École polytechnique fédérale de Lausanne

Master in Mathematics

An overview on different preimage-sampling algorithms in lattice cryptography

Supervisors:
Prof. **Alessandro Chiesa**
Giacomo Fenzi

Author:
Francesco Baldino

Semester project spring 2024

Contents

1 Introduction

An important problem in lattice cryptography is the problem of efficiently finding short preimages of any given vector $\mathbf{u}$ (also called syndrome) for some wide matrix $\mathbf{A}$, that is finding a short vector $\mathbf{v}$ such that

$$\mathbf{Av} = \mathbf{u}$$

This results in an interesting problem as, with appropriate parameters, being able to find short preimages is considered a hard problem (it is usually referred as the *Inhomogenous Short Integer Solution*, or *Module - Inhomogeneous Short Integer Solution* when working modulo q) and this gives a one-way function which can be used as building block for cryptography.

The "shortness" constraint is usually expressed by requiring that the norm of the vector $\mathbf{v}$ is limited by some constant or, as it is in this case, by requiring that the preimage follows a prescribed distribution having a "short" output with high probablity, such as a discrete gaussian distribution. This also ensures that the distribution of the preimages does not depend on the trapdoor $\mathbf{R}$, effectively hiding it. When working in modulo q it is important to check the "shortness" constraint is a bound smaller than q as otherwise, at least for the homogeneous version of the problem where $\mathbf{u} = \mathbf{0}$, the problem becomes trivial.

In 2012 Micciancio and Peikert proposed a new definition of a trapdoor for some matrix $\mathbf{A}$ consisting of a matrix $\mathbf{R}$ such that

$$\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix} = \mathbf{HG}$$

for a gadget matrix $\mathbf{G}$ and some invertible tag matrix $\mathbf{H}$ (which is not strictly necessary for the preimage problem and can be taken as $\mathbf{H} = \mathbf{I}$, but it allows for some more flexibility when applying this concept in actual protocols). As shown in section 3 this trapdoor $\mathbf{R}$ allows to reduce the problem from a preimage problem on $\mathbf{A}$ to a preimage problem on $\mathbf{G}$ which can be solved efficiently [MP12].

Given $\mathbf{A}$ and $\mathbf{R}$, one can consider them respectively as the public and the private key of some public-key protocol, such as a signature protocol where the syndrome is some message-dependant vector and the preimage is the signature, allowing for easy verification using $\mathbf{A}$.

The goal of this semester project is to unify in common notation and compare three different preimage-sampling algorithms from [MP12], [LW15] and [JRLS23]

that solve the problem above. In section 3 we will introduce the algorithms, with details on different parameter choices and the different approaches. We will not go into details on the correctness of the algorithms, for which we refer to the original papers. At the beginning of section 3 we will also show breifly how to obtain a matrix $\mathbf{A}$ and an associated trapdoor $\mathbf{R}$, taking for granted the results from [MP12] showing that for such a generated matrix $\mathbf{A}$ the preimage problem with suitable parameters is hard (namely, showing that all the components of this matrix are statistically close to uniformly random). In section 4 we will give first a theoretical comparison of the three algorithms, analysing the pros and cons of each, and then a numerical comparison citing the sperimental analysis from [JRLS23] for the concrete parameter choices.

2 Preliminaries

We will use lowercase boldface (e.g. $\mathbf{v}, \mathbf{u}, \mathbf{x}$) for vectors and uppercase boldface (e.g. $\mathbf{A}, \mathbf{H}, \mathbf{G}$) for matrices.

Let $K = \mathbb{Q}(\zeta)$ be an algebraic extension field of degree n. We will denote with R the ring of integers of K. Given a modulo $q \geq 2$, we let $R_q = R/qR$. In most applications we have $R = \mathbb{Z}[x]/\langle \Phi_\nu(x) \rangle$ where Φ_ν is the ν-th cyclotomic polynomial, with $n = \varphi(\nu)$.

