step is implemented by the means of combining several Bernoulli distributions dependent of the bits of $\|\mathbf{Sc}\|^2$ and $\langle \mathbf{z}, \mathbf{Sc} \rangle$.

There are two drawbacks from this approach. First, the algorithm described in [DDLL13] for performing this step is rather sophisticated, and as a result it takes a significant portion of the coding effort in [LD13,Pop14,Str14].

The second drawback is that this algorithm is actually vulnerable to side-channel attacks: Espitau et al. [EFGT17] have shown that a side-channel analysis of the signature traces can recover both $\|\mathbf{Sc}\|^2$ and $\langle \mathbf{z}, \mathbf{Sc} \rangle$, and from it the secret key. Interestingly, it might be possible to extend this attack to a timing attack, in which case the implementation of Strongswan [Str14], deployed on Windows, Linux, Mac OS, Android and iOS platforms, could also suffer from it.

Simple Rejection Sampling. We now make the following remark: the step 9 doesn't need to be made exactly. We can simply compute a value p_δ such that

$$1 - \delta \leq \frac{p_\delta}{p} \leq 1 + \delta,$$

sample a uniform deviate $u \leftarrow [0, 1]$ and accept if and only if $p_\delta \geq u$. By the equation 6, it is sufficient that p is computed with a relative error 2^{-37}. This can be done easily:

1. **In software,** one may simply resort to a standard implementation of the exp function, such as the one provided `math.h` for the C language. As long as the relative precision provided is more than 37 bits of precision, we can use equation 6. We note that many implementations of the exp function (like the one in `math.h`) provide 53 bits of precision, which is more than enough for our purposes.
2. **In hardware,** an implementation of the exp function may not always be available. There are many ways around this issue, we present two of them:
   - One may use Padé approximants as an efficient way to compute exp. Padé approximants are generalizations of Taylor series: they approximate a function f by a polynomial fraction $\frac{P_n}{Q_m}$ instead of a polynomial P_n. They usually converge extremely fast, and in the case of the exp function, the relative error between $\exp(z)$ and its Padé approximant is less than 2^{-37} for an approximation of order 4 and $|z| < 1/2$.[6] A more detailed analysis is provided in appendix, section A.1.
   - Another solution is to precompute the values $\exp(\frac{2^i}{2\sigma^2})$ for a small number of values $i \in \mathbb{N}$. This then allows to compute $\exp(\frac{z}{2\sigma^2})$ for any $z = \sum_i z_i 2^i$, since $\exp(\frac{z}{2\sigma^2}) = \prod_{z_i=1} \exp(\frac{2^i}{2\sigma^2})$.[7] For the parameters given

[6] It is easy reduce any input z to the case $|z| < 1/2$ by taking $z' \leftarrow z \bmod (\ln 2)$ and observing that $e^{\ln 2} = 2$. The precision loss is negligible.

[7] For negative values, exp may be computed by inversion, or if it is not available, by also precomputing $\exp(-\frac{2^i}{2\sigma^2})$.

by [DDLL13], $\|\mathbf{Sc}\|^2$ and $\langle \mathbf{z}, \mathbf{Sc} \rangle$ are integers and are less than 37 bits, which means that we would need to store at most 37 precomputed values.

For the two proposed solutions, a very pessimistic analysis estimates that we perform less than 80 elementary floating-point operations to compute p. While it might seem a lot for 3 exponentials, it is negligible compared to the total cost of a signature, which is around $O(n \log n)$ for $n = 512$ in the BLISS scheme. In addition, all the techniques we propose are easy to protect against side-channel attacks.

We note that our software solution and our hardware solution based on Padé approximants do not require to store any precomputed table.

In BLISS, explicitely computing the rejection bound as we did was discarded because of precision concerns. We note that all the security analysis in BLISS was performed using the SD, with only subsequent work [PDG14,BLL+15] using more adequate measures of divergence. Using the SD in our case would have required us compute transcendental functions with a precision 2^λ, which is impractical. The relative error lemma is the key which allows to argue that a floating-point approach is secure.

4.4 Squeezing the Standard Deviation of Trapdoor Samplers

Context. The two last sections are related to the most generic and powerful type of Gaussian sampling: trapdoor sampling. Algorithms for performing trapdoor sampling [Kle00,GPV08,Pei10,MP12] are essentially randomized variants of Babai's round-off and nearest plane algorithms [Bab85,Bab86]. For suitable parameters, they are statistically indistinguishable from a perfect Gaussian $D_{\Lambda,\sigma,\mathbf{c}}$.

For a cryptographic use, we want σ to be as small as possible in order to have the highest security guarantees. However, σ cannot be too small: if it is, then the trapdoor samplers will not behave anymore like perfect Gaussian oracles.[8] At the extreme case $\sigma = 0$, the samplers become deterministic and leak the shape of the basis used for sampling, exposing the associated schemes to key-recovery attacks described earlier. To avoid that, samplers usually come with lower bounds on σ for using it securely (see e.g. theorem 1 for Klein's sampler [Kle00,GPV08]).

Roadmap. Before continuing, we establish the roadmap for this section and the next one. In this section, we show that, if σ is large enough, a Gaussian sampler with infinite precision is as secure as an ideal Gaussian. In the next one, we show that a Gaussian sampler with finite precision is as secure as one with infinite precision. Of course, such analyses are already known. Our contribution here is to use the Rényi divergence to have more aggressive parameters for σ and the precision of the sampler.

