Lattice-based signatures

Rényi divergence analysis

Pijnenburg, J.A.M.

Award date:
2017

Eindhoven University of Technology

Department of Mathematics and Computer Science

Discrete Mathematics

Lattice-Based Signatures
Rényi Divergence Analysis

Author:
J.A.M. Pijnenburg
(s158700)

Supervisors:
prof. dr. S.D. Galbraith
prof. dr. T. Lange

A Master Thesis in Mathematics

August 31, 2017

Contents

Chapter 1

Introduction

Factoring large numbers is believed to be hard on classical computers. At least there is no known (classical) polynomial time algorithm to factor integers at present. However in 1994 Shor had already formulated a quantum polynomial time algorithm to factor integers efficiently on a quantum computer [1]. In 2001 Shor's algorithm has been implemented on a quantum computer using 10 qubits to factor the number 15 [2]. By transforming the factorization problem into an optimization problem, a group of Chinese researchers was able to factor 143 using only four qubits [3]. This method does not use Shor's algorithm and does not require prior knowledge of the answer. In fact it turns out they did not only factor 143, but a whole class of numbers which 143 belongs to. Hence, as of today the largest natural number factored on a quantum computer is 56153 [4]. Moreover the researchers in [4] point out that with the same technique and using 6 qubits the number 291311 can be factored on a quantum computer. Also 175 could be factored, using only 3 qubits and making it the first number to be factored on a quantum computer consisting of 3 prime factors. It is interesting to note that factoring numbers on a quantum computer via the discrete optimization algorithm is easiest for numbers consisting of two primes factors, such as used in RSA, because it will involve at most 3 qubit interactions [4]. On the contrary, classical algorithms find these numbers the most difficult to factor [4]. It has been difficult to scale up quantum computing, but we can conclude quantum computing is advancing. As soon as it can be applied on a large scale, current cryptosystems can be broken efficiently. It is clear the ability to factor large integers will break RSA but it should be noted Shor's algorithm [1] can also solve the discrete logarithm problem efficiently and

hence can be used to break elliptic curve cryptography.

Post-quantum cryptography focuses on cryptographic algorithms that will be resistant to attacks by quantum computers. An interesting candidate appears to be lattice-based cryptography. Some lattice-based cryptosystems come with security proofs which show that an attacker who can break the cryptosystem will also be able to solve problems on lattices that are known to be hard for quantum computers. The importance of a security proof becomes apparent when considering for example the GGH signature scheme [5] which was completely broken by Nguyen and Regev [6].

However, these schemes based on strong security assumptions often come with practical issues such as large key size, large signatures or long computation times. In order to improve upon the practical applicability of these schemes, this work will explore the Rényi divergence as an alternative to the statistical distance in order to obtain tighter security proofs. In doing so, we will be able to achieve smaller parameters for the desired security level. We will demonstrate this by applying the technique to Lyubashevsky's [7] signature scheme.

After going over the notation in chapter 2, we will discuss the background regarding signatures and lattices in chapter 3. This will lead us towards the SIS problem, which is crucial in the security assumption of Lyubashevsky's [7] signature scheme. In chapter 4 we will explore the Rényi divergence in order to develop a deeper mathematical understanding. We will provide simplified proofs for the case of discrete distributions as opposed to the general discussion in [8]. This chapter will culminate in the probability preservation property which explains why the Rényi divergence can be used to replace the statistical distance.

In chapter 5 we will first examine a signature scheme employing rejection sampling to acquaint ourselves with the topic. In section 5.2 we consider Lyubashevsky's [7] signature scheme which employs the discrete Gaussian distribution in order to obtain better practical results than in the first section. Next we suggest parameters that offer the desired security based on our tightened security proof and compare these with the original parameters. In section 5.4 we follow Lyubashevsky's idea to violate one of the requirements in the security proof and provide an alternative proof to significantly reduce the signature size. Observing the signature size in section 5.4 gets close to the maximum transmission unit, we will focus on reducing our signature size even further in the last section. At the cost of increasing computation time, we will obtain a signature size below the maximum transmission unit.

Finally in chapter 6 we draw attention to BLISS [9], a signature scheme based on Lyubashevsky's [7] signature scheme. It offers significant performance improvements compared to the original scheme by replacing the sampling distribution with the bimodal Gaussian distribution.

Chapter 2

Notation

In this chapter we will define the notation used throughout the work in order to avoid ambiguity.

We recall for any $p \geq 1$, the l_p-norm of a vector $x \in \mathbb{R}^n$ is given by

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

This work will use the following special cases, where we employ the convention $p = 2$ when p is omitted.

The l_1-norm (taxicab norm)

$$||x||_1 = \sum_{i=1}^{n} |x_i|,$$

the l_2-norm (Euclidean norm)

$$||x|| = ||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2},$$

and the l_∞-norm (max norm)

$$||x||_\infty = \lim_{p \to \infty} ||x||_p = \max_{i=1}^{n} |x_i|.$$

We will use log exclusively to denote the natural logarithm. The binary logarithm will always be denoted by $\log_2$.

As is standard, $\langle a, b \rangle$ will be used to denote the inner product of the vectors a and b.

For any two sets A and B, we will write $A \subset B$ to denote A is a subset of B. To denote A is a *proper* subset of B we would write $A \subsetneq B$.

Let $f : X \to \mathbb{R}$ be a function. The support of f, denoted $\text{supp}(f)$, is the set of points in X where f is non-zero:

$$\text{supp}(f) = \{x \in X \mid f(x) \neq 0\}.$$

We will define $\mathbb{R}_{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$ as shorthand notation for the set of non-negative real numbers.

Lastly, we would like to recall the notations of *asymptotic complexity*.

Let $f(n), g(n)$ be two positive real valued functions.

- $f = O(g)$ if there exist constants $c, k \in \mathbb{R}_{\geq 0}$ such that for all $n \geq k$, $f(n) \leq c \cdot g(n)$.
- $f = o(g)$ if $\lim_{n\to\infty} f(n)/g(n) = 0$.
- $f = \Omega(g)$ if there exist constants $c, k \in \mathbb{R}_{\geq 0}$ such that for all $n \geq k$, $f(n) \geq c \cdot g(n)$.
- $f = \omega(g)$ if $\lim_{n\to\infty} g(n)/f(n) = 0$.
- $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

Chapter 3

Preliminaries

In this chapter we will first recall the definition of a digital signature scheme, define the types of adversaries and discuss in what sense a digital signature scheme can be broken. We introduce lattices in section 3.2 and in the subsequent section we discuss lattice problems. In particular we focus on the hardness of the SIS problem as this problem is the crucial underlying security assumption in Lyubashevsky's [7] signature scheme.

