Exact Lattice Sampling from Non-Gaussian Distributions

Maxime Plançon[1⋆] and Thomas Prest[2]

[1] IBM Research and ETH Zurich, Switzerland
mpl@zurich.ibm.com
[2] PQShield, United Kingdom
thomas.prest@pqshield.com

Abstract. We propose a new framework for (trapdoor) sampling over lattices. Our framework can be instantiated in a number of ways. It allows for example to sample from uniform, affine and "product affine" distributions. Another salient point of our framework is that the output distributions of our samplers are perfectly indistinguishable from ideal ones, in contrast with classical samplers that are statistically indistinguishable. One caveat of our framework is that all our current instantiations entail a rather large standard deviation.

Keywords. Trapdoor sampling, lattice trapdoors, squaremonic functions, regular algorithms.

1 Introduction

Sampling over a lattice – usually using a trapdoor – is a fundamental building block in lattice-based cryptography. Since its inception [23,21], it has seen a myriad of applications such as full-domain hash signature schemes [21], identity-based encryption or IBE [21], hierarchical IBE [12,3,4], attribute-based encryption [10], standard model signatures and so on.

Given its importance, surprisingly few sampling methods have been proposed. The most prominent is arguably the Klein/GPV sampler [23,21], a randomized variant of Babai's nearest plane algorithm. Analogously, Peikert's sampler [28] randomizes Babai's round-off algorithm. Both samplers can sample over any lattice, provided a (preferably) short basis. The Micciancio-Peikert framework [26] and its variations operate at a slightly different level by constructing pseudorandom lattices along with trapdoors that allow to sample efficiently.

These proposals share two notable common points. First, they all sample from discrete *Gaussian* distributions. Gaussians come with their share of challenges in terms of implementation, precision analysis and side-channel analysis, and have often been replaced with simpler distributions whenever possible [6,16,9].

⋆ Most of this work was done while Maxime Plançon was an intern at PQShield.

To the best of our knowledge, the only attempt [25] to rely on other distributions than discrete Gaussians was restricted to the Micciancio-Peikert framework. A second common point is that they do not sample perfectly from a discretized ideal distribution, but statistically close to it. A blueprint for performing exact lattice sampling is proposed at the end of [11]; it is rather involved as it entails infinite sums of transcendental functions. To the best of our knowledge, neither [25] nor [11] have been implemented.

The motivation of this work is to propose alternative trapdoor samplers that lift the two aforementioned limitations: (a) being restricted to Gaussian distributions, (b) achieving only statistical correctness instead of the stronger notion of perfect correctness. In itself, lifting these limitations is conceptually interesting, and may further our theoretic comprehension of lattice sampling. From a practical perspective, a new approach with different strengths and weaknesses provides more avenues for optimization.

1.1 Our Contribution

We propose a new framework for lattice (trapdoor) sampling. At a high level, it requires two components. First, we require an $\mathcal{L}$-regular algorithm; intuitively, a regular algorithm maps the ambient space to a lattice $\mathcal{L}$ in a way that defines a $\mathcal{L}$-regular tiling of the space. This notion provides a natural abstraction of Babai's nearest plane and round-off algorithms, as well as any exact closest vector problem (or CVP) solver.

The second component is a $\mathcal{T}$-squaremonic function; the term *squaremonic* is a portmanteau of *square* and *harmonic*. This notion is a variation of harmonic functions over lattice tiles instead of balls. The key property of squaremonic functions is that rounding them over a lattice is equivalent (by translation) to discretizing them over the same lattice. The interplay between regular algorithms and squaremonic functions gives us a class of lattice samplers, corresponding to various instances of our framework. Our framework and its instantiations have two interesting properties.

- **Non-Gaussian distributions.** We can sample from uniform, affine and "product affine" distributions, discretized over a subset of a lattice – typically, its intersection with a L_p ball. This contrasts with classical lattice sampling algorithms, which are restricted to Gaussian distributions – with the exception of [25] in the setting of [26].
- **Exact sampling.** The output distribution of our algorithms are *exact* discrete distributions over a lattice, perfectly independent of the basis used. In comparison, existing lattice (trapdoor) samplers [21,28,26], entail a trade-off between the standard deviation of the (Gaussian) distribution and the correctness of the sampler (i.e. the divergence of its output from an ideal distribution), see [21,18,29]. In our case, there is a trade-off between the standard deviation of the distribution and the *running time* of the sampler. While the practical impact of this exactness is unclear, we believe it is conceptually interesting.

At a technical level, our approach is simple; one possible instantiation is to sample $\mathbf{x} \leftarrow f$ from a continuous distribution f, compute $\mathbf{B}\left\lfloor \mathbf{B}^{-1}\mathbf{x} \right\rceil$ and apply a simple rejection sampling step. We note that works by Regev [31] and Gentry, Peikert and Vaikuntanathan [21] considered doing exactly this. However, both works took f to be a continuous Gaussian; achieving statistical closeness then required an exponential standard deviation, and this approach was deemed non-viable by [21]. What allows us to unlock this situation is to rely on different distributions; the ones we choose are naturally more amenable to this approach.

One current drawback in our approach is that it entails standard deviations that are higher than the state of the art. Compared to the "classical" samplers of [21,28], the Euclidean norm can be larger by a factor $O(n^{1.5})$. We therefore consider reducing this standard deviation as a relevant and important open question.

We are not aware of any straightforward way to apply our approach with Gaussians or, conversely, to adapt the [21] sampler to the distributions we have chosen. Again, we see future results in either direction (either positive or negative) as interesting open questions.

Finally, we note that our main motivation for this work was the constructive side of lattice sampling (e.g. trapdoor sampling), and this is reflected in this document. However, *Gaussian* sampling over lattices has also been studied in the context of theoretical cryptanalysis and computational complexity theory [23,32,1,2].

1.2 Related Works

A few lattice sampling frameworks have been proposed; foundational algorithmic works are [23,21,28], which first proposed trapdoor samplers. The Micciancio-Peikert framework [26] and its follow-up works [13] directly construct lattices that can easily be sampled from using [21,28]. Note that unlike our work, these works only considered statistically correct sampling and Gaussian distributions. This is also true for their follow-up works, with the exception of the ones discussed below.

Sampling from non-Gaussian distributions in the Micciancio-Peikert framework was considered by [25], and sampling exactly via analytical techniques was studied by [11]. We note that [31,21] considered a similar idea to ours. Unfortunately, both works consider instantiating it with Gaussians, leading to statistical correctness and exponential standard deviation.

1.3 Acknowledgements

Thomas Prest is supported by the Innovate UK Research Grant 104423 (PQ Cybersecurity).

2 Preliminaries

2.1 Lattices

Linear Algebra. We use the column convention for vectors, which are written in bold lower case letters $\mathbf{v}$. Matrices are written in bold upper case letters $\mathbf{M}$. The notation $\mathbf{M} = (\mathbf{b}_1 \dots, \mathbf{b}_n)$ means that the i-th column of the matrix $\mathbf{M}$ is $\mathbf{b}_i$. The identity matrix of rank n is written $\mathbf{I}_n$ and the set of $n \times n$ invertible matrices with coefficients in a ring R is written $\mathrm{GL}_n(R)$.

Given a matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, its Gram-Schmidt orthogonalization (GSO) is the unique decomposition $\mathbf{B} = \widetilde{\mathbf{B}} \cdot \mathbf{U}$ such that $\mathbf{U} \in \mathbb{R}^{n \times n}$ is upper-triangular with ones on the diagonal and the colmuns of $\widetilde{\mathbf{B}}$ are pairwise orthogonal. For $n \in \mathbb{N}$, $r \in \mathbb{R}_+$ and $p \in \{1, 2, \infty\}$, we define the centered ℓ_p hyperball of radius r as $\mathcal{B}_p^n(r) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_p \leq r\}$. We introduce $s_1(\mathbf{B}) = \max_{\mathbf{x} \in \mathbb{R}^n \setminus 0^n} \frac{\|\mathbf{Bx}\|}{\|\mathbf{x}\|}$ to be the operator norm of a matrix $\mathbf{B}$ as an endomorphism of $(\mathbb{R}^n, \|\cdot\|_2)$, also known as spectral norm. The value of $s_1(\mathbf{B})$ is also the largest eigenvalue of $\mathbf{B}^t\mathbf{B}$. The fundamental parallelepiped associated to $\mathbf{B} \in \mathbb{R}^{m \times n}$ is $\mathcal{P}(\mathbf{B}) = \mathbf{B} \cdot [-1/2, 1/2]^n$.