Given a positive-definite invertible matrix $\mathbf{\Sigma}$, a center $\mathbf{c} \in R^d$ and a lattice $\Lambda \subset R^d$, we define the centered discrete gaussian distribution over the coset $\mathbf{c} + \Lambda$ with covariance matrix $\mathbf{\Sigma}$ as the distribution defined by the following density function

$$\mathcal{D}_{\sqrt{\mathbf{\Sigma}},\mathbf{c},\Lambda}(\mathbf{x}) = \frac{\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c},\Lambda}(\mathbf{x})}{\sum_{\mathbf{z}\in\mathbf{c}+\Lambda} \rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c},\Lambda}(\mathbf{z})}$$

where

$$\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c},\Lambda}(\mathbf{x}) = \begin{cases} \exp\left(-\pi \cdot \mathbf{x}^\intercal \mathbf{\Sigma}^{-1} \mathbf{x}\right) & \text{if } \mathbf{x} \in \mathbf{c} + \Lambda \\ 0 & \text{otherwise} \end{cases}$$

This can be generalised to the case where $\mathbf{\Sigma}$ is not invertible by using *Moore-Penrose pseudoinverse* in place of $\mathbf{\Sigma}^{-1}$. For simplicity's sake, we use the following notations:

- If $\mathbf{\Sigma} = s^2\mathbf{I}$, we denote the distribution as $\mathcal{D}_{s,\mathbf{c},\Lambda}$
- If $\mathbf{c} = \mathbf{0}$, we omit it from the subscript

- If both $\mathbf{c} = \mathbf{0}$ and $\Lambda = R^d$, we denote directly as $\mathcal{D}^d_{\sqrt{\mathbf{\Sigma}}}$ (and similarly we denote $\mathcal{D}^d_s$ if it also holds that $\mathbf{\Sigma} = s^2\mathbf{I}$)

For ease of notation, given a matrix $\mathbf{A} \in R^{h\times k}$ and a syndrome $\mathbf{u} \in R^k$ we will use $\mathbf{v} \leftarrow \mathbf{A}^{-1}_{\sqrt{\mathbf{\Sigma}}}(\mathbf{u})$ to denote the distribution $\mathbf{v} \leftarrow \mathcal{D}^h_{\sqrt{\mathbf{\Sigma}}}$ conditioned to $\mathbf{Av} = \mathbf{u}$. If there exists $\mathbf{x} \in R^h$ such that $\mathbf{Ax} = \mathbf{u}$, then this distribution is equivalent to $\mathcal{D}_{\sqrt{\mathbf{\Sigma}},\mathbf{x},\Lambda^\intercal(\mathbf{A})}$. Similarly as before, if $\mathbf{\Sigma} = s^2\mathbf{I}$ we simply denote $\mathbf{A}^{-1}_s(\mathbf{u})$.
Pay attention to the subscript: later on when dealing with preimages for the gadget matrix $\mathbf{G}$ we will both use $\mathbf{G}^{-1}_{\sqrt{\mathbf{\Sigma}}}$ as a distribution as well as $\mathbf{G}^{-1}$ being the entry-wise base b decomposition, which in practice behaves as a (right) inverse for $\mathbf{G}$.

3 Preimage-sampling algorithms

We will first introduce the setting for the preimage problem, as well as the procedure to obtain the matrix-trapdoor pair and the reduction on the problem over $\mathbf{G}$.

Fix a modulo q, a base b and dimensions d and m. Let R be a ring as defined in the preliminaries $R_q = R/qR$. Let $k = \lceil \log_b(q) \rceil$. Let $\bar{\mathbf{A}} \in R_q^{d\times m}$ be a wide matrix and let $\mathbf{R} \in R^{m\times dk}$ be the trapdoor. Let $\mathbf{H} \in R_q^{d\times d}$ be an invertible matrix and let $\mathbf{G} \in R_q^{d\times dk}$ be the gadget matrix associated to q in base b, that is $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\dots|b^{k-1}]$. Let $\mathbf{A} = [\bar{\mathbf{A}}|\mathbf{HG} - \bar{\mathbf{A}}\mathbf{R}]$ be the matrix for which we want to be able to find preimages[1]. Given a syndrome $\mathbf{u} \in R_q^d$, we want to find a vector $\mathbf{v} \in R_q^m$ such that

$$\mathbf{Av} = \mathbf{u}$$

following a prescribed distribution $\mathbf{A}^{-1}_s(\mathbf{u})$.