3.1 Digital signatures

Definition 3.1. [10] A digital signature scheme consists of three (probabilistic) polynomial time algorithms $(\mathcal{G}, \Sigma, \mathcal{V})$.

1. *Key generation algorithm* $\mathcal{G}$. $\mathcal{G}$ is a probabilistic algorithm with random tape ω. On input 2^k, where k is the security parameter, the algorithm $\mathcal{G}$ produces a pair (K_p, K_s) of matching public and secret keys.

2. *Signing algorithm* Σ. Given a message μ and a matching keypair (K_p, K_s), Σ produces a valid signature σ. Σ may be probabilistic with random tape ω.

3. *Verification algorithm* $\mathcal{V}$. Given a signature σ, a message μ and a public key K_p, $\mathcal{V}$ tests whether σ is a valid signature on μ with respect to K_p.

We distinguish two kinds of attacks on a digital signature scheme: key-only and message attacks. In the former the adversary only has access to the signer's public key. In the latter case the adversary also has access to a list of message and signature pairs. We consider four subcases of message attacks, depending on how the adversary can choose the list of message and signature pairs.

Definition 3.2. [10] We distinguish four different cases of message attacks.

1. *Plain known-message attack.* In this case the adversary has access to a list of message and signature pairs, but he cannot choose the messages.

2. *Generic chosen-message attack.* In this case the adversary can choose a list of messages for which he wants to obtain valid signatures before he attempts to break the digital signature scheme. However, he must choose the list of messages before accessing the public key of the signer. So the adversary uses the same attack against everyone.

3. *Directed chosen-message attack.* The adversary can choose a list of messages for which he wants to obtain valid signatures before he attempts to break the digital signature scheme. However, he now has access to the signer's public key before chosing the list of messages. Therefore this attack is specifically directed at the signer's public key.

4. *Adaptively chosen-message attack.* Here the adversary has access to the signer's public key and he can use the signer as an oracle. He does not have to specify his list of messages beforehand: he can query a message and adapt his next query based on the result.

Clearly, of these attacks, the adaptively chosen-message attack is the most powerful attack the adversary can undertake. One might wonder why the signer would cooperate as an oracle to the adversary, but in general the signer should be able to sign arbitrary documents without fear of compromising his security. Therefore we will allow the adversary an adaptively chosen-message attack in order to break the digital signature scheme.

Definition 3.3. [10] A digital signature scheme can be broken in the following sense.

1. *A total break.* The adversary obtains the signer's secret key.

2. *Universal forgery.* The adversary constructs an efficient algorithm which is able to sign any message.

3. *Selective forgery.* The adversary is able to produce a valid signature for a particular message, chosen before he attempts the forgery.

4. *Existential forgery.* The adversary is able to produce a valid pair of message and signature. The message may be random or nonsense.

Definition 3.4. *Negligibility.* A non-negative function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if for every $\gamma \in \mathbb{N}$ there exists a $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $f(k) \leq 1/k^{\gamma}$.

We will be most conservative and call a digital signature scheme (strongly) secure if an adversary using an adaptively chosen-message attack is unable to produce an existential forgery. The difference between Definition 3.5 and Definition 3.6 lies in a subtle change in the notion of what constitutes a new signature.

Definition 3.5. [7] A digital signature scheme $(\mathcal{G}, \Sigma, \mathcal{V})$ is *secure* if for every probabilistic polynomial time algorithm $\mathcal{A}$, after seeing the public key and $\{(\mu_1, \sigma_1), ..., (\mu_q, \sigma_q)\}$ for any q messages μ_i of its choosing where q is polynomial in the security parameter k, the probability that $\mathcal{A}$ produces a valid pair (μ, σ) such that $\mu \neq \mu_i \ \forall i \in \{1, .., q\}$, is negligibly small in k.

Definition 3.6. A digital signature scheme $(\mathcal{G}, \Sigma, \mathcal{V})$ is *strongly secure* if for every probabilistic polynomial time algorithm $\mathcal{A}$, after seeing the public key and $\{(\mu_1, \sigma_1), ..., (\mu_q, \sigma_q)\}$ for any q messages μ_i of its choosing where q is polynomial in the security parameter k, the probability that $\mathcal{A}$ produces a valid pair $(\mu, \sigma) \neq (\mu_i, \sigma_i) \ \forall i \in \{1, .., q\}$, is negligibly small in k.

3.2 Lattices

Definition 3.7. A *full rank lattice* $\mathcal{L} \subset \mathbb{R}^n$ is a set of integer linear combinations of n linearly independent vectors $b_1, ..., b_n \in \mathbb{R}^n$:

$$\mathcal{L}(b_1, ..., b_n) = \left\{ \sum_{i=1}^{n} z_i b_i \ \mid \ z_i \in \mathbb{Z} \right\}.$$

The set $\{b_1, ..., b_n\}$ is called a basis for the lattice and we can represent it by the matrix $B = [b_1, ..., b_n] \in \mathbb{R}^{n \times n}$, where each basis vector is a column. Then we can write

$$\mathcal{L}(B) = \{Bz \ \mid \ z \in \mathbb{Z}^n\}.$$

We will implicitly assume all lattices are of full rank.

Definition 3.8. Two lattice bases $B, B' \in \mathbb{R}^{n\times n}$ are called *equivalent* if they generate the same lattice, i.e $\mathcal{L}(B) = \mathcal{L}(B')$.

We will prove the following lemma that relates two equivalent lattice bases.