Lattices. A lattice is a discrete subgroup of $\mathbb{R}^m$. Given a set $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$ of linearly independent vectors in $\mathbb{R}^m$, we note $\mathcal{L}(\mathbf{B})$ the lattice generated by $\mathbf{B}$, that is

$$\mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^{n} c_i \mathbf{b}_i,\ \mathbf{c} \in \mathbb{Z}^m\}.$$

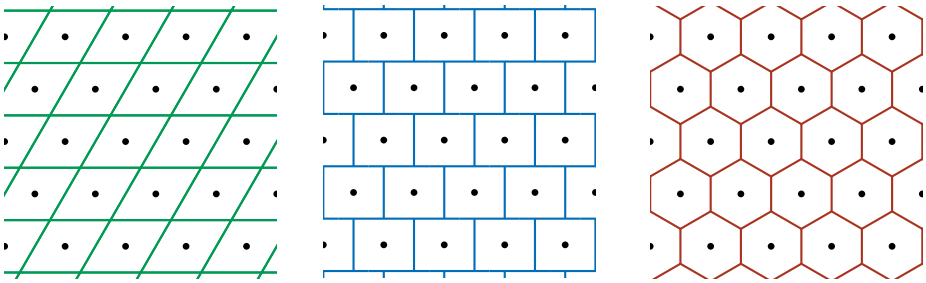
In such a case, we say that $\mathbf{B}$ is a basis of $\mathcal{L}(\mathbf{B})$. In this document, we only consider full-rank lattices; for lattices of the form $\mathcal{L} = \mathcal{L}(\mathbf{B})$, it implies that $\mathbf{B}$ is a square invertible matrix. While our arguments readily extend to the general case, this choice makes their exposition simpler. Given a lattice $\mathcal{L}$, we note $\mathrm{Vol}(\mathcal{L})$ its volume, that is the absolute value of the determinant of any basis $\mathbf{B}$ of $\mathcal{L}$: $\mathrm{Vol}(\mathcal{L}) = |\det \mathbf{B}|$. One can check that all the bases of $\mathcal{L}$ have the same determinant (in absolute value), and this definition is therefore consistent. We call a *trapdoor* of a lattice $\mathcal{L}$ any set τ that characterizes the lattice, and write $\mathcal{L}(\tau)$ the lattice characterized by the trapdoor τ. When the trapdoor is a basis $\mathbf{B}$, the notation $\mathcal{L}(\mathbf{B})$ is consistent. Finally, the Voronoi cell of a lattice can be defined as follows :

$$\mathcal{V}(\mathcal{L}) = \{\mathbf{z} \in \mathbb{R}^n \mid \forall \mathbf{x} \in \mathcal{L},\ \|\mathbf{z}\|_2 \leq \|\mathbf{x} - \mathbf{z}\|_2\}.$$

Informally, the Voronoi cell of a lattice is the set of vectors that are closer to the origin than to any other lattice point (see [14] for further information).

Lattice Tilings. A family $\{\mathcal{T}_i\}_{i \in I}$ of sets in $\mathbb{R}^n$ is a tiling or a tessellation of $\mathbb{R}^n$ and the sets are called tiles if the union of sets covers $\mathbb{R}^n$ and the set interiors are mutually disjoint. We focus our study on *lattice tilings*, which are tilings of the form $\mathcal{T} + \mathcal{L} = \{\mathcal{T} + x\}_{x \in \mathcal{L}}$ for some lattice $\mathcal{L}$. For such tilings, $\mathcal{T}$ is called a

prototile of the tiling. We note that if $\mathcal{T}$ is a prototile of $\mathcal{L}$, then $\mathrm{Vol}(\mathcal{T}) = \mathrm{Vol}(\mathcal{L})$. A tiling is called convex if all the tiles are compact convex bodies. The prototile of a convex lattice tiling is called a *parallelohedron*.[1] Figure 1 displays a few example of convex lattice tilings, for a fixed lattice but different parallelohedra.



(a) $\mathcal{T} = \mathcal{P}(\mathbf{B})$ (b) $\mathcal{T} = \mathcal{P}(\widetilde{\mathbf{B}})$ (c) $\mathcal{T} = \mathcal{V}(\mathcal{L})$

Fig. 1: A few examples of convex lattice tilings $\mathcal{L}(\mathbf{B}) + \mathcal{T}$, for different values of the prototile $\mathcal{T}$: the fundamental parallelelepiped associated to $\mathbf{B}$, $\widetilde{\mathbf{B}}$ or the Voronoi cell of $\mathcal{L}$.

In this document, we almost exclusively work with the tile $\mathcal{T} = \mathcal{P}(\mathbf{B})$, so we introduce the notation $M_p(\mathbf{B}) = \sup\{\|\mathbf{x}\|_p \mid \mathbf{x} \in \mathcal{P}(\mathbf{B})\}$, which is also half the operator norm of $\mathbf{B}$ as a linear operator from $(\mathbb{R}^n, \|\cdot\|_\infty)$ to $(\mathbb{R}^n, \|\cdot\|_p)$. For $a \in \mathbb{R}$, we also note $\overrightarrow{a} = (a, \ldots, a)$.

$\mathcal{L}$-regular algorithms. The first of the two ingredients in our framework is the notion of ($\mathcal{L}$-)regular algorithms.

Definition 1 (Regular algorithm). *Let $\mathbf{T}$ be a set of trapdoors. Let $\mathcal{A} : \mathbf{T} \times \mathbb{R}^n \to \mathbb{R}^n$ be a deterministic algorithm taking as an input a trapdoor τ of a lattice $\mathcal{L}$, a target vector $\mathbf{t}$, and outputting a lattice point $\mathbf{v}$.*[2]

We say that $\mathcal{A}$ is $\mathcal{L}$-regular if for all $\tau \in \mathbf{T}$ such that $\mathcal{L} = \mathcal{L}(\tau)$, $\mathcal{A}(\tau, \mathbf{0}) = \mathbf{0}$ and if the set of points $\mathbf{y}$ such that the following equality holds:

$$\forall \mathbf{x} \in \mathbb{R}^n, \mathcal{A}(\tau, \mathbf{x} + \mathbf{y}) = \mathcal{A}(\tau, \mathbf{x}) + \mathbf{y}, \tag{1}$$

is exactly the lattice $\mathcal{L}$. If $\mathcal{A}$ is $\mathcal{L}$-regular for any lattice $\mathcal{L} \in \mathrm{GL}_n(\mathbb{R})/\mathrm{GL}_n(\mathbb{Z})$, i.e for any couple $(\tau, \mathcal{L}(\tau))$, (1) holds for exactly $\mathbf{y} \in \mathcal{L}(\tau)$, we simply say that $\mathcal{A}$ is regular.

[1] It has been conjectured by Voronoi [35] that every parallelohedron $\mathcal{T} \subset \mathcal{R}^n$ is affine equivalent to the Voronoi cell of some lattice $\mathcal{L}' \subset \mathcal{R}^n$.

[2] From a practical viewpoint, one can think of the trapdoor as a short basis. The trapdoor can contain more information, such as the Gram-Schmidt orthogonalization (or GSO) of the basis, or any precomputation on the lattice.

If $\mathcal{A}$ is $\mathcal{L}$-regular, then $\mathbf{x} \mapsto \mathcal{A}(\tau, \mathbf{x}) - \mathbf{x}$ is $\mathcal{L}$-periodic and admits $\mathbf{0}$ as a fixed point. Any $\mathcal{L}$-regular algorithm induces a $\mathcal{L}$-tiling. Indeed, for $\mathbf{v} \in \mathcal{L}$, let:

$$\mathcal{T}_{\mathbf{v}} = \{\mathbf{x} \in \mathbb{R}^n | \mathcal{A}(\tau, \mathbf{x}) = \mathbf{v}\}.$$

One can easily show that $\{\mathcal{T}_{\mathbf{v}}\}_{\mathbf{v} \in \mathcal{L}}$ is a $\mathcal{L}$-tiling. Finally, it is easy to show that the image of $\mathbf{x} \mapsto \mathcal{A}(\tau, \mathbf{x})$ is exactly the lattice $\mathcal{L}(\tau)$.

Examples of $\mathcal{L}$-regular algorithms include Babai's algorithms [7]. The round-off algorithm (Algorithm 1) induces the lattice tiling illustrated in Figure 1a. The nearest plane algorithm (Algorithm 2) induces the lattice tiling illustrated in Figure 1b. Any exact CVP solver (*i.e.* any algorithm that outputs a closest lattice point to the target) is also a valid example of $\mathcal{L}$-regular algorithm, and its induced tiling is the Voronoi diagram of $\mathcal{L}$, illustrated in Figure 1c.