[8] If they did behave like perfect Gaussians when $\sigma \to 0$, then they would effectively solve the closest vector problem, which is a NP-hard problem.

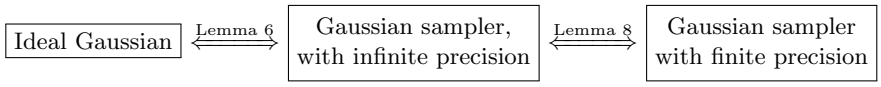


Fig. 2. Roadmap for asserting the security of a practical Gaussian sampler

Klein's sampler. We cannot analyse all the existing samplers in this article, so we now focus our attention on Klein's sampler [Kle00,GPV08]. It is described in algorithm 2.

Algorithm 2 $\text{KLEIN}_{\mathbf{L},\sigma}(\mathbf{t})$

Require: $\sigma \geq \eta_\epsilon(\mathbb{Z}^n) \cdot \|\mathbf{B}\|_{\text{GS}}$, the Gram-Schmidt orthogonalization $\mathbf{B} = \mathbf{L} \cdot \tilde{\mathbf{B}}$ and the values $\sigma_j = \sigma / \|\tilde{\mathbf{b}}_j\|$ for $j \in \{1, \ldots, n\}$
Ensure: A vector $\mathbf{z}$ such that $\mathbf{zB} \leftarrow D_{\Lambda(\mathbf{B}),\sigma,\mathbf{tB}}$
1: **for** $j = n, \ldots, 1$ **do**
2: $\quad c_j \leftarrow t_j + \sum_{i>j}(t_j - z_j)L_{ij}$
3: $\quad z_j \leftarrow D_{\mathbb{Z},\sigma_j,c_j}$
4: **return z**

An associated lower bound on σ for using algorithm 2 is given in theorem 1.

Theorem 1 ([DN12a, Th. 1], concrete version of [GPV08, Th. 4.1]). *Let $\epsilon = 2^{-\lambda}$. If $\sigma \geq \eta_\epsilon(\mathbb{Z}^n) \cdot \|\mathbf{B}\|_{\text{GS}}$, then the SD between $\text{KLEIN}_{\mathbf{L},\sigma}(\mathbf{t}) \cdot \mathbf{B}$ and the perfect discrete Gaussian $D_{\Lambda(\mathbf{B}),\sigma,\mathbf{tB}}$ is upper bounded by $2^{-\lambda}$.*

Combined to a standard SD-based argument, theorem 1 establishes that σ must be proportional to $\sqrt{\lambda}$ in order to claim λ bits of security when using algorithm 2. A better bound was established in [DLP14] but it remains proportional to $\sqrt{\lambda}$. In lemma 6, we establish a bound that is both (almost) independent of λ and smaller.

Lemma 6 (Rényi divergence of Klein's sampler). *For any $\epsilon \in (0, 1/4)$, if $\sigma \geqslant \eta_\epsilon(\mathbb{Z}^n) \cdot \|\mathbf{B}\|_{\text{GS}}$ then the Rényi divergence between $\mathcal{D} = D_{\Lambda(\mathbf{B}),\sigma,\mathbf{tB}}$ and the output distribution $\mathcal{D}_\epsilon$ of $\text{KLEIN}_{\mathbf{L},\sigma}(\mathbf{t}) \cdot \mathbf{B}$ verifies*

$$R_a(\mathcal{D}_\epsilon \| \mathcal{D}) \leq \left(1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}}\right)^{\frac{1}{a-1}} \underset{\delta \to 0}{\sim} 1 + \frac{a\delta^2}{2},$$

where $\delta = \left(\frac{1+\epsilon/n}{1-\epsilon/n}\right)^n - 1 \approx 2\epsilon$.

Proof. We note $\mathbf{v} = \mathbf{zB}$ and $\mathbf{c} = \mathbf{tB}$. As detailed in [GPV08], the probability that $\text{KLEIN}_{\mathbf{L},\sigma}(\mathbf{t})$ outputs a given $\mathbf{z}$ is proportional to

$$\prod_{i=1}^{n} \frac{1}{\rho_{\sigma_j,c_j}(\mathbb{Z})} \cdot \rho_{\sigma,\mathbf{c}}(\mathbf{v})$$

for $\sigma_j = \sigma/\|c_j\|$ and some $c_j \in \mathbb{R}$ that depends on $\mathbf{t}$ and $\mathbf{B}$. By assumption, $\sigma_j \geq \eta_\epsilon(\mathbb{Z}^n) \geq \eta_{\epsilon/n}(\mathbb{Z})$, therefore $\rho_{\sigma_j,c_j}(\mathbb{Z}) \in [\frac{1-\epsilon/n}{1+\epsilon/n}, 1] \cdot \rho_{\sigma_j}(\mathbb{Z})$ by [MR04, Lemma 4.4]. Since $\mathcal{D}(\mathbf{v})$ is proportional to $\rho_{\sigma,\mathbf{c}}(\mathbf{v})$ and $\mathcal{D}, \mathcal{D}_\epsilon$ both sum up to one, we have

$$\left(\frac{1-\epsilon/n}{1+\epsilon/n}\right)^n \leq \frac{\mathcal{D}_\epsilon}{\mathcal{D}} \leq \left(\frac{1+\epsilon/n}{1-\epsilon/n}\right)^n,$$

from which we may conclude by using the relative error lemma. □

Plugging this result with equation 6, we may use Klein's sampler with $\epsilon \leq 2^{-37}$, instead of $\epsilon \leq 2^{-\lambda}$ with the SD and $\epsilon \leq 2^{-\lambda/2}$ with the KLD [DLP14]. Compared to a SD-based analysis, this allows to squeeze σ by a factor $\sqrt{\lambda/37}$ that can be as large as ≈ 2.63 for $\lambda = 256$.