Lemma 3.9. *Two lattice bases $B, B' \in \mathbb{R}^{n\times n}$ are equivalent if and only if there exists a unimodular matrix $U \in \mathbb{R}^{n\times n}$ such that $B' = BU$.*

Proof. Suppose there exists a unimodular matrix $U \in \mathbb{R}^{n\times n}$ such that $B' = BU$. By definition of unimodularity, U^{-1} exists and $U^{-1}z \in \mathbb{Z}^n \iff z \in \mathbb{Z}^n$. Then

$$\begin{aligned}\mathcal{L}(B') &= \{B'z' \mid z' \in \mathbb{Z}^n\} = \{BUz' \mid z' \in \mathbb{Z}^n\} = \{BUU^{-1}z \mid U^{-1}z \in \mathbb{Z}^n\} \\ &= \{Bz \mid U^{-1}z \in \mathbb{Z}^n\} = \{Bz \mid z \in \mathbb{Z}^n\} = \mathcal{L}(B).\end{aligned}$$

Now suppose the above relation $\mathcal{L}(B') = \mathcal{L}(B)$ holds. Since B', B are bases, it follows there exists a unique invertible $U \in \mathbb{R}^{n\times n}$ such that $B' = BU$. But we have $U^{-1}z \in \mathbb{Z}^n \iff z \in \mathbb{Z}^n$, so U is unimodular by definition. □

Definition 3.10. We define the distance between two lattice points $x, y \in \mathcal{L}$ as

$$\text{dist}(x, y) = ||x - y||.$$

Definition 3.11. The *minimum distance* of a lattice $\mathcal{L}$, denoted $\lambda_1(\mathcal{L})$, is the minimum distance between any two distinct lattice points:

$$\lambda_1(\mathcal{L}) = \min\{\text{dist}(x, y) \mid x \neq y \in \mathcal{L}\}.$$

Lemma 3.12. *The minimum distance $\lambda_1(\mathcal{L})$ of a lattice $\mathcal{L}$ is equal to the length of the shortest nonzero lattice vector.*

$$\lambda_1(\mathcal{L}) = \min\{||x|| \mid x \in \mathcal{L}\backslash\{0\}\}.$$

Proof. Let $v_1, v_2 \in \mathcal{L}(B)$ be any two distinct lattice vectors such that $\lambda_1(\mathcal{L}(B)) = \text{dist}(v_1, v_2)$. Then, their difference is also in the lattice:

$$v_1 - v_2 = Bz_1 - Bz_2 = B(z_1 - z_2) \in \mathcal{L}(B).$$

Thus, $v_1 - v_2$ is a nonzero lattice with length equal to the minimum distance.

$$\lambda_1(\mathcal{L}(B)) = \text{dist}(v_1, v_2) = \text{dist}(v_1 - v_2, 0) = ||v_1 - v_2||$$

By definition, for any nonzero lattice vector v, we have:

$$\lambda_1(\mathcal{L}(B)) \leq \text{dist}(v, 0) = ||v||.$$

Therefore, $||v_1 - v_2||$ is the shortest length of any nonzero lattice vector. □

The minimum distance can be generalized to the i-th successive minimum.

Definition 3.13. [11] The i-th successive minimum $\lambda_i(\mathcal{L})$ is the smallest radius r such that the closed ball $\mathcal{B}(r) = \{x \in \mathbb{R}^n \mid ||x|| \leq r\}$ contains i linearly independent lattice points:

$$\lambda_i(\mathcal{L}) = \min\{r \in \mathbb{R}^n_{\geq 0} \mid \dim(\text{span}(\mathcal{L} \cap \mathcal{B}(r))) \geq i\}.$$

Definition 3.14. The *minimum distance* of a lattice $\mathcal{L}$ measured in the l_∞ norm is denoted by $\lambda_1^\infty(\mathcal{L})$:

$$\lambda_1^\infty(\mathcal{L}) = \min\{||x - y||_\infty \mid x \neq y \in \mathcal{L}\}.$$

Definition 3.15. The *determinant* of a lattice is the absolute value of the determinant of the basis matrix

$$\det(\mathcal{L}(B)) = |\det(B)|.$$

This value is also called the volume of the lattice and it is inversely related to the density of lattice points. The definition is well defined as the determinant of a lattice does not depend on the choice of basis.

Lemma 3.16. *Let $B, B' \in \mathbb{R}^{n \times n}$ be two equivalent lattice bases. Then*

$$\det(\mathcal{L}(B)) = \det(\mathcal{L}(B')).$$

Proof. By Lemma 3.9 we have $B' = BU$ for some unimodular matrix U.

$$\det(\mathcal{L}(BU)) = |\det(BU)| = |\det(B)| \cdot |\det(U)| = |\det(B)| = \det(\mathcal{L}(B)).$$

□

Definition 3.17. [12] The dual lattice of $\mathcal{L}$, denoted $\mathcal{L}^*$, is defined to be

$$\mathcal{L}^* = \{x \in \mathbb{R}^n \mid \forall y \in \mathcal{L}, \langle x, y \rangle \in \mathbb{Z}\}.$$

We will prove the following lemma to efficiently compute a basis for the dual of a lattice $\mathcal{L}$.

Lemma 3.18. *Let $\mathcal{L}(B)$ be a full rank lattice, then its dual $\mathcal{L}^*(B)$ is given by $\mathcal{L}((B^{-1})^T)$.*

Proof. B is a full rank matrix, so $(B^{-1})^T$ exists and is a full rank matrix. Take $x \in \mathcal{L}((B^{-1})^T)$ and $y \in \mathcal{L}(B)$.

$$\langle x, y \rangle = \langle (B^{-1})^T z_1, Bz_2 \rangle = ((B^{-1})^T z_1)^T Bz_2 = z_1^T B^{-1} Bz_2 = z_1^T z_2 \in \mathbb{Z}.$$

Thus $\mathcal{L}((B^{-1})^T) \subset \mathcal{L}^*(B)$.

Now take $x \in \mathcal{L}^*(B)$ and $y \in \mathcal{L}(B)$. By definition

$$\langle x, y \rangle = \langle x, Bz_2 \rangle = x^T Bz_2 = z_2^T B^T x \in \mathbb{Z} \qquad \forall z_2 \in \mathbb{Z}^n,$$

where we can take the transpose in the last equality because $x^T Bz_2$ is a scalar. Therefore $B^T x \in \mathbb{Z}^n$. Then we can write $x = (B^T)^{-1} z_1 = (B^{-1})^T z_1$ for some $z_1 \in \mathbb{Z}^n$. Thus $\mathcal{L}^*(B) \subset \mathcal{L}((B^{-1})^T)$.

Hence we can conclude $\mathcal{L}((B^{-1})^T) = \mathcal{L}^*(B)$. □

Corollary 3.19. *For any lattice $\mathcal{L}$ we have* $\det(\mathcal{L}^*) = 1/\det(\mathcal{L})$.