Algorithm 1: Babai round-off algorithm

Require: A basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ of $\mathcal{L}$, a target $\mathbf{x} \in \mathbb{R}^n$
Ensure: $\mathbf{v} \in \mathcal{L} \cap \{\mathbf{x} + \mathcal{P}(\mathbf{B})\}$
1: $\mathbf{t} \leftarrow \mathbf{B}^{-1} \cdot \mathbf{x}$
2: **for** $i = 1, \ldots, n$ **do**
3: $\quad z_i = \lfloor t_i \rceil$
4: **end for**
5: **return** $\mathbf{v} = \mathbf{B} \cdot \mathbf{z}$

Algorithm 2: Babai nearest plane algorithm

Require: A basis $\mathbf{B} \in \mathbb{R}^{n \times n}$ of $\mathcal{L}$, its GSO $\mathbf{B} = \widetilde{\mathbf{B}} \cdot \mathbf{U}$, a target $\mathbf{x} \in \mathbb{R}^n$
Ensure: $\mathbf{v} \in \mathcal{L} \cap \{\mathbf{x} + \mathcal{P}(\widetilde{\mathbf{B}})\}$
1: $\mathbf{t} \leftarrow \mathbf{B}^{-1} \cdot \mathbf{x}$
2: **for** $i = n, \ldots, 1$ **do**
3: $\quad z_i \leftarrow \left\lfloor t_i + \sum_{j>i}(t_j - z_j)U_{ij} \right\rceil$
4: **end for**
5: **return** $\mathbf{v} = \mathbf{B} \cdot \mathbf{z}$

2.2 Distributions

Let $\mathcal{D}$ be a distribution of density f over $\mathbb{R}^n$. With $\mathcal{L}$ a lattice of $\mathbb{R}^n$, we define the discretization of $\mathcal{D}$ over $\mathcal{L}$, and we write $\mathcal{D}_{\mathcal{L}}$ the distribution of density

$$f_{\mathcal{L}} : \mathbf{x} \in \mathcal{L} \longmapsto \frac{f(\mathbf{x})}{f(\mathcal{L})},$$

where $f(\mathcal{L}) = \sum_{y \in \mathcal{L}} f(\mathbf{y})$. Let X be a real random variable, we write respectively $\mathbb{E}(X)$ and $\mathbb{V}(X)$ respectively the expected value and the variance of X. Both notations extend to vectors by coordinate-wise application. For a subset $\Omega \subset \mathbb{R}^n$, we write its indicator function $\mathbf{1}_{\Omega}$.

Let f be the density of a probability distribution $\mathcal{D}$ over $\mathbb{R}$ that we want to sample from. We define the Inverse Cumulative Density Function (after $\mathrm{ICDF}_{\mathcal{D}}$)

as the reciprocal of the cumulative density function

$$\mathrm{ICDF}_{\mathcal{D}} = \left(x \longmapsto \int_{-\infty}^{x} f(t)\,\mathrm{d}\,t \right)^{-1}.$$

Proposition 1. *If the random variable U has a uniform distribution on $[0,1]$, then the distribution of $\mathrm{ICDF}_{\mathcal{D}}(U)$ is $\mathcal{D}$.*

If evaluating the ICDF of a given distribution $\mathcal{D}$ is possible, one can use Proposition 1 to sample from $\mathcal{D}$.

2.3 Squaremonic Functions

In this subsection, we introduce the second ingredient of our framework: a class of functions that behave nicely when discretized over a lattice.

Definition 2 (Squaremonicity). *Let $\mathcal{T}$ be a prototile of a $\mathcal{L}$-regular tiling. We say that a function $f : \Omega \subset \mathbb{R}^n \longmapsto \mathbb{R}$ is $\mathcal{T}$-squaremonic if*

$$\forall \mathbf{x} \in \mathbb{R}^n \text{ such that } \mathcal{T} + \mathbf{x} \subset \Omega,\ \frac{1}{\mathrm{Vol}(\mathcal{T})} \int_{\mathcal{T}+\mathbf{x}} f = f(\mathbf{x}). \tag{2}$$

We will refer to (2) as the squaremonic equation or the squaremonic property, and $\mathcal{T}$ is called a squaremonic tile of f. In addition, we say that a distribution is squaremonic if its density is squaremonic.

Notice that due to the linearity of the integral, for a given prototile $\mathcal{T}$, the set of $\mathcal{T}$-squaremonic functions is a linear space. We stress that these squaremonic functions are not only a theoretical object. Indeed, constant functions, linear functions (hence affine functions) and affine product functions admit squaremonic tiles. More details are given in Section 4.

The name *square-harmonic* or squaremonic is a portmanteau of *square* and *harmonic*. This name stems from a similarity between these squaremonic functions and harmonic functions. Harmonic functions on an open subset $\Omega \subset \mathbb{R}^n$ are the solutions of the equation $\Delta f = 0$, where $\Delta = \sum \partial_i^2$ is the Laplacian operator. Harmonicity is equivalent to the *Mean Value Property*, that is

$$\forall \mathbf{x} \in \Omega,\ \forall r > 0 \text{ such that } B_2^n(\mathbf{x}, r) \subset \Omega,\ \frac{1}{\mathrm{Vol}(B_2^n(\mathbf{x}, r))} \int_{B_2^n(\mathbf{x},r)} f = f(\mathbf{x}). \tag{3}$$

Informally, the mean value of a harmonic function over a Euclidean ball of center $\mathbf{x}$ and radius r is the value of f in the center $\mathbf{x}$. The property (2) verified by squaremonic functions is similar to (3): the mean value of f over a fundamental domain of a lattice is the value of f in a fixed point of the fundamental domain.

The scalability of the radius of the ball for harmonic functions makes a substantial difference with the mean value property for squaremonic functions. Indeed, the tile over which the mean value is calculated cannot, in general, be stretched out, which seems to provide less rich properties. Nonetheless, this resemblance between harmonic and squaremonic functions extends to the maximum principle, that is, the maximum of a harmonic function over the topologic closure of an open set Ω is the maximum over its boundary $\partial\Omega$. A similar yet weaker property holds for squaremonic functions : the maximum of a squaremonic function over the topologic closure of an open set Ω is the maximum over its thickened boundary $\{\mathbf{x} \in \Omega \mid \mathcal{T} + \mathbf{x} \subset \Omega\}$. To our knowledge, a few other properties of harmonic functions can be translated similarly into a squaremonic equivalent. Harmonic analysis is a vastly studied subject, the interested reader can refer to [27,24,34,5]. A crucial setup for squaremonicity in dimension n is the prototile $\mathcal{H}_n = [0,1]^n$ of the lattice $\mathbb{Z}^n$.

3 Our Framework

In this section we introduce the sampler framework, prove its correctness and give an analysis on how to set the parameters from a theoretical point of view.

3.1 Framework Description

As mentioned in the previous sections, the main idea of the sampler is to discretize a continuous distribution over a chosen lattice $\mathcal{L}$. The sampler needs to be provided with two algorithms : $\mathsf{Sample}\mathcal{D}$ to sample from the continuous distribution $\mathcal{D}$, and an $\mathcal{L}$-regular CVP algorithm $\mathcal{A}$ to discretize the distribution over the lattice. The conditions for the sampler to be correct and its running time are specified in Theorem 1.

Algorithm 3: Squaremonic Sampler

Require: A trapdoor τ of a lattice $\mathcal{L}$, a target $\mathbf{c}$
Ensure: $\mathbf{x}$ sampled from $\mathcal{D}_{\mathcal{L}}$ of support $\Omega_{\mathbf{c}} \subset \mathbb{R}^n$

```
while True do
    y ← SampleD                          {SampleD samples from D}
    x = A(τ, c + y)
    if x − c ∈ Ω then
        Return x
    end if
end while
```

Theorem 1. *Let τ be a trapdoor of a lattice $\mathcal{L}$ and $\mathcal{A} : \mathbf{T} \times \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be a deterministic algorithm with $\tau \in \mathbf{T}$. Let $\mathcal{D}$ be a distribution of density f over some subset $\Omega' \subset \mathbb{R}^n$. Let $\Omega \subset \Omega'$ be a measurable set, and $\mathbf{c} \in \mathbb{R}^n$. Suppose that:*

1. *$\mathcal{D}$ is sampleable in polynomial time.*
2. *$\mathcal{A}$ is an $\mathcal{L}$-regular algorithm inducing a tiling of prototile $\mathcal{T} = \mathcal{T}_{\mathbf{0}}$.*
3. *The density f is $\mathcal{T}$-squaremonic.*
4. *The set Ω is such that $\Omega \subset \{\mathbf{x} \mid \mathbf{x} + \mathcal{T} \subset \Omega'\} \subset \Omega'$. We moreover require that testing that some vector $\mathbf{x}$ is in Ω can be done efficiently.*

Then the output $\mathbf{x}$ of Algorithm 3 is distributed as follows:

$$\mathbf{x} \sim \{(\mathbf{c} + \mathcal{D})\mathbf{1}_{\Omega_{\mathbf{c}}}\}_{\mathcal{L}}, \tag{4}$$

Where $\Omega_{\mathbf{c}} = \Omega + \mathbf{c}$. In addition, the expected number of iterations of the **while** *loop is $\mathcal{D}((\mathcal{L} - \mathbf{c}) \cap \Omega + \mathcal{T})^{-1}$.*

Proof. We prove separately correctness and the number of iterations.