While it might seem a small gain, the security of trapdoor samplers is very sensitive to standard deviations variations. We estimate that this factor 2.63 allows to gain up to 30 bits of security (this claim is supported by e.g. [Pre15, Table 6.1]). A similar analysis for Peikert's sampler [Pei10] yields a similar gain.

4.5 Trapdoor Sampling in Standard Precision

For our last application of the Rényi divergence, we conclude our analysis of Klein's sampler (algorithm 2), by performing its precision analysis. This section shows that it can be used safely in small precision.

First, we give a lemma that bounds the ratio of two Gaussian sums in $\mathbb{Z}$ with slightly different centers and standard deviations.

Lemma 7 (Ratio of Gaussian Sums in $\mathbb{Z}$). *Let two arbitrary centers $t, \bar{t} \in \mathbb{R}$ and standard deviations $\sigma, \bar{\sigma} > 0$. Let the Gaussian functions $\rho(z) = \rho_{\sigma,t}(z)$, $\bar{\rho}(z) = \rho_{\bar{\sigma},\bar{t}}(z)$ and the distributions $\mathcal{D}(z) = \rho(z)/\rho(\mathbb{Z})$, $\bar{\mathcal{D}}(z) = \bar{\rho}(z)/\bar{\rho}(\mathbb{Z})$. Let $u(z) = \frac{(z-\bar{t})^2}{2\bar{\sigma}^2} - \frac{(z-t)^2}{2\sigma^2}$. Then*

$$e^{-\mathbb{E}_{z\leftarrow\mathcal{D}}[u]} \leq \frac{\bar{\rho}(\mathbb{Z})}{\rho(\mathbb{Z})} \leq e^{-\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}}[u]}$$

Proof. We first prove the left inequality. We have

$$\begin{aligned}
&\bar{\rho}(z) = e^{-u(z)}\rho(z) \\
\Rightarrow\ &\tfrac{\bar{\rho}(z)}{\rho(\mathbb{Z})} = e^{-u(z)}\mathcal{D}(z) \\
\Rightarrow\ &\tfrac{\bar{\rho}(\mathbb{Z})}{\rho(\mathbb{Z})} = \mathbb{E}_{z\leftarrow\mathcal{D}}[e^{-u(z)}] \\
\Rightarrow\ &\tfrac{\bar{\rho}(\mathbb{Z})}{\rho(\mathbb{Z})} \geq e^{-\mathbb{E}_{z\leftarrow\mathcal{D}}[u(z)]}
\end{aligned}$$

Where the last inequality comes from Jensen's inequality: since e is convex, $\mathbb{E}[e^{-u}] \geq e^{\mathbb{E}[-u]}$. Following the same reasoning, one gets

$$\left(\bar{\mathcal{D}}(z)e^{u(z)} = \frac{\rho(z)}{\bar{\rho}(\mathbb{Z})}\right) \Rightarrow \left(\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}}[e^u] = \frac{\rho(\mathbb{Z})}{\bar{\rho}(\mathbb{Z})}\right) \Rightarrow \left(\frac{\bar{\rho}(\mathbb{Z})}{\rho(\mathbb{Z})} \leq e^{-\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}}[u]}\right)$$

□

This lemma is useful in the sense that it provides a relative error bound, which will be used in the next lemma in order use the relative error lemma. We now give a bound on the required precision for using safely Klein's sampler.

Lemma 8. *Let $\mathcal{D}$ (resp. $\bar{\mathcal{D}}$) be the output distribution of algorithm 2 over the input $\mathbf{t}$ (resp. $\bar{\mathbf{t}}$), using precomputed values $(\mathbf{L}, (\sigma_j)_j)$ (resp. $(\bar{\mathbf{L}}, (\bar{\sigma}_j)_j)$). Let $\delta, \epsilon \in (0, .01)$. We note:*

- $T = n\|\mathbf{L}\|_{\max}(1.1 + \sigma\sqrt{2\pi} \cdot \|\mathbf{B}^{-1}\|_2)$
- $C = 1.3n\delta(\frac{T\sqrt{2\pi}}{\eta_\epsilon(\mathbb{Z}^n)} + 2\pi + 1)$

If we have the following (error) bounds on the input of algorithm 2:

- $\mathbf{t} \in [-.5, .5]^n$
- $\|\bar{\mathbf{t}} - \mathbf{t}\|_\infty \leq \delta$
- $|\bar{\sigma_j} - \sigma_j| \leq \delta\sigma_j$ *for all* j
- $\|\bar{\mathbf{L}} - \mathbf{L}\|_{\max} \leq \delta\|\mathbf{L}\|_{\max}$

Then we have this inequality:

$$e^{-C} \leq \frac{\bar{\mathcal{D}}}{\mathcal{D}} \leq e^C.$$

The lemma 8 covers – but is not limited to – the case where $\mathbf{L}$ and the $(\sigma_j)_j$'s are known up to a relative error, and $\mathbf{t}$ up to an absolute error. For any $\mathbf{z} \in \mathbb{Z}^n$, $D_{\mathbb{Z}^n,\sigma,\mathbf{z}+\mathbf{t}} = \mathbf{z} + D_{\mathbb{Z}^n,\sigma,\mathbf{t}}$, so it is perfectly reasonable to suppose $\mathbf{t} \in [-.5, .5]^n$.