Definition 3.20. [13] A *q-ary lattice* $\mathcal{L}$ is a lattice satisfying $q\mathbb{Z}^m \subset \mathcal{L} \subset \mathbb{Z}^m$ for some integer q.

Remark. Any integer lattice $\mathcal{L} \subset \mathbb{Z}^m$ is a q-ary lattice for some q, for example $q = \det(\mathcal{L})$, because $\det(\mathcal{L}) \cdot e_i \in \mathcal{L}$ for every unit vector e_i with $1 \leq i \leq m$.

However for our cryptographic purposes we will consider q-ary lattices with $q < \det(\mathcal{L})$. In particular we define for a given matrix $A \in \mathbb{Z}_q^{n \times m}$, for some integers n, m, q, two m-dimensional q-ary lattices,

$$\Lambda_q(A) = \{y \in \mathbb{Z}^m \mid y = A^T z \bmod q \text{ for some } z \in \mathbb{Z}^n\}$$
$$\Lambda_q^\perp(A) = \{y \in \mathbb{Z}^m \mid Ay = 0 \bmod q\}.$$

The first q-ary lattice is generated by the rows of A. The second contains all vectors that are orthogonal modulo q to the rows of A. For the reader more familiar with linear codes there is a comparison to be made as this corresponds one-to-one with linear codes in $\mathbb{Z}_q^m$. The first lattice corresponds to the code with generator matrix A and the second corresponds to the code with parity check matrix A.

To show that these two sets are indeed lattices, we will construct a basis for them. For our applications we will generate A uniformly at random with

$m \geq 2n$ and assume q is prime. Then it holds with high probability that A contains n linearly independent columns over $\mathbb{Z}_q$ [13]. Without loss of generality we rearrange A such that the first n columns are linearly independent and we write $A = (A_1 \mid A_2)$ for $A_1 \in \mathbb{Z}_q^{n \times n}$ and $A_2 \in \mathbb{Z}_q^{n \times m-n}$.

Now consider a vector $y \in \Lambda_q(A)$. We can write $y = A^T z = (A_1^T z, A_2^T z)^T = (z', A_2^T (A_1^T)^{-1} z')^T = (z', (A_1^{-1} A_2)^T z')^T$ for $z' = A_1^T z$. Thus the lattice basis of $\Lambda_q(A)$ is given by

$$\begin{pmatrix} I_n & 0 \\ (A_1^{-1} A_2)^T & qI_{m-n} \end{pmatrix}.$$

Next consider a vector $y \in \Lambda_q^{\perp}(A)$, we can write $y = (y_1, y_2)^T$. Then $Ay = A_1 y_1 + A_2 y_2 = 0 \bmod q$. Therefore $y_1 = -A_1^{-1} A_2 y_2 \bmod q$. We obtain $y = (-A_1^{-1} A_2 y_2, y_2)^T \bmod q$. Thus the lattice basis of $\Lambda_q^{\perp}(A)$ is given by

$$\begin{pmatrix} qI_n & -A_1^{-1} A_2 \\ 0 & I_{m-n} \end{pmatrix}.$$

These lattices are dual to each other up to normalization as from the definition it follows $\Lambda_q^{\perp}(A) = q \cdot \Lambda_q(A)^*$ and $\Lambda_q(A) = q \cdot \Lambda_q^{\perp}(A)^*$. We can easily verify this with our bases.

$$q \cdot \left(\begin{pmatrix} I_n & 0 \\ (A_1^{-1} A_2)^T & qI_{m-n} \end{pmatrix}^{-1} \right)^T = q \cdot \begin{pmatrix} I_n & 0 \\ -\frac{1}{q}(A_1^{-1} A_2)^T & \frac{1}{q} I_{m-n} \end{pmatrix}^T = \begin{pmatrix} qI_n & -A_1^{-1} A_2 \\ 0 & I_{m-n} \end{pmatrix},$$

$$q \cdot \left(\begin{pmatrix} qI_n & -A_1^{-1} A_2 \\ 0 & I_{m-n} \end{pmatrix}^{-1} \right)^T = q \cdot \begin{pmatrix} \frac{1}{q} I_n & \frac{1}{q} A_1^{-1} A_2 \\ 0 & I_{m-n} \end{pmatrix}^T = \begin{pmatrix} I_n & 0 \\ (A_1^{-1} A_2)^T & qI_{m-n} \end{pmatrix}.$$

Definition 3.21. A lattice $\mathcal{L}$ is *cyclic* if for all $x = (x_1, ..., x_{n-1}, x_n)^T \in \mathcal{L}$ we have that $(x_n, x_1, ..., x_{n-1})^T \in \mathcal{L}$.

Definition 3.22. [12] The *Gaussian function on* $\mathbb{R}^n$ *centered at* c *with parameter* $s > 0$ is defined by:

$$\forall x \in \mathbb{R}^n, \rho'_{s,c}(x) = \exp(-\pi ||x - c||^2 / s^2).$$

When the subscript c is omitted, the function is taken to be centered around 0. We denote for a subset $S \subset \mathbb{R}^n$, $\rho'_{s,c}(S) = \sum_{x \in S} \rho'_{s,c}(x)$.

For completeness we include the definition of the smoothing parameter for a lattice. The smoothing parameter will be used in Theorem 3.29 to describe a reduction between lattice problems.

Definition 3.23. [12] For any lattice $\mathcal{L}$ and $\epsilon \in \mathbb{R}, \epsilon > 0$, the *smoothing parameter* $\eta_\epsilon(\mathcal{L})$ is the smallest $s \in \mathbb{R}, s > 0$ such that $\rho'_{1/s}(\mathcal{L}^* \backslash \{0\}) \leq \epsilon$.

Lemma 3.24. *For any $\omega(\sqrt{\log n})$ function, there is a negligible $\epsilon(n)$ for which $\eta_\epsilon(\mathcal{L}) \leq \omega(\sqrt{\log n})/\lambda_1^\infty(\mathcal{L}^*)$.*

Proof. Lemma 2.6 in [12]. □

Remark. If $\mathcal{L}$ is the integer lattice $\mathbb{Z}^n$, we have $\lambda_1^\infty(\mathcal{L}^*) = 1$, and obtain $\eta_\epsilon(\mathcal{L}) \leq \omega(\sqrt{\log n})$.