Correctness. First, we note that the output $\mathbf{x}$ of Algorithm 3 is necessarily in $\mathcal{L} \cap \Omega_{\mathbf{c}}$. On one hand, it follows from the $\mathcal{L}$-regularity of $\mathcal{A}$ that $\mathbb{P}[\mathbf{x} = \mathbf{v}] = 0$ for any $\mathbf{v} \notin \mathcal{L}$. On the other hand, $\mathbf{x}$ is rejected at step 4 if and only if $\mathbf{x} - \mathbf{c} \notin \Omega$.

Now, we study the probability that $\mathbf{v} \in \mathcal{L} \cap \Omega_{\mathbf{c}}$ is output. The random variable $\mathbf{y}$ follows the distribution $\mathcal{D}$, hence $\mathbf{c} + \mathbf{y}$ follows the distribution $\mathbf{c} + \mathcal{D}$. At the end of step 3:

$$\begin{aligned}
\mathbb{P}[\mathbf{x} = \mathbf{v}] &= \mathbb{P}_{\mathbf{y} \leftarrow \mathcal{D}}[\mathbf{c} + \mathbf{y} \in \mathcal{T}_{\mathbf{v}}] && (5)\\
&= \int_{\mathcal{T}_{\mathbf{v}}} f(\mathbf{t} - \mathbf{c}) \, \mathrm{d}\, \mathbf{t} && (6)\\
&= \int_{(\mathbf{v}-\mathbf{c})+\mathcal{T}} f(\mathbf{t}) \, \mathrm{d}\, \mathbf{t} && (7)\\
&= \mathrm{Vol}(\mathcal{T}) \cdot f(\mathbf{v} - \mathbf{c}) && (8)
\end{aligned}$$

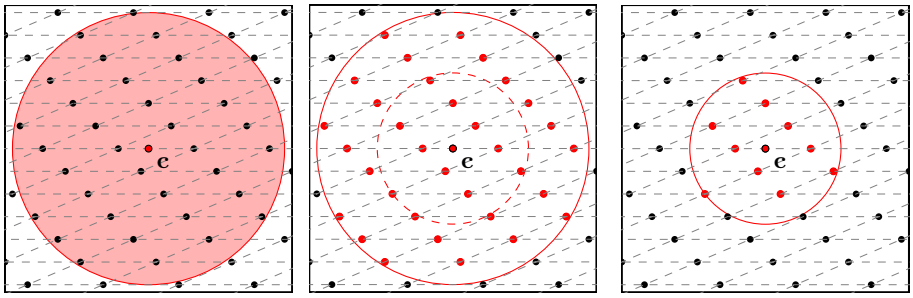
(5) follows from the fact that $\mathbf{v}$ is output if and only if $\mathbf{c} + \mathbf{v}$ is in $\mathcal{T}_{\mathbf{v}}$. Since $\Omega + \mathcal{T} \subseteq \Omega'$ and $\mathbf{v} \in \Omega_{\mathbf{c}}$, it implies that $\mathcal{T}_{\mathbf{v}} \subseteq \Omega'_{\mathbf{c}}$ and (6) is therefore valid. (7) is a simple change of variable (translation by $\mathbf{c}$). Finally, and most crucially, (8) follows from the $\mathcal{T}$-squaremonicity of f. Therefore the distribution of $\mathbf{x}$ is proportional to $f(\mathbf{v} - \mathbf{c})$, and its support is exactly $\mathcal{L} \cap \Omega_{\mathbf{c}}$. The result follows.

Number of iterations. The support of the output of Algorithm 3 is exactly $\mathcal{L} \cap \Omega_{\mathbf{c}}$. Combining this fact with (6), the probability that Algorithm 3 terminates at a given iteration is:

$$P := \sum_{\mathbf{v} \in \mathcal{L} \cap \Omega_{\mathbf{c}}} \int_{\mathcal{T}_{\mathbf{v}}} f(\mathbf{t} - \mathbf{c}) \, \mathrm{d}\, \mathbf{t} = \sum_{\mathbf{v} \in (\mathcal{L} - \mathbf{c}) \cap \Omega} \int_{\mathcal{T}_{\mathbf{v}}} f(\mathbf{t}) \, \mathrm{d}\, \mathbf{t} = \mathcal{D}((\mathcal{L} - \mathbf{c}) \cap \Omega + \mathcal{T}) \tag{9}$$

and the expected number of iterations is $1/P$. □

Figure 2 provides a visual illustration of our framework. A continuous distribution is sampled (Figure 2a) and discretized (Figure 2b) via a regular algorithm (here, the round-off algorithm). Finally, a rejection step (Figure 2c) discards all points outside $\Omega_{\mathbf{c}}$ since these might leak information about the basis used.



(a) $\mathcal{D}$ (b) $\mathcal{D}_{\{\mathcal{L}\cap\Omega'_{\mathbf{c}}\}}$ (c) $\mathcal{D}_{\{\mathcal{L}\cap\Omega_{\mathbf{c}}\}}$

Fig. 2: Visual illustration of Algorithm 3. A continuous distribution is sampled (2a), then discretized (2b), and finally rejection sampling is applied (2c).

Security. The security of our framework is given by the independance between the trapdoor used in the regular algorithm and the output distribution. The security of our framework is therefore immediate from Theorem 1. Indeed, the output distribution of the proposed sampler is *perfectly* indistinguishable from the ideal distribution $\{\mathbf{c}+\mathcal{D}\}_{\mathcal{L}\cap\Omega_{\mathbf{c}}}$. There is therefore no leakage of information as long as Ω is independent from the trapdoor. In the instantiations we propose, this is indeed the case. This differs from classical samplers, which are only *statistically* indistinguishable from ideal distributions.

Using our sampler in a specific cryptographic scheme may mandate additional context-specific requirements. For example, using it in a signature scheme *à la* [21] requires the distribution to satisfy something analog to their leftover hash lemma [21, Lemma 5.1], but this will be verified easily for the (currently) large distributions that our framework samples from.

3.2 Rejection Rate and Parameter Analysis

The acceptance rate P is the weight of the discretization $\mathcal{D}_{\mathcal{L}}$ over $\Omega + \mathbf{c}$, which by squaremonicity of f is given by $P = \mathcal{D}(\mathcal{L} \cap (\Omega + \mathbf{c}) + \mathcal{T})$. While the concrete value of P seems hard to calculate, we have two strategies to give an estimate. The first strategy is to lower bound the probability P by considering the set

$$\Omega'' = \bigcup_{\substack{A\subset\Omega' \\ A-\mathcal{T}\subset\Omega}} A.$$

Indeed, it follows from the definition that $\mathbf{y} \in \Omega''$ implies $\mathbf{x} \in \Omega + \mathbf{c}$, hence the sample is accepted, therefore we have $P \geq P'' := \mathcal{D}(\Omega'')$. Using this conservative bound in our implementation to choose the parameters of the distribution underestimates the actual empirical acceptance rate by a constant factor. The second strategy is to use the so-called Gaussian heuristic, that is a prediction of the number of lattice points in a measurable convex body of $\mathbb{R}^n$.

Heuristic 1 (Gaussian Heuristic). *Let $\mathcal{L}$ be a full rank lattice of $\mathbb{R}^n$. For a convex body $K \subset \mathbb{R}^n$, we have*

$$|\mathcal{L} \cap K| \simeq \frac{\mathrm{Vol}(K)}{\mathrm{Vol}(\mathcal{L})}.$$

Under Heuristic 1, we have $|\mathcal{L} \cap \Omega| = \mathrm{Vol}(\Omega)/\mathrm{Vol}(\mathcal{L})$. With, for example, f constant (corresponding to the uniform distribution) over Ω', we have $P = \mathrm{Vol}(\Omega)/\mathrm{Vol}(\mathcal{L}) \cdot \mathrm{Vol}(\mathcal{L})/\mathrm{Vol}(\Omega') = \mathrm{Vol}(\Omega)/\mathrm{Vol}(\Omega')$. According to our own experiments, this estimate is very accurate for constant distributions. For other distributions, we extrapolate the previous formula to

$$P = \mathcal{D}(\Omega).$$

While this estimate is unlikely to be accurate[3], it matches with our experiments on uniform and affine distributions for reasonable (constant) acceptance rates.

Heuristic 2. *With notations from Theorem 1, the probability $P = \mathcal{D}((\mathcal{L} - c) \cap \Omega + \mathcal{T})$ is $\mathcal{D}(\Omega)$.*

We note that as we narrow the support $\Omega_{\mathbf{c}}$ of the final distribution, the acceptance rate of our algorithm will become increasingly lower. In practice, lowering the standard deviations given in Table 1 even by a constant factor can result in an huge blow-up in the running time.

4 Instantiations of our Framework

In this section, we instantiate the framework described in Section 3 with various examples of distributions and the RoundOff algorithm. The first subsection presents properties of squaremonic functions. We give a partial study of their properties and examples.

Instantiating our framework implies setting four components: a continuous support Ω', a probability distribution $\mathcal{D}$ over Ω', an $\mathcal{L}$-regular algorithm of prototile $\mathcal{T}$ and a set Ω[4] such that the support of the sampled distribution is $\mathcal{L} \cap \Omega$. We only consider the RoundOff algorithm for explicit examples and explain how to extend the results to both the NearestPlane algorithm, and the ExactCVP when it is possible.