It is clear that $\mathbf{R}$ is a trapdoor for the matrix $\mathbf{A}$, as by construction it holds

$$\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix} = \mathbf{HG}$$

The reason why it is useful to have the trapdoor to solve this problem is that since the relationship holds, then to obtain a vector $\mathbf{v}$ such that $\mathbf{Av} = \mathbf{u}$ (albeit with a skewed distribution) it suffices to find $\mathbf{z}$ such that $\mathbf{Gz} = \mathbf{H^{-1}}(\mathbf{u})$ and letting

[1]Ideally we would also want it to be difficult to find preimages for this matrix without knowledge of $\mathbf{R}$. This can indeed be achieved, and precise details on this can be found in section 5.2 of [MP12], but the main idea is that it is possible to generate $\bar{\mathbf{A}}$ and $\mathbf{R}$ such that $\mathbf{A}$ is statistically close to uniform

$\mathbf{v} = \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}$. Such a $\mathbf{v}$ clearly satisfies the desired equality, and can be computed efficiently (if one can compute efficiently $\mathbf{z}$) and it is then "just" a matter of fixing the distribution. Indeed, such a vector $\mathbf{v}$ would have a skewed gaussian distribution having covariance matrix that depends on $\mathbf{R}$, so giving $\mathbf{v}$ would leak information on the trapdoor. Instead, we would like to make sure that $\mathbf{v}$ follows the prescribed target distribution that does not depend on $\mathbf{R}$, making sure not to leak any information on $\mathbf{R}$. To do so, one could add a perturbation to the result that would hopefully fix the distribution, such as letting $\mathbf{p}$ be some perturbation and outputting $\mathbf{v} = \mathbf{p} + \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}$. This of course would change the result of $\mathbf{Av} = \mathbf{u} + \mathbf{Ap}$, which is not the desired result. To solve this issue, one could preventively modify the syndrome and sampling $\mathbf{z}$ such that $\mathbf{Gz} = \mathbf{H}^{-1}(\mathbf{u} - \mathbf{Ap})$ so that with this choice of $\mathbf{v}$ we indeed get $\mathbf{Av} = \mathbf{u}$.

As for the matter of finding $\mathbf{z}$ such that $\mathbf{Gz} = \mathbf{u}$, i.e. solving the reduced problem, we refer to section 4 of [MP12], where one can find efficient algorithms to solve this problem for a variety of parameter choices.

3.1 Micciancio-Peikert sampling (2012)

For this algorithm we let $m = 2d$. As for the choice of the parameter s, this is influenced by the way of generating $\mathbf{z}$. For this purpose, Micciancio and Peikert use an algorithm which is able to sample from $\mathbf{G}^{-1}_{\sqrt{\mathbf{\Sigma_G}}}(\mathbf{u})$ for some parameter-dependant positive-definite matrix $\mathbf{\Sigma_G}$. The subroutine proposed in [MP12] obtains $\mathbf{\Sigma_G} = s_{\mathbf{G}}^2 \mathbf{I}_{dk}$ where $s_{\mathbf{G}} = 2 \cdot \omega(\sqrt{\log d})$ or $s_{\mathbf{G}} = \sqrt{5} \cdot \omega(\sqrt{\log d})$ depending on the modulo q, but the sampling algorithm is presented with a generic $\mathbf{\Sigma_G}$ to allow for easy substitution of any other subroutine for sampling from $\mathbf{G}^{-1}_{\sqrt{\mathbf{\Sigma_G}}}(\mathbf{u})$. The parameter s is then chosen such that $s \geq s_1 \left(\begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \cdot \sqrt{\mathbf{\Sigma_G}} \right)$, that is such that the matrix