3.3 Lattice problems

Definition 3.25. [14] *Short Integer Solution problem* denoted $\mathrm{SIS}(m, n, q, \beta)$. Let $n \in \mathbb{Z}$ be the primary security parameter, $q, m \in \mathbb{Z}$ such that $q = \mathrm{poly}(n)$, $m = \mathrm{poly}(n)$ and $\beta \in \mathbb{R}, \beta > 0$. Given a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$, the goal is to find a nonzero integer vector $z \in \mathbb{Z}^m \backslash \{0\}$, such that $Az = 0 \bmod q$ and $||z|| < \beta$.

Remark. The SIS problem is to find a short non-zero vector in the lattice $\Lambda_q^\perp(A)$.

β needs to be large enough such that a solution exists. Note that $\beta \geq q$ makes the problem trivial: $(q, 0, ..., 0) \in \mathbb{Z}^m$ would be a solution.

Lemma 3.26. *Let $m > n\frac{\log q}{\log 2}$, $n, q > 1$ and $\beta > \sqrt{m}$. Then there exists a solution to the SIS(m, n, q, β) problem.*

Proof. Let $z \in \{0, 1\}^m$. There are

$$2^m > 2^{\frac{n \log q}{\log 2}} = e^{n \log q} = q^n,$$

such vectors. Then, by the pigeonhole principle, there must exist 2 vectors $z_1 \neq z_2$ such that $Az_1 = Az_2 \bmod q$. Then $z_1 - z_2 \neq 0$ satisfies $A(z_1 - z_2) = 0 \bmod q$ and $||z_1 - z_2|| \leq \sqrt{m} < \beta$. □

Lemma 3.27. *Let $m > n > 1$ and let $q^* \in \mathbb{N}$ be minimal with $q^* > q^{\frac{n}{m}}$. Let $\beta > \sqrt{m} q^*$. Then there exists a solution to the SIS(m, n, q, β) problem.*

Proof. Let $z \in \{0, 1, ..., q^*\}^m$. There are

$$(q^*)^m > (q^{\frac{n}{m}})^m = q^n,$$

such vectors. Then, by the pigeonhole principle, there must exist 2 vectors $z_1 \neq z_2$ such that $Az_1 = Az_2 \bmod q$. Then $z_1 - z_2 \neq 0$ satisfies $A(z_1 - z_2) = 0 \bmod q$ and $||z_1 - z_2|| \leq \sqrt{m}q^* < \beta$. □

Definition 3.28. [14] *Approximate Shortest Independent Vectors Problem (SIVP$_\gamma$).* Let $\mathcal{L}(B)$ be a full rank n-dimensional lattice. For $\gamma = \gamma(n) \in \mathbb{R}$, the SIVP$_\gamma$ problem is to find n linearly independent lattice vectors $b_1, ..., b_n \in \mathcal{L}(B)$ such that $||b_i|| \leq \gamma \cdot \lambda_i(\mathcal{L}(B))$ for all $i \in \{1, ..., n\}$.

Theorem 3.29. *Let n and $m = \text{poly}(n)$ be integers, let $\beta \geq \beta_\infty \geq 1$ be reals, let $Z = \{z \in \mathbb{Z}^m \mid ||z||_2 \leq \beta \wedge ||z||_\infty \leq \beta_\infty\}$ and let $q \geq \beta \cdot n^\delta$ for some constant $\delta > 0$. For some $\gamma = \max\{1, \beta\beta_\infty/q\} \cdot O(\beta\sqrt{n})$, there is an efficient reduction from SIVP$_{\gamma \cdot \omega_n}$ on n-dimensional lattices to solving SIS(m, n, q, β) over Z with non-negligible advantage. Here $\omega_n = \omega(\sqrt{\log n})$ is the smoothing parameter of the integer lattice $\mathbb{Z}^n$.*

Proof. See Theorem 3.8 in [14]. □

Remark. The l_∞ bound on the SIS solutions can be removed by setting $\beta_\infty = \beta$. Then $||z||_\infty \leq ||z|| \leq \beta = \beta_\infty$ automatically holds. We note that the approximating factor γ becomes larger if β_∞ becomes larger. Micciancio and Peikert [14] note this may indicate that restricting the l_∞ norm in addition to the l_2 norm makes the problem qualitatively harder. Furthermore $\beta_\infty \ll \beta$ is natural in the usual formulations of collision-resistant hash functions based on SIS.

Ajtai has shown that worst case SIS problems reduce to the average case [15]. So solving SIS(m, n, q, β) on the average with non-negligible probability is at least as hard as approximating SIVP in the worst case on n-dimensional lattices to within factors $\gamma \cdot \omega_n$, with γ and ω_n as in Theorem 3.29.

Khot showed that SIVP$_{O(1)}$ is NP-hard [16]. So for any constant approximation factor SIVP remains hard. However, we have based the hardness of the SIS problem on the hardness of SIVP$_{\tilde{O}(n^{1.5})}$. The assumption is that it is hard to approximate the Shortest Independent Vector Problem (SIVP) within polynomial approximation factors [17]. However, it has been shown that SIVP$_{\tilde{O}(n)}$ is no longer NP-hard.

Theorem 3.30. *SIVP$_{\tilde{O}(n)}$ is not NP-hard unless P=NP.*

Proof. See [18]. □

Chapter 4

Rényi Divergence

The Rényi divergence was introduced by Rényi [19] as a measure of information that generalizes the Kullback-Leibler divergence, which will be defined later in this chapter. A recent review of the Rényi divergence and Kullback-Leibler divergence is given in [8].

The Rényi divergence can be used as an alternative to the statistical distance as a measure of closeness of distributions. At the end of the chapter we prove the probability preservation property which shows how the Rényi divergence can be used in security proofs of cryptographic protocols. Let P, Q be two probability distributions. If an event A occurs with a specific probability under P, then the Rényi divergence gives us a lower bound of this probability under Q.

The authors of [20] have applied the Rényi divergence to security proofs in lattice-based cryptography. However, they redefine the Rényi divergence as the exponential of the classical definition [8]. In this work we will stick with the classical definition.

The Rényi divergence depends on a parameter that is called its order. We will first differentiate between simple orders and extended orders, as the definition of the Rényi divergence is dependent on this distinction.

Definition 4.1. [8] An $a \in \mathbb{R}$ is a *simple order* if $a > 0$ and $a \neq 1$. We call 0, 1 and ∞ *extended orders*.

In particular we will see later that the Rényi divergence of order 1 equals the Kullback-Leibler divergence.