Table 1 sums up the standard deviation (up to a constant factor that we omit) of the distribution that can be sampled with our framework for a constant acceptance rate (independent of the dimension n). A ✗ mark means that the distribution is in general not squaremonic over the tile of the regular algorithm.

[3] The difference with uniform is that in general the integrals of f over $\Omega \backslash (\mathcal{L} \cap \Omega + \mathcal{T})$ and $(\mathcal{L} \cap \Omega + \mathcal{T}) \cap (\Omega' \backslash \Omega)$ do not compensate each other.

[4] Remind that testing that a vector is in the set Ω has to be easy. In the examples we develop, such a test reduces to computing an ℓ_p norm for p in $\{1, 2, \infty\}$, and an inequality.

Table 1: Standard deviation achieved in constant acceptance rate for the regular algorithm/distribution couple by Algorithm 3.

	ℓ_∞ Uniform	ℓ_2 Uniform	Affine	Affine product
RoundOff	$n^{1.5}s_1(\mathbf{B})$	$ns_1(\mathbf{B})$	$n^{1.5}s_1(\mathbf{B})$	$n^{1.5}s_1(\mathbf{B})$
NearestPlane	$n^{1.5}s_1(\widetilde{\mathbf{B}})$	$ns_1(\widetilde{\mathbf{B}})$	$n^{1.5}s_1(\widetilde{\mathbf{B}})$	$n^{1.5}s_1(\widetilde{\mathbf{B}})$
ExactCVP	$n^{1.5}\rho_\infty(\mathcal{L})$	$n^{1.5}\rho_2(\mathcal{L})$	$n^{1.5}\rho_1(\mathcal{L})$	✗

4.1 Mean Value Property over Regular Tilings

In this section, we provide two interpretations of squaremonicity which may be more intuitive than the arguably abstract Definition 2. We also explain how in many cases, we can reduce the study of a $\mathcal{T}$-squaremonic function (for an arbitrary tile $\mathcal{T}$), to studying a $\mathcal{H}_n$-squaremonic function (for the hypercube $\mathcal{H}_n = [0,1]^n$), then to studying n $[0,1]$-squaremonic functions. This simplifies many subsequent proofs and arguments.

Intuitive interpretations. There are at least two geometric interpretations of the notion of squaremonic distributions. The first one is as follows. Consider a distribution $\mathcal{D}$ and a $\mathcal{L}$-regular tiling $\mathcal{L} + \mathcal{T}$ of the space, of prototile $\mathcal{T}$. We define $\mathcal{D}_1$ as the discretization of $\mathcal{D}$ over $\mathcal{L}$. In addition, we define $\mathcal{D}_2$ as a $\mathcal{T}$-rounding of $\mathcal{D}$ over $\mathcal{L}$; more precisely, given a point $\mathbf{v} \in \mathcal{L}$, we set $\mathcal{D}_2(\mathbf{v})$ as the integral of the density function of $\mathcal{D}$ over the compact set $\mathbf{v} + \mathcal{T}$. Saying that $\mathcal{D}$ is $\mathcal{T}$-squaremonic implies that $\mathcal{D}_1$ and $\mathcal{D}_2$ are the same distribution. Note that when $\mathcal{D}$ is a continuous Gaussian, $\mathcal{D}_1$ and $\mathcal{D}_2$ are what is commonly called a *discrete Gaussian* and *rounded Gaussian*, respectively. For Gaussians, $\mathcal{D}_1$ and $\mathcal{D}_2$ are in general *not* equal.

For an interpretation more oriented towards mathematical analysis, consider a continuous function $f : \mathbb{R} \longrightarrow \mathbb{R}$. For all $x \in \mathbb{R}$, via the intermediate value theorem, there exists a number $0 < a < 1$ such that

$$\int_x^{x+1} f(t)\,\mathrm{d}\,t = f(x+a).$$

If f is monotonic, then $a(x)$ is unique; we then abusively note $a : x \mapsto a(x)$ the continuous function mapping x to $a(x)$. Here, saying that f is squaremonic with respect to the tile $\mathcal{T} = [0,1)$ is equivalent to saying that a is constant.

Separable squaremonic functions. In this paragraph, we split the set of squaremonic functions in two: those that satisfy a $\mathrm{GL}_n(\mathbb{R})$ stability condition,

and those that have separated variables (both sets are not disjoint). This simplifies proofs and makes arguments clearer.

We assume for this discussion that the regular algorithm we use is $\mathsf{RoundOff} \circ t_{\mathbf{Ba}}$ with $t_{\mathbf{Ba}}$ the translation by $\mathbf{Ba}$, so $\mathcal{T} = \mathcal{P}(\mathbf{M}) - \mathbf{Ma}$ for some matrix $\mathbf{M}$ and some vector $\mathbf{a} \in \mathcal{H}_n$. Let $\mathcal{C}$ be a set of functions such that $\forall f \in \mathcal{C},\ \forall \mathbf{M} \in \mathrm{GL}_n(\mathbb{R}),\ f \circ \mathbf{M} \in \mathcal{C}$. All the examples we develop in this paper (namely constant, affine and affine product) share this $\mathrm{GL}_n(\mathbb{R})$ stability property. Let f be a function from such a set $\mathcal{C}$. The study of the squaremonic equation of f relatively to $\mathcal{P}(\mathbf{M}) - \mathbf{Ba}$ reduces via a substitution to the study of $f \circ \mathbf{M}^{-1}$'s squaremonicity over a translation of $\mathcal{H}_n$, the canonical tile. This fact suggests that for such classes of functions, we study the squaremonicity in the canonical setup first, and extend the result to any basis of any lattice.

Let us consider a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, together with the translated $\mathsf{RoundOff}$ tile $\mathcal{P}(\mathbf{B}) - \mathbf{Ba}$. As argued in the previous paragraph, we first study the squaremonicity of some function f over the canonical tile $\mathcal{H}_n$. In this paragraph, we study squaremonic functions with separated variables, that is squaremonic functions f such that $f(\mathbf{x}) = \prod_{i=1}^{n} f_i(x_i)$ (which correspond to all our instantiations, except affine and ℓ_2 uniform). Then, the squaremonicity of f relatively to $\mathcal{H}_n - \mathbf{a}$ is equivalent to $\forall\ 1 \leq i \leq n,\ f_i$ is squaremonic relatively to $\mathcal{H}_1 - a_i$. This fact comes from the factorization of the squaremonic integral into 1-dimensional integrals. Assume, all f_i's are squaremonic relatively to $\mathcal{H}_1 - a_i$, then

$$\begin{aligned}
\int_{\mathcal{H}_n - \mathbf{a} + \mathbf{x}} f(\mathbf{t}) \,\mathrm{d}\,\mathbf{t} &= \int_{\mathcal{H}_n - \mathbf{a} + \mathbf{x}} \prod_{i=1}^{n} f_i(t_i) \,\mathrm{d}\, t_i \\
&= \prod_{i=1}^{n} \int_{[-a_i, 1-a_i] + x_i} f(t_i) \,\mathrm{d}\, t_i \\
&= \prod_{i=1}^{n} f_i(x_i) \\
&= f(\mathbf{x}),
\end{aligned}$$

and f is squaremonic relatively to the canonical tile (with an offset parameter $\mathbf{a}$ for the unit hypercube), and vice-versa. Constant distributions (over a hypercube) and affine product distributions (functions of the type $\mathbf{x} \longmapsto \prod_i (a_i x_i + b_i)$, for some vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$) share this property that their variables are separated, therefore their n-dimensional squaremonicity reduces to n times 1-dimensional squaremonicity. Moreover, sampling from a separated variables density can be done coordinate-wise in parallel (as each coordinate is independent from the others), which is algorithmic-wise convenient.

4.2 Uniform Distributions with the RoundOff Algorithm

Let Ω be a compact subset of $\mathbb{R}^n$. We define the (continuous) uniform distribution over Ω as the distribution of density proportional to $\mathbf{1}_\Omega$, and write it

$\mathcal{U}(\Omega)$. We will consider two different shapes for Ω : the ℓ_2 balls, to optimize the shortness of the output and the ℓ_∞ balls to optimize the speed and simplicity of the sampler. Because constant functions are squaremonic for any prototile, any regular algorithm would work.

Uniform distribution over a hypercube. The first example of uniform distribution over a lattice we give is the uniform distribution over a hypercube. The reason for this choice is that over a hypercube, the coordinates of the random vector are independent, which makes the continuous sampling very easy.