$$\mathbf{\Sigma_p} = s^2 \mathbf{I}_{d(2+k)} - \begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{\Sigma_G} \begin{bmatrix} \mathbf{R^T} & \mathbf{I} \end{bmatrix} = s^2 \mathbf{I}_{d(2+k)} - s_{\mathbf{G}}^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^T & \mathbf{R} \\ \mathbf{R}^T & \mathbf{I} \end{bmatrix}$$

is positive-definite. As mentioned in the analysis in [JRLS23] this leads to a gaussian width proportional to $s = \Theta(b(\sqrt{2nd} + \sqrt{ndk}))$.

The way Micciancio and Peikert solve the distribution issue is to sample a perturbation $\mathbf{p}$ from a gaussian of width $\mathbf{\Sigma_p}$ so that by the convolution theorem, the

perturbed value $\mathbf{v} = \mathbf{p} + \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\mathbf{z}$ has precisely distribution $\mathbf{A}_s^{-1}(\mathbf{u})$.

The resulting algorithm is given in Algorithm 1.

Algorithm 1 Micciancio-Peikert preimage-sampling algorithm

Input: $\bar{\mathbf{A}}, \mathbf{R}, \mathbf{H}, \mathbf{G}, s, \mathbf{u}$ as above

Output: A short preimage $\mathbf{v} \in R_q^d$ following a distribution statistically close to $\mathbf{A}_s^{-1}(\mathbf{u})$

1: $\mathbf{p} \leftarrow \mathcal{D}^{d(2+k)}_{\sqrt{\mathbf{\Sigma_P}}}$

2: Parse $\mathbf{p} \rightarrow \begin{bmatrix}\mathbf{p}_1\\ \mathbf{p}_2\end{bmatrix}$ with $\mathbf{p}_1 \in R^{2d}$ and $\mathbf{p}_2 \in R^{dk}$

3: $\bar{\mathbf{w}} \leftarrow \bar{\mathbf{A}}(\mathbf{p}_1 - \mathbf{R}\mathbf{p}_2) \mod qR$

4: $\mathbf{w} \leftarrow \mathbf{G}\mathbf{p}_2 \mod qR$

5: $\mathbf{y} \leftarrow \mathbf{H}^{-1}(\mathbf{u} - \bar{\mathbf{w}}) - \mathbf{w} = \mathbf{H}^{-1}(\mathbf{u} - \mathbf{A}\mathbf{p})$

6: $\mathbf{z} \leftarrow \mathbf{G}^{-1}_{\sqrt{\mathbf{\Sigma_G}}}(\mathbf{y})$

7: $\mathbf{v}_2 \leftarrow \mathbf{p}_2 + \mathbf{z}$

8: $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}$

9: **return** $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix}$

This algorithm is very straightforward, and will end up having the best parameter sizes of all. The main downside is the high specialization, as it is built only on gaussian distributions, and the necessity of sampling from a gaussian of width $\mathbf{\Sigma_p}$ which is highly non-spherical, and that can be too computationally expensive in some contexts.

3.2 Lyubashevsky-Wichs sampling (2015)

The approach developed by Lyubashevsky and Wichs [LW15] to fix the distribution of the output is to obtain the desired distribution via rejection-sampling. To do so, they make use of a rejection-sampling lemma which allows a bit more flexibility on the desired source and target distributions (respectively the distribution of the perturbation and the distribution of the output). For comparison sake we will give directly the instantiation in the gaussian setting. The original algorithm also gives an instantiation for uniform distributions.

We will give a slightly modified version of the algorithm presented in [LW15] as the original algorithm solves for $\mathbf{A} = [\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R} - \mathbf{H}\mathbf{G}]$ instead of $\mathbf{A} = [\bar{\mathbf{A}}|\mathbf{H}\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}]$.