We are now able to define the Rényi divergence for simple orders.

Definition 4.2. [8] Let P, Q be two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$, with probability mass functions $p(x), q(x)$ respectively. For any simple order a, the Rényi divergence of order a is defined by

$$R_a(P||Q) = \frac{1}{a-1} \log \mathbb{E}_P \left[\left(\frac{p(X)}{q(X)} \right)^{a-1} \right],$$

where $\mathbb{E}_P$ denotes the expectation with respect to probability measure P.

This definition naturally generalizes to continuous distributions.

Definition 4.3. Let P, Q be two probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$, with probability density functions $p(x), q(x)$ respectively. For any simple order a, the Rényi divergence of order a is defined by

$$R_a(P||Q) = \frac{1}{a-1} \log \mathbb{E}_P \left[\left(\frac{p(X)}{q(X)} \right)^{a-1} \right] = \frac{1}{a-1} \log \int \left(\frac{p(x)}{q(x)} \right)^{a-1} dP(x).$$

However, we will only consider discrete distributions for our lattice purposes. The main purpose of this chapter is to provide conceptually easier proofs for the discrete case than the proofs found in the literature [8] for the general case.

Lemma 4.4. *Let P, Q be two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$. For any simple order a, the Rényi divergence of order a is equal to*

$$R_a(P||Q) = \frac{1}{a-1} \log \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a}.$$

Proof. The proof follows directly from the definition of the expected value.

$$\begin{aligned} R_a(P||Q) &= \frac{1}{a-1}\log \mathbb{E}_P\left[(p(X)/q(X))^{a-1}\right] \\ &= \frac{1}{a-1}\log \sum_{x\in \mathrm{supp}(P)} p(x)\left(\frac{p(x)}{q(x)}\right)^{a-1} \\ &= \frac{1}{a-1}\log \sum_{x\in \mathrm{supp}(P)} \frac{p(x)^a}{q(x)^{a-1}} \\ &= \frac{1}{a-1}\log \sum_{x\in \mathrm{supp}(P)} p(x)^a q(x)^{1-a}. \end{aligned}$$

□

Definition 4.5. [8] Let $(\Omega, \mathcal{F})$ be a measurable space. Let P, Q be two discrete probability measures on $(\Omega, \mathcal{F})$ such that $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$. For any sub-$\sigma$-algebra $\mathcal{G} \subset \mathcal{F}$, we denote by $P_{|\mathcal{G}}, Q_{|\mathcal{G}}$ the restriction of P, Q respectively to $\mathcal{G}$.

By considering a coarser σ-algebra we are essentially summing observations together. This should not increase the divergence of P with respect to Q, which is formalized in the following theorem.

Theorem 4.6. *(Data processing inequality) Let $(\Omega, \mathcal{F})$ be a measurable space. Let P, Q be two discrete probability measures such that* $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$*. For any simple order a and any sub-σ-algebra $\mathcal{G} \subset \mathcal{F}$*

$$R_a(P||Q) \geq R_a(P_{|\mathcal{G}}||Q_{|\mathcal{G}}).$$

Proof. See [8]. □

Theorem 4.7. *Let $(\Omega, \mathcal{F})$ be a measurable space. Let P, Q be two discrete probability measures such that* $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$*. For a partition $\mathcal{P} \subset \mathcal{F}$, let $P_{|\sigma(\mathcal{P})}$ and $Q_{|\sigma(\mathcal{P})}$ denote the restrictions of P and Q to the σ-algebra generated by $\mathcal{P}$.*

For any simple order a

$$R_a(P||Q) = \sup_{\mathcal{P}} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})})$$

where the supremum is taken over all finite partitions $\mathcal{P} \subset \mathcal{F}$.

Proof. By Theorem 4.6 we already have

$$R_a(P||Q) \geq \sup_{\mathcal{P}} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})}).$$

Therefore we will now prove the reverse inequality. Let $\mathcal{Q} = \{\{x_1\}, \{x_2\}, ...\}$ be the partition of $\text{supp}(P)$ into singletons. $\mathcal{Q}$ is countable because P is discrete. Let $\mathcal{P}_n = \{\{x_1\}, ..., \{x_{n-1}\}, \bigcup_{i\geq n}\{x_i\}\}$ of size n. Then

$$\begin{aligned}
\lim_{n\to\infty} R_a(P_{|\sigma(\mathcal{P}_n)}||Q_{|\sigma(\mathcal{P}_n)}) &= \lim_{n\to\infty} \frac{1}{a-1} \log \sum_{A\in\mathcal{P}_n} P(A)^a Q(A)^{1-a} \\
&\geq \lim_{n\to\infty} \frac{1}{a-1} \log \sum_{i=1}^{n-1} P(\{x_i\})^a Q(\{x_i\})^{1-a} \\
&= R_a(P_{|\sigma(\mathcal{Q})}||Q_{|\sigma(\mathcal{Q})}) = R_a(P||Q)
\end{aligned}$$

where we have used

$$\begin{aligned}
P(\bigcup_{i\geq n}\{x_i\})^a Q(\bigcup_{i\geq n}\{x_i\})^{1-a} &\geq 0 \qquad\qquad (a > 1), \\
\lim_{n\to\infty} P(\bigcup_{i\geq n}\{x_i\})^a Q(\bigcup_{i\geq n}\{x_i\})^{1-a} &= 0 \qquad\qquad (0 < a < 1).
\end{aligned}$$

Hence

$$R_a(P||Q) \leq \sup_{\mathcal{P}} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})}).$$

□

We will now provide defintions for the Rényi divergence of extended orders. The Rényi divergence of order 0 is defined as the limit of order a approaching 0 from above.

Definition 4.8. [8] For any P, Q two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$,

$$R_0(P||Q) = \lim_{a\downarrow 0} R_a(P||Q).$$

The Rényi divergence of order 1 is defined as the limit of order a approaching 1 from below.

Definition 4.9. [8] For any P, Q two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$,

$$R_1(P||Q) = \lim_{a \uparrow 1} R_a(P||Q).$$

The Rényi divergence of order ∞ is defined as the limit of order a going to ∞.

Definition 4.10. [8] For any P, Q two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$,

$$R_\infty(P||Q) = \lim_{a \uparrow \infty} R_a(P||Q).$$

These limits exist because for $a \in [0, \infty]$ the Rényi divergence $R_a(P||Q)$ is increasing in a, as we will prove in the following theorem.