Proposition 2 (ℓ_∞ Uniform with RoundOff instantiation). *Let $\mathbf{B}$ be a basis of a lattice $\mathcal{L}$ and $\mathbf{c} \in \mathbb{R}^n$. The instantiation of Algorithm 3 with*

1. $\Omega' = [-nM_\infty(\mathbf{B}), nM_\infty(\mathbf{B})]^n$
2. $\Omega = [-(n-1)M_\infty(\mathbf{B}), (n-1)M_\infty(\mathbf{B})]^n$
3. $\mathcal{D} = \mathcal{U}(\Omega')$
4. $\mathcal{A} = \mathsf{RoundOff}(\mathbf{B}, \cdot)$

satisfies the requirements of Theorem 1 and its acceptance rate is heuristically and asymptotically $P \longrightarrow 1/e$. In other words, the uniform distribution over the hypercube of radius $(n-1)M_\infty(\mathbf{B})$ and center $\mathbf{c}$ is sampleable in polynomial time using Algorithm 3.

Proof. Correctness. We check the requirements of Theorem 1. For Item 1, sampling from $\mathcal{D}$ is done easily by sampling uniformly over $[0,1]^n$ and applying an affine transformation. For Item 2, the RoundOff algorithm is indeed $\mathcal{L}$-regular of prototile $\mathcal{P}(\mathbf{B})$. For Item 3, the density $\mathbf{1}_{\Omega'}$ is trivially $\mathcal{P}(\mathbf{B})$-squaremonic. Finally, for Item 4, by triangle inequality, if $\mathbf{y} = \mathbf{x} + \mathbf{t} \in \Omega + \mathcal{P}(\mathbf{B})$, $\|\mathbf{y}\|_\infty \leq (n-1)M_\infty(\mathbf{B}) + M_\infty(\mathbf{B})$, so $\Omega + \mathcal{P}(\mathbf{B}) \subset \Omega'$, and the instantiation is correct.

Expected running time and radius. Under Heuristic 1, the probability of a sample to be accepted is given by

$$P \simeq \frac{\mathrm{Vol}(\Omega)}{\mathrm{Vol}\,\Omega'} = \left(\frac{(n-1)M_\infty(\mathbf{B})}{nM_\infty(\mathbf{B})}\right)^n = \left(1 - \frac{1}{n}\right)^n.$$

□

Proposition 3. *Let X be a random variable following the discrete uniform distribution $\mathcal{U}([-R,R]^n)_\mathcal{L}$. Then, the expected ℓ_2 norm of X is bounded by*

$$\|X\|_2 \leq \frac{n^2}{2} s_1(\mathbf{B}).$$

Proof. With notations from Proposition 2 and using inequalities between ℓ_2 and ℓ_∞ norms, we have

$$\|X\|_2 \leq \sqrt{n}\|X\|_\infty \leq n^{1.5} M_\infty/2 \leq \frac{n^2}{2} s_1(\mathbf{B})$$

This completes the proof. □

Uniform distribution over an ℓ_2 hyperball. In this paragraph we give a second example of uniform distribution, this time over a ℓ_2 ball of dimension n. Although sampling uniformly random from this set seems more complicated than a hypercube, the ℓ_2 norm of the output random variable with parameter associated to a constant acceptance rate is in average $\sqrt{n}$ lower in the hyperball than in the hypercube (Propositions 3 and 5). As in the previous example, we chose the usual RoundOff algorithm, but any regular CVP algorithm would work.

Proposition 4 (ℓ_2 Uniform with RoundOff instantiation). *The instantiation of Algorithm 3 with*

1. $\Omega' = \mathcal{B}_2^n(nM_2(\mathbf{B}))$
2. $\Omega = \mathcal{B}_2^n((n-1)M_2(\mathbf{B}))$
3. $\mathcal{D} = \mathcal{U}(\Omega')$
4. $\mathcal{A} = \mathsf{RoundOff}(\mathbf{B}, \cdot)$

satisfies the requirements of Theorem 1 and its rejection rate is heuristically and asymptotically $P \longrightarrow 1/e$. In other words, for any center $\mathbf{c} \in \mathbb{R}^n$, the distribution $\mathcal{U}_{\mathcal{L},\mathbf{c}}(nM_2(\mathbf{B}))$ is sampleable in polynomial time.

Proof. Correctness. We check the requirements of Theorem 1. For Item 1, there are several polynomial-time algorithms that sample from the continuous uniform distribution over ℓ_2 balls, for example an ICDF-like algorithm using the identity $\mathcal{U}([0,1])^{1/n} \cdot \mathcal{U}(\mathcal{S}_{n-1}) \sim \mathcal{U}(\mathcal{B}_2^n(1))$ (sampling from the unit sphere $\mathcal{S}_{n-1}$ can be done using results from [15,33], or normalizing spherical Gaussian distributions), using the algorithm from [8], etc. For Item 2, the RoundOff is $\mathcal{L}$ regular for any basis and has prototile $\mathcal{T} = \mathcal{P}(\mathbf{B})$. For Item 3, the distribution $\mathcal{D}$ has density $\mathbf{1}_{\Omega'}$ which is trivially squaremonic over $\mathcal{P}(\mathbf{B})$. Finally, for Item 4, with $M_2(\mathbf{B}) = \max\{\|\mathbf{x}\| \mid \mathbf{x} \in \mathcal{P}(\mathbf{B})\}$, we have $\Omega' \subset \Omega + \mathcal{T}$ via the triangular inequality of $\|.\|_2$.

Expected running time and radius. Under Heuristic 1, the probability of a sample to be accepted is given by

$$P \simeq \frac{\mathrm{Vol}(\Omega)}{\mathrm{Vol}(\Omega')}.$$

The volume of a n-hyperball is homogeneous to the n-th power of its radius. Otherly said, $P \simeq ((n-1)M_2(\mathbf{B})/(nM_2(\mathbf{B})))^n$, which completes the proof. □

Proposition 5. *Let $\mathcal{L}$ be a lattice of basis $\mathbf{B}$. Let X be the output of the instantiation of Proposition 4 (X follows $\mathcal{U}((n-1)M_2(\mathbf{B}))_{\mathcal{L}}$). Then we have*

$$\|X\|_2 \leq \frac{n^{1.5}}{2} s_1(\mathbf{B}).$$

Proof. We have $X \in \mathcal{B}_2^n((n-1)M_2)$, therefore $\|X\|_2 \leq (n-1)M_2$. Moreover, we have $M_2 \leq \frac{\sqrt{n}}{2} s_1(\mathbf{B})$, which completes the proof. □

In both uniform examples above, using the NearestPlane (respectively an ExactCVP) instead of the RoundOff is also valid and yields similar results, substituting $\mathcal{P}(\mathbf{B})$ by $\mathcal{P}(\widetilde{\mathbf{B}})$ (respectively the Voronoi cell of the lattice), and $M_p(\mathbf{B})$ by $M_p(\widetilde{\mathbf{B}})$ (respectively $\rho_p(\mathcal{L})$, the *covering radius* of the lattice, relatively to the ℓ_p norm).

4.3 Affine Distributions with the RoundOff Algorithm

Let $R > 0$ and $\mathcal{B}^n_{1+}(R) = \mathcal{B}^n_1(R) \cap \mathbb{R}^n_+$. We define the affine distribution $\mathcal{A}_n(R, R')$ of parameters R and $R' \geq R$ over $\mathcal{B}^n_{1+}(R)$ as the distribution of density $\mathbf{x} \longmapsto (R' - \|\mathbf{x}\|_1)\, \mathbf{1}_{\mathcal{B}^n_{1+}(R)}$. We define numbers $m^i_1(\mathbf{B}) = \max\{|x_i| \mid \mathbf{x} \in \mathcal{P}(\mathbf{B})\}$, and the point $\mathbf{m}_1(\mathbf{B})$ which coordinates are the $m^i_1(\mathbf{B})$'s.