The version presented here will find preimages for the latter, to keep the simmetry with the other algorithms. In the end it is just a matter of changing some signs.

This algorithm uses slightly larger parameters with $m = dk$ (compared to the previous $m = 2d$). For the width of the gaussian, this algorithm uses $s = 2k_2\sqrt{\lambda}$ where $k_2 = \Theta(m + \sqrt{\lambda m})$ is a probabilistic bound on $||\mathbf{R}\mathbf{v}_2||$, [Ver11]. This leads to a slightly more challenging parameters comparison as the dependency on the security level λ is left implicit in the choice of the other parameters in the other two algorithms. In the analysis from [JRLS23], it is shown that concretely this leads to a width proportional to $s = \alpha \cdot (b-1)\sqrt{ndk}(2\sqrt{ndk} + t)$ for a constant factor $\alpha \approx 8$, which in practice, when applied to a signature scheme, lead to signature sizes roughly twice as big as the ones obtained from [MP12].

This algorithm breaks the simmetry on the perturbation of $\mathbf{v}_1$ and $\mathbf{v}_2$, since as pointed out more precisely in [JRLS23] there shouldn't be a need of perturbation on $\mathbf{v}_2$ to hide the secret key $\mathbf{R}$ since $\mathbf{v}_2$ does not depend in any way on the secret key. This leads to leaving $\mathbf{v}_2 = \mathbf{z}$ unperturbed and perturbing only the $\mathbf{v}_1$ component of the output, using rejection sampling to "hide" $\mathbf{R}$, as in to obtain a distribution that does not depend on $\mathbf{R}$.

Notice that this algorithm does not use the $\mathbf{G}$-sampler like in [MP12] to find a suitable preimage $\mathbf{z}$, as it uses directly the deterministic function $\mathbf{G}^{-1}$. Note that this is not the inverse of the matrix $\mathbf{G}$, as this matrix isn't invertible, but it is the entry-wise base b decomposition function. It holds that $\mathbf{G} \cdot \mathbf{G}^{-1}\mathbf{x} = \mathbf{x}$.

The adapted version of the algorithm from [LW15] is given in algorithm 2. We stress that this is only the gaussian instantiation and not the general instantiation, which allows any source and target distribution as long as rejection sampling can be performed to obtain a distribution statistically close to the target one.

Algorithm 2 Lyubashevsky-Wichs preimage-sampling with rejection sampling

Input: $\bar{\mathbf{A}}, \mathbf{R}, \mathbf{H}, \mathbf{G}, s, \mathbf{u}$ as above

Output: A short preimage $\mathbf{v} \in R_q^d$ following a distribution statistically close to $\mathbf{A}_s^{-1}(\mathbf{u})$

1: $\mathbf{p}_1 \leftarrow \mathcal{D}_s^{dk}$
2: $\bar{\mathbf{w}} \leftarrow \bar{\mathbf{A}}(\mathbf{p}_1) \mod qR$
3: $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{H}^{-1}(\mathbf{u} - \bar{\mathbf{w}}))$
4: $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$
5: Sample a continuous $u \leftarrow \mathcal{U}([0, 1])$
6: **if** $u > \min\left(1, \frac{\mathcal{D}_s^{dk}(\mathbf{v}_1)}{M \cdot \mathcal{D}_s^{dk}(\mathbf{p}_1)}\right)$ **then**
7: Go to 1
8: **end if**
9: **return** $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$

3.3 Jeudy-Roux–Langlois-Sanders sampling (2023)

Finally, we introduce the algorithm given proposed by Jeudy. Roux-Langlois and Sanders [JRLS23]. This is an optimization of the algorithm proposed in [LW15].