Theorem 4.11. *For any two discrete probability distributions P, Q such that $\text{supp}(P) \subset \text{supp}(Q)$, the function $f : [0, \infty] \to [0, \infty]$, $a \mapsto R_a(P||Q)$ is increasing.*

Proof. Let $a < b$ be simple orders. For $x \geq 0$ the function $x \mapsto x^{\frac{a-1}{b-1}}$ is concave if $a > 1$ and convex if $a < 1$. In both cases we have by Jensen's inequality

$$\begin{aligned}
R_a(P||Q) &= \frac{1}{a-1} \log \mathbb{E}_P \left[\left(\frac{p(x)}{q(x)} \right)^{a-1} \right] \\
&= \frac{1}{a-1} \log \mathbb{E}_P \left[\left(\frac{p(x)}{q(x)} \right)^{(b-1)\frac{a-1}{b-1}} \right] \\
&\leq \frac{1}{a-1} \log \mathbb{E}_P \left[\left(\frac{p(x)}{q(x)} \right)^{b-1} \right]^{\frac{a-1}{b-1}} \\
&= \frac{1}{b-1} \log \mathbb{E}_P \left[\left(\frac{p(x)}{q(x)} \right)^{b-1} \right] = R_b(P||Q),
\end{aligned}$$

because in the convex case for $a < 1$, we also have $\frac{1}{a-1} < 0$, which reverses the inequality again.

The result for the extended orders follows from the simple orders.

$$\begin{aligned} R_0(P||Q) &= \inf_{0<a<1} R_a(P||Q). \\ R_1(P||Q) &= \sup_{0<a<1} R_a(P||Q) \leq \inf_{a>1} R_a(P||Q). \\ R_\infty(P||Q) &= \sup_{a>1} R_a(P||Q). \end{aligned}$$

□

We will now prove two lemmas we will use to prove results about the Rényi divergence of extended orders and the continuity of the Rényi divergence in its order a. Lemma 4.12 concerns orders between 0 and 1, while Lemma 4.13 provides the same result for orders greater than 1.

Lemma 4.12. *For any sequence $(a_n)_{n=0}^{\infty}$ with $a_n \in (0,1)$ that converges to $a \in [0,1]$*

$$\lim_{n\to\infty} \sum_{x\in\text{supp}(P)} p(x)^{a_n} q(x)^{1-a_n} = \sum_{x\in\text{supp}(P)} \lim_{n\to\infty} p(x)^{a_n} q(x)^{1-a_n}.$$

Proof. For all x and for all $a_n \in (0,1)$

$$\begin{aligned} 0 \leq p(x)^{a_n} q(x)^{1-a_n} &\leq \max\{p(x)^{a_n} p(x)^{1-a_n}, q(x)^{a_n} q(x)^{1-a_n}\} \\ &= \max\{p(x), q(x)\} \\ &\leq p(x) + q(x). \end{aligned}$$

Hence,

$$0 \leq \sum_{x\in\text{supp}(P)} p(x)^{a_n} q(x)^{1-a_n} \leq \sum_{x\in\text{supp}(P)} (p(x) + q(x)) \leq 2.$$

Thus the result follows from Lebesgue's dominated convergence theorem. □

Lemma 4.13. *Let $\mathcal{A} = \{a \mid 1 < a < \infty \text{ and } R_a(P||Q) < \infty\}$. For any sequence $(a_n)_{n=0}^{\infty}$ with $a_n \in \mathcal{A}$ that converges to $a \in \mathcal{A} \cup \{1\}$*

$$\lim_{n\to\infty} \sum_{x\in\text{supp}(P)} p(x)^{a_n} q(x)^{1-a_n} = \sum_{x\in\text{supp}(P)} \lim_{n\to\infty} p(x)^{a_n} q(x)^{1-a_n}.$$

Proof. There exists a real number $b \geq a$ such that $b \in \mathcal{A}$ and $a_n \leq b$ for sufficiently large n. (If $a_n \leq a$ for all n, we can pick $b = a$ and otherwise we can pick $b = a_i$ such that $a_{i+j} \leq a_i$ for all $j > 0$.) By convexity of $p(x)^{a_n} q(x)^{1-a_n}$ in a_n we have for $a_n \leq b$

$$p(x)^{a_n} q(x)^{1-a_n} \leq (1 - \frac{a_n}{b}) p(x)^0 q(x)^1 + \frac{a_n}{b} p(x)^b q(x)^{1-b} \leq q(x) + p(x)^b q(x)^{1-b}.$$

We know

$$\sum_{x \in \mathrm{supp}(P)} q(x) \leq 1,$$
$$\sum_{x \in \mathrm{supp}(P)} p(x)^b q(x)^{1-b} < \infty,$$

because $b > 1$ and $R_b(P||Q) < \infty$.

Thus the result follows from Lebesgue's dominated convergence theorem. □

We are now ready to evaluate the limits in the definitions of the Rényi divergence of extended orders.

Theorem 4.14. *For any two discrete probability distributions P, Q such that* $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$*, the Rényi divergence of order* 0 *is equal to*

$$R_0(P||Q) = -\log \sum_{x \in \mathrm{supp}(P)} q(x).$$

Proof. We have

$$\begin{aligned} R_0(P||Q) &= \lim_{a \downarrow 0} R_a(P||Q) = \lim_{a \downarrow 0} \frac{1}{a-1} \log \sum_{x \in \mathrm{supp}(P)} p(x)^a q(x)^{1-a} \\ &= -\log \lim_{a \downarrow 0} \sum_{x \in \mathrm{supp}(P)} p(x)^a q(x)^{1-a} \\ &= -\log \sum_{x \in \mathrm{supp}(P)} \lim_{a \downarrow 0} p(x)^a q(x)^{1-a} \\ &= -\log \sum_{x \in \mathrm{supp}(P)} \mathbb{1}_{\{p(x)>0\}} q(x) \\ &= -\log \sum_{x \in \mathrm{supp}(P)} q(x), \end{aligned}$$

where we have used Lemma 4.12 to interchange the sum and limit. The last equality follows because we are summing over the support of P. □

Definition 4.15. For any P, Q two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$, the *Kullback-Leibler divergence* is defined by

$$KL(P||Q) = \sum_{x \in \text{supp}(P)} p(x) \log \frac{p(x)}{q(x)}.$$

Theorem 4.16. *For any two discrete probability distributions P, Q such that $\text{supp}(P) \subset \text{supp}(Q)$, the Rényi divergence of order 1 is equal to the Kullback-Leibler divergence.*