Proposition 6 (Affine distribution with RoundOff instantiation). *Let* $\mathbf{B}$ *be a basis of a lattice* $\mathcal{L}$ *and* $\mathbf{c} \in \mathbb{R}^n$*. The instantiation of Algorithm 3 with*

1. $\Omega' = \mathcal{B}^n_{1+}(R') - \mathbf{m}_1(\mathbf{B})$, *with* $R' = (n+1)(M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1)$
2. $\Omega = \mathcal{B}^n_{1+}(R)$, *with* $R = n(M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1)$
3. $\mathcal{D} = \mathcal{A}_n(R', R') - \mathbf{m}_1(\mathbf{B})$
4. $\mathcal{A} = \mathsf{RoundOff}(\mathbf{B}, \cdot)$

is correct and its acceptance rate is heuristically and asymptotically $P \longrightarrow 1/e$*. In other words, the distribution* $\mathcal{A}(R, R')_{\mathcal{L},\mathbf{c}}$ *is sampleable in polynomial time.*

Proof. Correctness. We check the requirements of Theorem 1. For Item 1, Algorithm 4 is an example of such polynomial time continuous sampler, which correctness is stated by Lemma 1. For Item 2, the RoundOff algorithm is $\mathcal{L}$-regular, with prototile $\mathcal{P}(\mathbf{B})$. For Item 3, we want to prove that $f : \mathbf{x} \longmapsto R - \|\mathbf{x}\|_1$ is squaremonic for $\mathcal{P}(\mathbf{B})$. Notice that f is affine, that is $f = R + u$, with R constant and $u : \mathbb{R}^n \longrightarrow \mathbb{R}$ linear. With $\mathcal{H}^-_n = \{\mathbf{x} \in \mathcal{H}_n - \vec{1/2} \;/\; x_1 \leq 0\}$ and $\mathcal{H}^+_n = -\mathcal{H}^-_n$, we have the following :

$$\int_{\mathcal{P}(\mathbf{B})+\mathbf{x}} f(\mathbf{y})\,\mathrm{d}\,\mathbf{y} = \int_{\mathcal{P}(\mathbf{B})+\mathbf{x}} (R + u(\mathbf{y}))\,\mathrm{d}\,\mathbf{y} \tag{10}$$

$$= \int_{\mathcal{P}(\mathbf{B})} (R + u(\mathbf{z}) + u(\mathbf{x}))\,\mathrm{d}\,\mathbf{z} \tag{11}$$

$$= \mathrm{Vol}(\mathcal{P}(\mathbf{B}))f(\mathbf{x}) + \int_{\mathcal{P}(\mathbf{B})} u(\mathbf{z})\,\mathrm{d}\,\mathbf{z} \tag{12}$$

$$= \det \mathcal{L} f(\mathbf{x}) + \det \mathcal{L} \left(\int_{\mathcal{H}^-_n} u + \int_{\mathcal{H}^+_n} u \right) \tag{13}$$

$$= \det \mathcal{L} f(\mathbf{x}) \tag{14}$$

where (11) comes from the substitution $\mathbf{y} = \mathbf{z} + \mathbf{x}$, (12) comes from the linearity of the integral, (13) comes from the substitution $\mathbf{B}\mathbf{w} = \mathbf{z}$ and splitting the integral over $\mathcal{H}_n$ into the positive and negative part, and (14) comes from the fact that as u is linear, it is odd and the two integrals cancel each other. Finally, for Item 4, by the triangular inequality, if $\mathbf{x} = \mathbf{y} + \mathbf{t} \in \Omega + \mathcal{P}(\mathbf{B})$, then $\mathbf{x} + \mathbf{m}_1(\mathbf{B}) \in \mathbb{R}^n_+$ and $\|\mathbf{x}\|_1 \leq nM_1(\mathbf{B}) + M_1(\mathbf{B})$, so $\Omega + \mathcal{T} \subset \Omega'$.

Expected running time and radius According to Heuristic 2, the acceptance rate P is given by

$$P = \int_{\Omega} f.$$

Let R be the radius of Ω. First, f is proportional to $\mathbf{x} \longmapsto R + M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1 - \|\mathbf{x} + \mathbf{m}_1(\mathbf{B})\|_1$. The graph of the latter function describes the $n+1$ dimensional ℓ_1 ball over the all-positive quadrant $\mathbb{R}^n_+$ of $\mathbb{R}^n$, so the normalization factor is $1/2^n$ times the volume of the ℓ_1 ball of dimension $n+1$ and radius $R + M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1$. In the end,

$$f(\mathbf{x}) = \frac{(n+1)!}{\left(R + M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1\right)^{n+1}} \cdot \left(R + M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1 - \|\mathbf{x}\|_1\right).$$

Now, we calculate the acceptance rate, writing M_1 instead of $M_1(\mathbf{B})$ and $\mathbf{m}_1$ instead of $\mathbf{m}_1(\mathbf{B})$ to save space.

$$\begin{aligned}
P &= \int_{\mathcal{B}^n_{1+}(R)} f(\mathbf{t})\,\mathrm{d}\,\mathbf{t} && (15)\\
&= \frac{(n+1)!}{\left(R + M_1 + \|\mathbf{m}_1\|_1\right)^{n+1}} \int_{\mathcal{B}^n_{1+}(R)} \left(R + M_1 + \|\mathbf{m}_1\|_1 - \|\mathbf{x} + \mathbf{m}_1\|_1\right)\mathrm{d}\,\mathbf{x} && (16)\\
&= \frac{(n+1)!}{\left(R + M_1 + \|\mathbf{m}_1\|_1\right)^{n+1}} \cdot \int_{\mathcal{B}^n_{1+}(R)} \left(R + M_1 - \|\mathbf{x}\|_1\right)\mathrm{d}\,\mathbf{x} && (17)\\
&= \frac{(n+1)!}{\left(R + M_1 + \|\mathbf{m}_1\|_1\right)^{n+1}} \left(\mathrm{Vol}(\mathcal{B}^{n+1}_{1+}(R)) + M_1\,\mathrm{Vol}(\mathcal{B}^n_{1+}(R))\right) && (18)\\
&= \left(1 - \frac{M_1 + \|\mathbf{m}_1\|_1}{R + M_1 + \|\mathbf{m}_1\|_1}\right)^{n+1} + \frac{(n+1)M_1}{R}\left(1 - \frac{M_1 + \|\mathbf{m}_1\|_1}{R + M_1 + \|\mathbf{m}_1\|_1}\right)^{n+1} && (19)\\
&= \left(1 - \frac{M_1 + \|\mathbf{m}_1\|_1}{R + M_1 + \|\mathbf{m}_1\|_1}\right)^{n+1}\left(1 + \frac{(n+1)M_1}{R}\right), && (20)
\end{aligned}$$

where (15) is the estimate Section 3.2, (17) follows from the fact that $\|\mathbf{x} - \mathbf{m}_1\|_1 = \|\mathbf{x}\|_1 + \|\mathbf{m}_1\|_1$, (18) follows from the linearity of the integral and from the fact that the graph of $\mathbf{x} \longmapsto R - \|\mathbf{x}\|_1$ over $\mathcal{B}^n_{1+}(R)$ describes the set $\mathcal{B}^{n+1}_{1+}(R) \cap \mathbb{R}^n_+$, and (19) follows from the fact that $\mathrm{Vol}(\mathcal{B}^n_{1+}(r)) = \frac{r^n}{(n+1)!}$. Finally, one can check that if the radius of Ω verifies $R = n(M_1 + \|\mathbf{m}_1\|_1)$, then P converges to $1/e$. □

Lemma 1. *There is a polynomial time algorithm, that on input a dimension n and a radius R' outputs a sample from $\mathcal{A}_n(R', R')$.*

Proof. Algorithm 4 is such an algorithm, its proof of correctness is deferred to Appendix A. □

Algorithm 4: Continuous affine sampler

Require: Dimension n, radius R'
Ensure: A sample from $\mathcal{A}_n(R', R')$
$\mathbf{x} = \vec{0}$
$\mathbf{u} \leftarrow^{\$} \mathcal{H}_n$ {Here, $\leftarrow^{\$}$ means sampled uniformly at random}
for $i = n, \dots, 1$ **do**
  $x_i = \left(R' - \sum_{j=i+1}^{n} x_j\right)\left(1 - (1-u_i)^{1/(i+1)}\right)$
end for
return $\mathbf{x}$

Proposition 7. *Let $n \in \mathbb{N}$. Let X be a random variable following the distribution $\mathcal{A}_n(R, R')$, $R = n\left(M_1(\mathbf{B}) + \|\mathbf{m}_1(\mathbf{B})\|_1\right)$. Then, we have*

$$\mathbb{E}(\|X\|_2) \leq 2n^2 s_1(\mathbf{B}).$$

Proof. Let Y be a random variable following $\mathcal{A}(R', R')$. We have $\mathbb{E}(\|X\|_2) \leq \mathbb{E}(\|Y\|_2)$, and we use Jensen's inequality :

$$\mathbb{E}(\|X\|_2) \leq \sqrt{\mathbb{E}(\|Y\|_2^2)}.$$

One can check that the variance of Y_1 is asymptotically equal to $\left(\frac{R'}{n}\right)^2$.

$$\begin{aligned}\mathbb{E}(\|Y\|_2^2) &= n\mathbb{E}(Y_1^2) \\ &\sim n(M_1 + \|\mathbf{m}_1\|_1)^2\end{aligned}$$

Now, notice that $M_1(\mathbf{B}) \leq \|\mathbf{m}_1\|_1 \leq n^{1.5} s_1(\mathbf{B})$, which completes the proof. □

Proposition 7 bounds the expected ℓ_2 norm of a random variable following the continuous affine distribution. While the instantiation of Algorithm 3 would sample from the discretization of the latter distribution, we expect the discrete distribution to have similar moments as the continuous one. This similarity can be quantified using Riemann-sum-like arguments. Using the NearestPlane (respectively an ExactCVP) instead of the RoundOff is also valid (as long as the prototile of the algorithm is symmetrical, which is the case for the NearestPlane and the ExactCVP) and yields similar results, substituting $\mathcal{P}(\mathbf{B})$ by $\mathcal{P}(\widetilde{\mathbf{B}})$ (respectively the Voronoi cell of the lattice).