The idea behind this optimization is to reasonably suppose that the syndrome $\mathbf{u}$ cannot be controlled by an adversary and is instead an uniformly random variable. This is the case, for example, in an *Hash-and-Sign* protocol where $\mathbf{u}$ would be (a representation of) the hash of the message, and as such would be an uniformly distribuited value. This reasonable additional hypothesis allows to relax a bit the parameters at no security cost (although at a loss of generality for applications of the algorithm).

This allows to use $m = 2d$ which is the same parameter choice as the original algorithm from [MP12]. As more precisely explained in Corollary 3.1 of [JRLS23], choosing $s = \Theta(b\sqrt{ndk}(\sqrt{2nd} + \sqrt{ndk}))$ enables the authors to use a rejection-sampling lemma similar to the one in [LW15] to conclude that the distribution of the output is statistically close to the desired one, hence the algorithm is correct.

We give the resulting algorithm in algorithm 3

Algorithm 3 Jeudy-Roux–Langlois-Sanders preimage-sampling with rejection sampling

Input: $\bar{\mathbf{A}}, \mathbf{R}, \mathbf{H}, \mathbf{G}, s, \mathbf{u}$ as above

Output: A short preimage $\mathbf{v} \in R_q^d$ following a distribution statistically close to $\mathbf{A}_s^{-1}(\mathbf{u})$

1: $\mathbf{p}_1 \leftarrow \mathcal{D}_s^{2d}$
2: $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{H}^{-1}(\mathbf{u} - \bar{\mathbf{A}}\mathbf{p}_1))$
3: $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$
4: Sample a continuous $u \leftarrow \mathcal{U}([0,1])$
5: **if** $u > \min\left(1, \frac{\mathcal{D}_s^{2d}(\mathbf{v}_1)}{M \cdot \mathcal{D}_s^{2d}(\mathbf{p}_1)}\right)$ **then**
6: Go to 1
7: **end if**
8: **return** $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$

The resulting algorithm, while using the same sizes as the original from [MP12], uses gaussians that are much wider. At the same time, just like in [LW15] this construction is more general and while we only exhibit the gaussian instantiation, this could also be applied to different source and target distributions. Again just like in [LW15], this algorithm does not require to sample from highly non-spherical gaussians, which can be computationally expensive.

4 Comparisons

We will now give a theoretical and numerical comparison of the three algorithms from section 3. As noted in the algorithm explanations, they mainly differ in the following aspects:

- The method used to ensure the desired target distribution for the output
- The parameter m, influencing the dimension of both public and private keys, as well as the dimension of the preimages, which can matter in contexts such as a signature scheme
- The width s, again influencing the expected norm of the preimages
- The method used to obtain a solution of the reduced problem on the gadget matrix $\mathbf{G}$

In the theoretical comparison we will focus mainly on the first and the second point. We will analyse the difference in gaussian widths in the numerical comparison using as a starting point the numerical analysis from section 4 of [JRLS23].

4.1 Theoretical comparison

Starting with the analysis on the parameter m, we note that both algorithm 1 and algorithm 3 achieve $m = 2d$, while algorithm 2 uses $m = dk$. Since in any reasonable setting one should expect to have $k \gg 2$, algorithms 1 and 3 obtain much smaller dimensions, which is clearly desirable.

The main difference, though, is the way they obtain the desired target distribution for the output. On one side, algorithm 1 obtains the desired gaussian by perturbing with some randomness from a suitably crafted gaussian, the purpose of which is to fix the skewed distribution of the preimage $\begin{bmatrix} \mathbf{R} \\ \mathbf{I} \end{bmatrix} \mathbf{z}$. To do so, they sample from a gaussian having covariance matrix being the difference of the target covariance and the covariance of said preimage. By the convolution theorem, this clearly results in the target distribution, but it ends up using a highly non-spherical gaussian for the perturbation. As pointed out int [JRLS23], this operation is computationally expensive, and ends up representing the most part of the computation time.