Proof. This proof is rather long, so we explain its outline first. In ①, we establish a bound $A \leq \frac{\mathcal{D}(\mathbf{z})}{\bar{\mathcal{D}}(\mathbf{z})} \leq B$, for some expressions A, B. In ②, we establish $|A| \leq C$ and ③, we establish $|B| \leq C$. We conclude in ④.

① Let $\mathbf{z} = \sum_j \hat{z}_j \in \mathbb{Z}^n$ be a possible output of both samplers. We note $\mathbf{v} = \mathbf{zB}$ and $\mathbf{c} = \mathbf{tB}$. There exist a unique n-tuple $(c_j)_j$ (resp. $(\bar{c_j})_j$) such that at each step j, $\mathcal{E}$ (resp. $\bar{\mathcal{E}}$) samples a discrete Gaussian in $\mathbb{Z}$ around c_j (resp. $\bar{c_j}$).

The probability that $\mathbf{z}$ is output by $\mathcal{E}$ is $\mathcal{D}(\mathbf{z}) = \prod_j \mathcal{D}_j(\hat{z}_j) = \prod_j \frac{\rho_j(\hat{z}_j)}{\rho_j(\mathbb{Z})}$, where $\rho_j = \rho_{\mathbb{Z},\sigma_j,c_j}$ is uniquely defined by $\mathbf{z}$. Similarly, $\bar{\mathcal{D}}(\mathbf{z}) = \prod_j \frac{\bar{\rho}_j(\hat{z}_j)}{\bar{\rho}_j(\mathbb{Z})}$, where $\bar{\rho}_j = \rho_{\mathbb{Z},\bar{\sigma}_j,\bar{c_j}}$. We have

$$\frac{\mathcal{D}(\mathbf{z})}{\bar{\mathcal{D}}(\mathbf{z})} = \prod_j \frac{\rho_j(\hat{z}_j)}{\rho_j(\mathbb{Z})} \frac{\bar{\rho}_j(\mathbb{Z})}{\bar{\rho}_j(\hat{z}_j)} = \prod_j \frac{\rho_j(\hat{z}_j)}{\bar{\rho}_j(\hat{z}_j)} \frac{\bar{\rho}_j(\mathbb{Z})}{\rho_j(\mathbb{Z})}$$

For each j, let $u_j(z) = \frac{(z-\bar{c_j})^2}{2\bar{\sigma}_j^2} - \frac{(z-c_j)^2}{2\sigma_j^2}$. Lemma 7 yields:

$$e^{-\mathbb{E}_{z\leftarrow\mathcal{D}_j}[u_j]} \leq \frac{\bar{\rho}_j(\mathbb{Z})}{\rho_j(\mathbb{Z})} \leq e^{-\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}_j}[u_j]}$$

So that we have:

$$\sum_j \left[u_j(\hat{z}_j) - \mathbb{E}_{z\leftarrow\mathcal{D}_j}[u_j]\right] \leq \log\left(\frac{\mathcal{D}(\mathbf{z})}{\bar{\mathcal{D}}(\mathbf{z})}\right) \leq \sum_j \left[u_j(\hat{z}_j) - \mathbb{E}_{z\leftarrow\bar{\mathcal{D}}_j}[u_j]\right] \quad (10)$$

Let A and B be the left and right terms of the equation 10. If we can bound A and B, then we will be able to conclude by the relative error lemma.

② Now, we bound A. We write $\bar{\sigma}_j = (1 + \delta_{\sigma_j})\sigma_j$, where each $|\delta_{\sigma_j}| \leq \delta$ by hypothesis. Developing u_j yields:

$$u_j(z_j) = \frac{1}{2(1+\delta_{\sigma_j})^2\sigma_j^2}\left[(c_j - \bar{c_j})^2 + 2(c_j - \bar{c_j})(z_j - c_j) - (2\delta_{\sigma_j} + \delta_{\sigma_j}^2)(z_j - c_j)^2\right] \tag{11}$$

In order to bound $c_j - \bar{c_j}$, we note that numerically, c_j is exactly $t_j + \langle \mathbf{t} - \mathbf{z}, \mathbf{l}_j \rangle$, where $\mathbf{l}_j$ is the j-th row of $(\mathbf{L}^t - I_n)$. Noting $\bar{\mathbf{t}} = \mathbf{t} + \delta_{\mathbf{t}}$, $\bar{\mathbf{l}}_j = \mathbf{l}_j + \delta_{\mathbf{l}_j}$ and $L = \|\mathbf{L}\|_{\max}$, we have:

$$\bar{c_j} = c_j + \delta_{\mathbf{t},j} + \langle \delta_{\mathbf{t}}, \mathbf{l}_j \rangle + \langle \mathbf{t} - \mathbf{z}, \delta_{\mathbf{l}_j} \rangle + \langle \delta_{\mathbf{t}}, \delta_{\mathbf{l}_j} \rangle$$