5 Open Problems

Better Efficiency. The main drawback of our framework is that our instantiations suffer large standard deviations. The $O(n^{1.5})$ overhead factor roughly comes from two different problems. The first one is that we measure the size of the output of our algorithms with the ℓ_2 norm, but the distributions sometimes have shapes more amenable to the ℓ_1 or ℓ_∞ norm. We usually lose a $\sqrt{n}$ factor due to the norm inequalities, but measuring the ℓ_∞ norm of the output, for example, can be relevant in cryptography.[5]

The second reason is that, informally, when the support of the distribution is an ℓ_p ball, the radius of the ball increases as the standard deviation increases, but its volume increases as its radius to the power n. The acceptance rates of the distributions defined over ℓ_p balls of radius r have the following form:

$$\left(1 - \frac{M_p(\mathbf{B})}{r + M_p(\mathbf{B})}\right)^n$$

and we lose a factor $O(n)$ by setting $r = nM_p(\mathbf{B})$.

While this seems to prevent the framework from being practically efficient, there are several ways to improve its performance. First, it seems that by being more permissive on the rejection sampling step in our sampler framework, one can find a precision/size trade-off, trading perfect indistinguishability for statistical indistinguishability. As mentioned in the introduction, the idea of using a regular algorithm to round a continuous distribution was, to our knowledge, only attempted on Gaussian distributions, yielding a very large standard deviation to compensate the lack of squaremonicity of the Gaussian density function. We leave for future work to study the behaviour of the standard deviation when the density function is only "ϵ-$\mathcal{T}$-squaremonic". In addition, our proofs use inequalities on the quality of the basis in a *worse-case* scenario. In a cryptographic context, it is likely that we will obtain outputs with shorter norms.

More Instantiations. There are likely more squaremonic functions than the ones we exhibited. Harmonic functions are at the core of a vastly studied domain in mathematics, and we believe that squaremonic functions may enjoy similarly rich properties. We tried – unsuccessfully – to find a partial differential equation equivalent to the squaremonic mean value property, which eventually may lead to finding more squaremonic functions. The more squaremonic functions we find, the more sampleable distributions we have, with potentially improved instantiations.

Precision of Floating-Point Arithmetic. We make an extensive use of continuous distributions in this work. This raises the question of the necessary precision for floating-point arithmetic operations. Solving this question will be key to efficiently and securely instantiating our algorithms.

[5] For example, although it does not use trapdoor sampling, the signature scheme Dilithium [17] relies for its security on the MSIS problem with the ℓ_∞ norm.

Secure Implementation. Finally, our algorithms may require to sample from non-uniform distributions. This sampling should be performed in a manner that does not leak side-channel information (e.g. timing). Indeed, several attacks [19] have been mounted against implementations of lattice-based signature schemes that leaked such information. In addition, a secure implementation would need to ensure that the acceptance probability does not depend of the private key, in order to prevent timing attacks in the line of [20].

References

1. Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete Gaussian sampling: Extended abstract. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 733–742. ACM Press, June 2015.
2. Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2^n time - the discrete Gaussian strikes again! In Venkatesan Guruswami, editor, *56th FOCS*, pages 563–582. IEEE Computer Society Press, October 2015.
3. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [22], pages 553–572.
4. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In Rabin [30], pages 98–115.
5. Thomas Alazard. Analyse et équations aux dérivées partielles. 2017.
6. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.
7. László Babai. On lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
8. Franck Barthe, Olivier Guédon, Shahar Mendelson, Assaf Naor, et al. A probabilistic approach to the geometry of the pn-ball. *The Annals of Probability*, 33(2):480–513, 2005.
9. Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1006–1018. ACM Press, October 2016.
10. Xavier Boyen. Attribute-based functional encryption on lattices. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 122–142. Springer, Heidelberg, March 2013.
11. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.
12. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Gilbert [22], pages 523–552.
13. Yilei Chen, Nicholas Genise, and Pratyay Mukherjee. Approximate trapdoors for lattices and smaller hash-and-sign signatures. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 3–32. Springer, Heidelberg, December 2019.

14. John Horton Conway and Neil JA Sloane. Low-dimensional lattices. vi. voronoi reduction of three-dimensional lattices. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 436(1896):55–68, 1992.
15. Craig Cumbus. Uniform sampling in the hypersphere via latent variables and the gibbs sampler. 1996.
16. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018. https://tches.iacr.org/index.php/TCHES/article/view/839.
17. Léo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals–dilithium: Digital signatures from module lattices. 2018.
18. Léo Ducas and Phong Q. Nguyen. Faster Gaussian lattice sampling using lazy floating-point arithmetic. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 415–432. Springer, Heidelberg, December 2012.
19. Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 1857–1874. ACM Press, October / November 2017.
20. Pierre-Alain Fouque, Paul Kirchner, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Key recovery from Gram-Schmidt norm leakage in hash-and-sign signatures over NTRU lattices. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 34–63. Springer, Heidelberg, May 2020.
21. 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. ACM Press, May 2008.
22. Henri Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Heidelberg, May / June 2010.
23. Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th SODA*, pages 937–941. ACM-SIAM, January 2000.
24. E.H. Lieb, M. Loss, M.A. LOSS, and American Mathematical Society. *Analysis*. Crm Proceedings & Lecture Notes. American Mathematical Society, 2001.
25. Vadim Lyubashevsky and Daniel Wichs. Simple lattice trapdoor sampling from a broad class of distributions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 716–730. Springer, Heidelberg, March / April 2015.
26. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.
27. Camil Muscalu and Wilhelm Schlag. *Harmonic functions; Poisson kernel*, volume 1 of *Cambridge Studies in Advanced Mathematics*, page 28–51. Cambridge University Press, 2013.
28. Chris Peikert. An efficient and parallel Gaussian sampler for lattices. In Rabin [30], pages 80–97.
29. Thomas Prest. Sharper bounds in lattice-based cryptography using the Rényi divergence. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 347–374. Springer, Heidelberg, December 2017.

30. Tal Rabin, editor. *CRYPTO 2010*, volume 6223 of *LNCS*. Springer, Heidelberg, August 2010.
31. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
32. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
33. Min-Zhi Shao and Norman Badler. Spherical sampling by archimedes' theorem. *Technical Reports (CIS)*, page 184, 1996.
34. E.M. Stein, T.S. Murphy, and Princeton University Press. *Harmonic Analysis: Real-variable Methods, Orthogonality, and Oscillatory Integrals*. Monographs in harmonic analysis. Princeton University Press, 1993.
35. Georges Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287, 1908.

A Appendix

Proof. [Proof of Lemma 1] Let X be a random variable following $\mathcal{A}_n(R,R)$. The variable X has a density, which is given by $f : \mathbf{x} \longmapsto \frac{1}{\mathrm{Vol}(\mathcal{B}_1^{n+1}(R))}(R - \|\mathbf{x}\|_1)$. We write $A = \frac{1}{\mathrm{Vol}(\mathcal{B}_1^{n+1})}$, and calculate the density g of X_n.

$$
\begin{aligned}
g(x_n) &= A \int_{\mathbb{R}^{n-1}} f(t_1, \dots, t_{n-1}, x_n)\, \mathrm{d}\, t_1 \dots \mathrm{d}\, t_{n-1} \\
&= A \int_{\mathbb{R}^{n-1}} \left(R - x_n - \sum_{i=1}^{n-1} t_i \right) \mathrm{d}\, t_1 \dots \mathrm{d}\, t_{n-1}. \\
&= A \frac{1}{2} \int_{\mathbb{R}^{n-2}} \left[-\left(R - x_n - \sum_{i=1}^{n-2} t_i \right)^2 \right]_{t_1=0}^{R - x_n - \sum_{i=2}^{n-1} t_i} \mathrm{d}\, t_2 \dots t_{n-1} \\
&= A \frac{1}{2} \int_{\mathbb{R}^{n-2}} \left(R - x_n - \sum_{i=2}^{n-2} t_i \right)^2 \mathrm{d}\, t_2 \dots t_{n-1} \\
&\vdots \\
&= \frac{A}{n!} (R - x_n)^n
\end{aligned}
$$

Finally, the density of X_n is

$$\boxed{g(x_n) = \frac{(n+1)}{R^{n+1}} (R - x_n)^n.}$$

Now, we compute the cumulative density function :

$$
\begin{aligned}
G(x_n) &= \int_0^{x_n} g(t_n)\, \mathrm{d}\, t_n \\
&= \frac{1}{R^{n+1}} \left[-(R - t_n)^{n+1} \right]_0^{R - x_n} \\
&= \frac{1}{R^{n+1}} \left(R^{n+1} - \big(R - x_n\big)^{n+1} \right).
\end{aligned}
$$

Finally,

$$\boxed{G(x_n) = 1 - (1 - \frac{x_n}{R})^{n+1}.}$$

The function ICDF_{x_n} is the reciprocal of G, and the result follows from Proposition 1. □