On the other side, both algorithm 2 and algorithm 3 obtain the desired target distribution via rejection sampling. This not only solves the problem of having to sample from highly non-spherical gaussians, but also allows for a much wider range of possible source and target distributions: while algorithm 1 is only applicable for gaussian distributions, algorithms 2 and 3 allow for any source and target distributions for which rejection-sampling is possible, that is for any source and target distribution with reasonably small Rényi divergence (see theorem 3.1 of [JRLS23]). This, of course, comes at a cost of possible repetition: while algorithm 1 is "one-shot", algorithms 2 and 3 have to perform rejection sampling with possibly many repetitions, though with a reasonable bound on the number of repetitions (see theorem 3.1 of [LW15]).

In conclusion, algorithm 1 from [MP12] leads to potentially smaller parameters and a more succint algorithm, where algorithm 2 from [LW15] and algorithm 3 from [JRLS23] are preferrable for their adaptability and being potentially less computationally expensive.

Furthermore, while algorithm 3 seems to be better than algorithm 2 in every aspect, we remind that [JRLS23] obtains reduced dimensions by introducing the additional hypothesis that the syndrome $\mathbf{u}$ is uniformly random and not controllable by an andversary. This is clearly the case for applications such as an *Hash-and-sign* paradigm, but is inherently less flexible.

4.2 Numerical comparison

We take and analyse the results from the estimates in section 4 of [JRLS23]. The results are related to the application of these algorithms to an *Hash-and-sign* protocol for a target $\lambda = 128$. The parameters d and q are chosen to minimize the size of the signature for the given security level. The value $\lambda_{\text{M-SIS}}$ is the resulting security level for the associated *M-SIS* problem for the chosen parameters. The ring degree is set to $n = 256$. The results are given for different choices of the basis b.

	$\lambda_{\text{M-SIS}}$	q	d	s
$b = 2$	146	$\approx 2^{15.2}$	5	2596
$b = 4$	150	$\approx 2^{15.6}$	5	3461
$b = q^{1/5}$	147	$\approx 2^{16.8}$	5	7661
$b = q^{1/3}$	131	$\approx 2^{19.7}$	5	56804
$b = q^{1/2}$	154	$\approx 2^{26.7}$	7	6616938

Table 1: results relative to algorithm 1

	$\lambda_{\text{M-SIS}}$	q	d	s
$b = 2$	131	$\approx 2^{23.6}$	6	572109
$b = 4$	130	$\approx 2^{23.8}$	6	901768
$b = q^{1/5}$	130	$\approx 2^{27.3}$	6	5586865
$b = q^{1/3}$	133	$\approx 2^{30.6}$	7	105308864
$b = q^{1/2}$	138	$\approx 2^{40.5}$	9	96061795597

Table 2: results relative to algorithm 2

	$\lambda_{\text{M-SIS}}$	q	d	s
$b = 2$	157	$\approx 2^{22.5}$	6	362140
$b = 4$	151	$\approx 2^{23.2}$	6	645772
$b = q^{1/5}$	134	$\approx 2^{25.6}$	6	3576993
$b = q^{1/3}$	137	$\approx 2^{30.3}$	7	90206170
$b = q^{1/2}$	138	$\approx 2^{40.3}$	9	90202905475

Table 3: results relative to algorithm 3

As we can see, algorithm 1 leads to gaussians that are much tighter than the ones obtained from algorithms 2 and 3. This supports the results obtained from the theory (though an exact comparison would be difficult) showing that the algorithm from [MP12] obtains generally smaller parameters, leading to better results at the cost of restriction to gaussian distributions only and a possibly more computationally expensive algorithm.

References

[JRLS23] Corentin Jeudy, Adeline Roux-Langlois, and Olivier Sanders. Phoenix: Hash-and-sign with aborts from lattice gadgets. Cryptology ePrint Archive, Paper 2023/446, 2023. https://eprint.iacr.org/2023/446.

[LW15] Vadim Lyubashevsky and Daniel Wichs. Simple lattice trapdoor sampling from a broad class of distributions. PKC, 2015.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. EUROCRYPT, 2012.

[Ver11] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices, 2011.