Thus

$$\begin{aligned} |\bar{c_j} - c_j| &\leq \delta_{\mathbf{t},j} + \|\delta_{\mathbf{t}}\|\|\mathbf{l}_j\| + \|\delta_{\mathbf{l}_j}\|\|\mathbf{t} - \mathbf{z}\| + \|\delta_{\mathbf{t}}\|\|\delta_{\mathbf{l}_j}\| \\ &\leq \delta(nL + 1) + \delta nL\sigma\sqrt{2\pi} \cdot \|\mathbf{B}^{-1}\|_2 + \delta^2 nL \\ &\leq \delta \cdot T \end{aligned} \tag{12}$$

In equation 12, we used the fact that:

- $\|\delta_{\mathbf{t}}\| \leq \delta\sqrt{n}$
- $\|\delta_{\mathbf{l}_j}\| \leq \delta\|\mathbf{l}_j\| \leq \delta\sqrt{n}L$
- $\|\mathbf{t} - \mathbf{z}\| \leq \|\mathbf{c} - \mathbf{v}\| \cdot \|\mathbf{B}^{-1}\|_2 \leq \sigma\sqrt{2\pi n} \cdot \|\mathbf{B}^{-1}\|_2$, with the last inequality coming from [MR07, Lemma 4.4] (see lemma 10 in the appendix)

We have:

$$\begin{aligned} A &= \sum_j \frac{1}{2(1+\delta_{\sigma_j})^2\sigma_j^2} \left[2(c_j - \bar{c_j})(\hat{z}_j - c_j - \mathbb{E}_{z_j \leftarrow \mathcal{D}_j}[z_j - c_j]) - (2\delta_{\sigma_j} + \delta_{\sigma_j}^2)[(\hat{z}_j - c_j)^2 - \mathbb{E}_{z_j \leftarrow \mathcal{D}_j}[(z_j - c_j)^2]]\right] \\ |A| &\leq \sum_j \frac{1.1}{2\sigma_j^2} \left[2|c_j - \bar{c_j}|(|\hat{z}_j - c_j| + \sqrt{2\pi}\epsilon\sigma_j) + 2\delta[(\hat{z}_j - c_j)^2 + \sigma_j^2 + 2\pi\epsilon\sigma_j^2]\right] \\ &\leq \frac{1.1}{\sigma^2}\sum_j \left[\delta T(\|\tilde{\mathbf{b}}_j\|^2 \cdot |\hat{z}_j - c_j| + \|\tilde{\mathbf{b}}_j\|\sqrt{2\pi}\epsilon\sigma) + \delta[\|\tilde{\mathbf{b}}_j\|^2(\hat{z}_j - c_j)^2 + \sigma^2 + 2\pi\epsilon\sigma^2]\right] \\ &\leq \frac{1.1\delta}{\sigma^2} \left[T \max_j \|\tilde{\mathbf{b}}_j\|(\|\mathbf{v} - \mathbf{c}\|_1 + \sqrt{2\pi}\epsilon\sigma n) + [\|\mathbf{v} - \mathbf{c}\|^2 + n\sigma^2 + 2\pi n\epsilon\sigma^2]\right] \\ &\leq 1.1\delta \left[T(n\sqrt{2\pi} + \sqrt{2\pi}\epsilon n)/\eta_\epsilon(\mathbb{Z}^n) + [2\pi n + n + 2\pi n\epsilon]\right] \\ &\leq 1.2\delta n \left[T\sqrt{2\pi}/\eta_\epsilon(\mathbb{Z}^n) + 2\pi + 1\right] \end{aligned} \tag{13}$$

In equation 13, the first line develops the formula for A by using equation 11. For the second line, we use [MR07, Lemma 4.2] (see lemma 9 in the appendix) to bound the two expected values and the term 1.1 to absorb parasitic terms in δ_{σ_j} and ϵ.

The third line replaces σ_j by $\sigma/\|\tilde{\mathbf{b}}_j\|$ and $|c_j - \bar{c_j}|$ by the bound $\delta \cdot T$ from equation 12. For the fourth line, we notice that $\sum_j \|\tilde{\mathbf{b}}_j\| \cdot |\hat{z}_j - c_j| = \|\mathbf{v} - \mathbf{c}\|_1$ and $\sum_j \|\tilde{\mathbf{b}}_j\|^2 \cdot (\hat{z}_j - c_j)^2 = \|\mathbf{v} - \mathbf{c}\|_2^2$ (both equalities follow directly from the lemma 4.4 of [GPV08]).

In the fifth line, we use the bounds $\|\mathbf{v}-\mathbf{c}\|_2 \leq \sigma\sqrt{2\pi n}$, and $\|\mathbf{v}-\mathbf{c}\|_1 \leq \sigma n\sqrt{2\pi}$: the first one comes from [MR07, Lemma 4.4], and the second one follows from the fact that there exists a vector $\mathbf{u}$ with coefficients being only ± 1 such that $\|\mathbf{v}-\mathbf{c}\|_1 = |\langle \mathbf{v}-\mathbf{c}, \mathbf{u}\rangle|$. Applying the Cauchy-Schwartz theorem yields the bound. The last line simplifies as much as possible the expression.

③ We now bound B, the right part of equation 10. We can write u_j as follows:

$$u_j(z_j) = \frac{1}{\bar{\sigma}_j^2}\left[-(1+\delta_{\sigma_j})^2(c_j-\bar{c_j})^2 + 2(1+\delta_{\sigma_j})^2(c_j-\bar{c_j})(z_j-c_j) - (2\delta_{\sigma_j}+\delta_{\sigma_j}^2)(z_j-\bar{c_j})^2\right] \tag{14}$$

To bound B, we replace the u_j in each $u_j(\hat{z}_j)$ by the expression in equation 11, and the u_j in each $\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}_j}[u_j]$ by the expression of equation 14. This yields:

$$\begin{aligned} |B| \leq \sum_j \frac{1.1(c_j-\bar{c_j})^2}{\bar{\sigma}_j^2} &+ \sum_j \frac{1.1}{2\sigma_j^2}\left[2|c_j-\bar{c_j}|\cdot|\hat{z}_j-c_j| + 2|\delta_{\sigma_j}|\cdot|\hat{z}_j-c_j|^2\right] \\ &+ \sum_j \frac{1}{2\bar{\sigma}_j^2}\left[2|c_j-\bar{c_j}|\cdot|\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}_j}[z_j-\bar{c_j}]| + 2|\delta_{\sigma_j}|\cdot\mathbb{E}_{z\leftarrow\bar{\mathcal{D}}_j}[(z_j-\bar{c_j})^2]\right] \\ \leq \frac{1.1n(\delta\cdot T)^2}{\eta_\epsilon(\mathbb{Z}^n)^2} &+ 1.1n\delta\left[T\sqrt{2\pi}/\eta_\epsilon(\mathbb{Z}^n) + 2\pi + 1\right] \\ &+ 1.1\delta\epsilon\left[T\sqrt{2\pi}/\eta_\epsilon(\mathbb{Z}^n) + 2\pi\right], \end{aligned}$$

where the bound over $|B|$ is obtained using the same techniques as for $|A|$. Overall, we see that $|A|, |B| \leq C$.

④ To conclude, we have $-C \leq \log(\frac{\mathcal{D}(\mathbf{z})}{\bar{\mathcal{D}}(\mathbf{z})}) \leq C$, so $e^{-C} \leq \frac{\mathcal{D}(\mathbf{z})}{\bar{\mathcal{D}}(\mathbf{z})} \leq e^C$. □

Practical implications of Lemma 8. We can now easily – given a few simplifications – apply the relative error lemma. Even though in theory we have $\|\mathbf{M}\|_2 \leq n\|\mathbf{M}\|_{\text{GS}}$, this is a worst-case bound [Pei10, Lemma 5.1]. In practice, it is reasonable to assume $\|\mathbf{B}\|_2 = O(\sqrt{\log n}) \cdot \|\mathbf{B}\|_{\text{GS}}$, with a small constant factor in the big O [Pre15, Section 6.5.2].[9]

In addition, we make the simplification $\|\mathbf{B}^{-1}\|_{\text{GS}} \approx \|\mathbf{B}\|_{\text{GS}}^{-1}$,[10] which gives $\sigma\|\mathbf{B}^{-1}\|_2 \approx \sqrt{\log n} \cdot \eta_\epsilon(\mathbb{Z}^n)$. It is also easy to make $\|\mathbf{L}\|_{\max} = 1$, so we consider that this is the case. Removing terms which are clearly negligible, and since $e^C \underset{C\to 0}{\sim} 1 + C$, we have

$$1 - C' \leq \frac{\bar{\mathcal{D}}}{\mathcal{D}} \leq 1 + C', \qquad \text{with } C' \approx 8 \cdot n^2\sqrt{\log n} \cdot \delta. \tag{15}$$

For typical values of n (say, $n = 1024$), we can take $\delta = 2^{-37}/C' \approx 2^{-61}$, which is secure as per the argument of section 3.3. Therefore, precision 61 is sufficient to securely use Klein's sampler.

[9] Or alternatively, $\|\mathbf{B}\|_2 = O(\sqrt{\log q}) \cdot \|\mathbf{B}\|_{\text{GS}}$ (see e.g. [Pei10, Lemma 5.2])
[10] As an example, for NTRU matrices, this is true up to a factor 1.17^2 [DLP14]

5 Conclusion and Open Problems

To conclude, we expose a few perspectives and open problems that we have encountered. Most of them are related to implementing the techniques we have introduced, but in our opinion extending our techniques to other cryptographic schemes is probably the most challenging problem.

The revisited table approach. It remains to see how the CoDF-based algorithm we proposed in section 4.2 can be efficiently implemented and protected against side-channel attacks. Our approach also seems highly composable with existing techniques, and it would be interesting to find combinations that achieve better overall efficiency.[11] Per example, a natural question would be to see how to combine it with Knuth-Yao trees (see e.g. [DG14]).

Rejection sampling in practice. The techniques that we described in section 4.3 remain to be implemented, to assess their efficiency and whether they can easily be made impervious against side-channel attacks.

Precision analysis of trapdoor samplers. It would be interesting to apply the precision analysis of section 4.5 to other samplers, such as the one of [Pei10]. A promising candidate would be a randomized variant of Ducas and Prest's fast Fourier nearest plane [DP16]. The fast Fourier transform is known to be very stable numerically, and since this algorithm has the same structure, it seems likely that it will inherit this stability and require less than 53 bits of precision.

Other types of schemes and problems. All the applications that we give are related to signature schemes and in the context of search problems. For future work, we would like to apply our results to other schemes, in particular encryption schemes, by following the ideas of [BLL+15] or developing new ones. Achieving the same efficiency in the presence of decision problems is a related – and in our opinion, important – question.

Acknowledgements

I would like to thank Fabrice Mouhartem, Damien Stehlé and Michael Walter for useful discussions. I would also like to thank Ange Martinelli and Thomas Ricosset for reviewing an earlier version of this work.

This work has been supported in part by the BPI-funded project "RISQ".

[11] In a sense, this is what we did at the end of section 4.2, as the algorithm 10 from [DDLL13] is meant to be used in conjunction with two other algorithms (11 and 12).

References

ABB10a. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [Gil10], pages 553–572. 3

ABB10b. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Rabin [Rab10], pages 98–115. 3

ADPS16. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343. USENIX Association, 2016. 2

Ass06. Walter Van Assche. Pad and hermite-pad approximation and orthogonality, 2006. 28

Bab85. L Babai. On lova´sz' lattice reduction and the nearest lattice point problem. In *Proceedings on STACS 85 2Nd Annual Symposium on Theoretical Aspects of Computer Science*, New York, NY, USA, 1985. Springer-Verlag New York, Inc. 17

Bab86. László Babai. On lovasz' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1), 1986. 17

BGG+14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Nguyen and Oswald [NO14], pages 533–556. 3

BGM+16. Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 209–224, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany. 5

BLL+15. 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 Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 3–24, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany. 5, 6, 7, 9, 10, 17, 23

Boy10. Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517, Paris, France, May 26–28, 2010. Springer, Heidelberg, Germany. 3

Boy13. Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg, Germany. 3

Cac97. Christian Cachin. *Entropy Measures and Unconditional Security in Cryptography*. PhD thesis, 1997. 1

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [Gil10], pages 523–552. 3

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal Gaussians. In Ran Canetti and Juan A.

Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 40–56, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. 2, 3, 4, 15, 16, 17, 23

DG14. 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. 23

DLP14. Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany. 5, 18, 19, 22

DN12a. Léo Ducas and Phong Q. Nguyen. Faster Gaussian lattice sampling using lazy floating-point arithmetic. In Wang and Sako [WS12], pages 415–432. 5, 18

DN12b. Léo Ducas and Phong Q. Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUSign countermeasures. In Wang and Sako [WS12], pages 433–450. 2

DP16. Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In Sergei A. Abramov, Eugene V. Zima, and Xiao-Shan Gao, editors, *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2016, Waterloo, ON, Canada, July 19-22, 2016*, pages 191–198. ACM, 2016. 23

EFGT17. Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Generalized howgrave-grahamszydlo and side-channel attacks against BLISS. Publication status unknown, 2017. https://almasty.lip6.fr/~espitau/bin/SCBliss. 15, 16

GGH97. Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 112–131, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany. 2

Gil10. Henri Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany. 24

GJSS01. Craig Gentry, Jakob Jonsson, Jacques Stern, and Michael Szydlo. Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 1–20, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany. 2

GLP12. Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 530–547, Leuven, Belgium, September 9–12, 2012. Springer, Heidelberg, Germany. 3

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press. 3, 4, 12, 17, 18, 21

GS02. Craig Gentry and Michael Szydlo. Cryptanalysis of the revised NTRU signature scheme. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 299–320, Amsterdam, The Netherlands, April 28 – May 2, 2002. Springer, Heidelberg, Germany. 2

HHGP+03. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Marc Joye, editor, *CT-RSA 2003*, volume 2612 of *LNCS*, pages 122–140, San Francisco, CA, USA, April 13–17, 2003. Springer, Heidelberg, Germany. 2

Kle00. Philip N. Klein. Finding the closest lattice vector when it's unusually close. In *SODA*, 2000. 3, 4, 17, 18

LD13. Tancr`de Lepoint and Léo Ducas. Proof-of-concept software implementation of BLISS, 2013. http://bliss.di.ens.fr/bliss-06-13-2013.zip. 16

LP15. Vadim Lyubashevsky and Thomas Prest. Quadratic time, linear space algorithms for Gram-Schmidt orthogonalization and Gaussian sampling in structured lattices. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 789–815, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. 4

LPSS14. San Ling, Duong Hieu Phan, Damien Stehlé, and Ron Steinfeld. Hardness of k-LWE and applications in traitor tracing. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 315–334, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. 5

LSS14. Adeline Langlois, Damien Stehlé, and Ron Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In Nguyen and Oswald [NO14], pages 239–256. 5

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany. 2

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In Pointcheval and Johansson [PJ12], pages 738–755. 3

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Pointcheval and Johansson [PJ12], pages 700–718. 3, 17

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press. 19

MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 2007. 6, 21, 22, 28

MW17. Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. CRYPTO, 2017. http://eprint.iacr.org/2017/259. 3, 4, 5, 10, 11

NO14. Phong Q. Nguyen and Elisabeth Oswald, editors. *EUROCRYPT 2014*, volume 8441 of *LNCS*, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. 24, 26

NR06. Phong Q. Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 271–288, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. 2

Pad92. Henri Padé. *Sur la représentation approchée d'une fonction par des fractions rationnelles*. Phd thesis, 1892. 28

PDG14. Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 353–370, Busan, South Korea, September 23–26, 2014. Springer, Heidelberg, Germany. 3, 5, 8, 17

Pei10. Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Rabin [Rab10], pages 80–97. 2, 3, 4, 12, 17, 19, 22, 23

PJ12. David Pointcheval and Thomas Johansson, editors. *EUROCRYPT 2012*, volume 7237 of *LNCS*, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany. 26

POG15. Thomas Pöppelmann, Tobias Oder, and Tim Güneysu. High-performance ideal lattice-based cryptography on 8-bit ATxmega microcontrollers. In Kristin E. Lauter and Francisco Rodríguez-Henríquez, editors, *LATINCRYPT 2015*, volume 9230 of *LNCS*, pages 346–365, Guadalajara, Mexico, August 23–26, 2015. Springer, Heidelberg, Germany. 3

Pop14. Thomas Poppelmann. Proof-of-concept hardware implementation of BLISS, 2014. https://www.sha.rub.de/media/attachments/files/2014/09/lattice_processor_final_publication.zip. 16

Pre15. Thomas Prest. *Gaussian Sampling in Lattice-Based Cryptography*. Theses, École Normale Supérieure, December 2015. 4, 5, 19, 22

Rab10. Tal Rabin, editor. *CRYPTO 2010*, volume 6223 of *LNCS*, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany. 24, 27

R61. Alfrd Rnyi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, Berkeley, Calif., 1961. University of California Press. 2

Saa15. Markku-Juhani O. Saarinen. Gaussian sampling precision in lattice cryptography. Cryptology ePrint Archive, Report 2015/953, 2015. http://eprint.iacr.org/2015/953. 5

Str14. StrongSwan. Bimodal lattice signature scheme (BLISS), 2014. https://wiki.strongswan.org/projects/strongswan/wiki/BLISS. 2, 16

TT15. Katsuyuki Takashima and Atsushi Takayasu. Tighter security for efficient lattice cryptography via the Rényi divergence of optimized orders. In Man Ho Au and Atsuko Miyaji, editors, *ProvSec 2015*, volume 9451 of *LNCS*, pages 412–431, Kanazawa, Japan, November 24–26, 2015. Springer, Heidelberg, Germany. 5

vEH14. Tim van Erven and Peter Harremoës. c. *IEEE Trans. Information Theory*, 60(7):3797–3820, 2014. 7

Wan10. Andrew Wan. *Learning, Cryptography, and the Average Case*. Phd thesis, Columbia University, 2010. 2

WS12. Xiaoyun Wang and Kazue Sako, editors. *ASIACRYPT 2012*, volume 7658 of *LNCS*, Beijing, China, December 2–6, 2012. Springer, Heidelberg, Germany. 25

A Appendix

A.1 Padé Approximants

In this section, we give a very succinct explanation of Padé approximants in the context that interests us. A more detailed introduction can be found in e.g. [Ass06]. Informally, Padé approximants can be described as generalizations of Taylor series, as the latter approximate $(n+1)$-differentiable functions as

$$f(x) = P_n(x) + O(z^{n+1}),$$

with P_n a polynomial of degree n, whereas Padé approximants provide an approximation of the form

$$Q_m(x)f(x) = P_n(x) + O(z^{n+m+1}),$$

with P_n and Q_m being polynomials of degree n and m.

While Padé approximants are in general much trickier to compute than their Taylor series counterparts, such approximants are well known for the exponential function. Let $m = n$ and

$$P_n(x) = Q_n(-x) = \sum_{k=0}^{n} \frac{(2n-k)!n!x^k}{(2n)!(n-k)!k!}. \tag{16}$$

Then we have [Pad92]:

$$\left| \frac{P_n(x)}{Q_n(x)} - e^x \right| = \frac{(n!)^2 x^{2n+1} e^x}{(2n)!(2n+1)!}(1 + o(1)) \tag{17}$$

Since our goal is to have a relative error less than 2^{-37}, taking $(m, |x|) \leq (4, .5)$ or $(m, |x|) \leq (5, 1)$ is sufficient.

A.2 Classical Lemmas

Lemma 9. *[MR07, Lemma 4.2] Let Λ be a n-dimensional lattice, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^n$ a vector of norm 1 and reals $\epsilon \in (0,1)$, $\sigma \geq 2\eta_\epsilon(\Lambda)$. The following inequalities hold:*

$$\left|\mathbb{E}_{\mathbf{x} \leftarrow D_{\Lambda,\sigma,\mathbf{c}}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u} \rangle]\right| \leq \frac{\sqrt{2\pi}\epsilon\sigma}{1-\epsilon}$$

$$\left|\mathbb{E}_{\mathbf{x} \leftarrow D_{\Lambda,\sigma,\mathbf{c}}}[\langle \mathbf{x} - \mathbf{c}, \mathbf{u} \rangle^2] - \sigma^2\right| \leq \frac{2\pi\epsilon\sigma^2}{1-\epsilon}$$

Lemma 10. *[MR07, Lemma 4.4] Let Λ be a n-dimensional lattice, $\mathbf{c} \in \mathbb{R}^n$, and reals $\epsilon \in (0,1)$, $\sigma \geq \eta_\epsilon(\Lambda)$. We have:*

$$\mathbb{P}_{\mathbf{x} \leftarrow D_{\Lambda,\sigma,\mathbf{c}}}[\|\mathbf{x} - \mathbf{c}\| \geq \sigma\sqrt{2\pi n}] \leq \frac{1+\epsilon}{1-\epsilon} 2^{-n}$$