Proof. By definition

$$R_1(P||Q) = \lim_{a \uparrow 1} R_a(P||Q) = \lim_{a \uparrow 1} \frac{1}{a-1} \log \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a}.$$

By Lemma 4.12 we can interchange the sum and the limit

$$\lim_{a \uparrow 1} \log \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a} = \log \sum_{x \in \text{supp}(P)} \lim_{a \uparrow 1} p(x)^a q(x)^{1-a} = \log \sum_{x \in \text{supp}(P)} p(x) = 0.$$

Obviously $a - 1$ tends to 0 for a going to 1, so by l'Hôpital's rule it follows

$$R_1(P||Q) = \lim_{a \uparrow 1} \frac{\frac{d}{da} \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a}}{\sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a}} = \lim_{a \uparrow 1} \frac{\sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a} \log(\frac{p(x)}{q(x)})}{\sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a}}.$$

Again by Lemma 4.12 we have

$$\lim_{a \uparrow 1} \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a} = \sum_{x \in \text{supp}(P)} \lim_{a \uparrow 1} p(x)^a q(x)^{1-a} = \sum_{x \in \text{supp}(P)} p(x) = 1.$$

So

$$\begin{aligned} R_1(P||Q) &= \lim_{a \uparrow 1} \sum_{x \in \text{supp}(P)} p(x)^a q(x)^{1-a} \log \left(\frac{p(x)}{q(x)} \right) \\ &= \lim_{a \uparrow 1} \sum_{x \in \text{supp}(P)} \log \left(\frac{p(x)}{q(x)} \right) q(x) \cdot \left(\frac{p(x)}{q(x)} \right)^a. \end{aligned}$$

It is the sum of exponential functions in a, which are convex and increasing. Hence the sum is also convex and increasing in a. Therefore its derivative is

increasing in a and positive, respectively. Then we obtain by the monotone convergence theorem

$$\begin{aligned}\lim_{a\uparrow 1}\sum_{x\in\mathrm{supp}(P)} p(x)^a q(x)^{1-a}\log\left(\frac{p(x)}{q(x)}\right) &= \sum_{x\in\mathrm{supp}(P)}\lim_{a\uparrow 1} p(x)^a q(x)^{1-a}\log\left(\frac{p(x)}{q(x)}\right)\\ &= \sum_{x\in\mathrm{supp}(P)} p(x)\log\left(\frac{p(x)}{q(x)}\right).\end{aligned}$$

Plugging in the results we obtain

$$R_1(P||Q) = \sum_{x\in\mathrm{supp}(P)} p(x)\log\left(\frac{p(x)}{q(x)}\right).$$

□

Theorem 4.17. *Let P, Q be two discrete probability distributions such that* $\mathrm{supp}(P)\subset\mathrm{supp}(Q)$. *If $R_1(P||Q)=\infty$ or there exists a real number $b>1$ such that $R_b(P||Q)$ is finite, then the Rényi divergence of order 1 is equal to*

$$R_1(P||Q) = \lim_{a\downarrow 1} R_a(P||Q).$$

Proof. If $R_1(P||Q)=\infty$, then $R_b(P||Q)\geq R_1(P||Q)=\infty$ for all $b>1$ by Theorem 4.11. So assume there exists a b such that $R_b(P||Q)$ is finite. Then

$$\lim_{a\downarrow 1}\sum_{x\in\mathrm{supp}(P)} p(x)^a q(x)^{1-a}\log\left(\frac{p(x)}{q(x)}\right) \leq R_b(P||Q) < \infty.$$

So we can apply the monotone convergence theorem again analogous to the proof above to obtain

$$\lim_{a\downarrow 1} R_a(P||Q) = \sum_{x\in\mathrm{supp}(P)} p(x)\log\left(\frac{p(x)}{q(x)}\right).$$

□

We will now evaluate the limit in the definition of the Rényi divergence of infinite order. We will first prove the result in the case that probability distribution P has finite support. In the proof we will subsequently reduce the general case to the finite case in order to prove the claim.

Theorem 4.18. *For any two discrete probability distributions P, Q such that* $\operatorname{supp}(P) \subset \operatorname{supp}(Q)$*, the Rényi divergence of order ∞ is equal to*

$$R_\infty(P||Q) = \log \sup_{x \in \operatorname{supp}(P)} \frac{p(x)}{q(x)}.$$

Proof. First assume P has finite support. Then $\max_{x \in \operatorname{supp}(P)} \frac{p(x)}{q(x)}$ exists and is equal to the supremum. Denote $z_\infty = \max_{x \in \operatorname{supp}(P)} \frac{p(x)}{q(x)}$ and let $a > 1$. We have

$$\left[\sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)}{q(x)} \right)^{a-1} \right]^{\frac{1}{a-1}} = z_\infty \left[\sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)/q(x)}{z_\infty} \right)^{a-1} \right]^{\frac{1}{a-1}}.$$

By definition of z_∞

$$\frac{p(x)/q(x)}{z_\infty} \leq 1 \qquad \forall x \in \operatorname{supp}(P)$$

with equality at least once. Let $y \in \operatorname{supp}(P)$ such that $z_\infty = \frac{p(y)}{q(y)}$.

Note

$$p(y) = p(y) \left(\frac{p(y)/q(y)}{z_\infty} \right)^{a-1} \leq \sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)/q(x)}{z_\infty} \right)^{a-1},$$

$$\left[\sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)/q(x)}{z_\infty} \right)^{a-1} \right]^{\frac{1}{a-1}} \leq 1.$$

Then

$$z_\infty \cdot p(y)^{\frac{1}{a-1}} \leq \left[\sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)}{q(x)} \right)^{a-1} \right]^{\frac{1}{a-1}} \leq z_\infty.$$

By the squeeze theorem it follows

$$\lim_{a \to \infty} \left[\sum_{x \in \operatorname{supp}(P)} p(x) \left(\frac{p(x)}{q(x)} \right)^{a-1} \right]^{\frac{1}{a-1}} = z_\infty.$$

So

$$\begin{aligned} R_\infty(P||Q) &= \lim_{a\to\infty} \log \left[\sum_{x\in \mathrm{supp}(P)} p(x) \left(\frac{p(x)}{q(x)} \right)^{a-1} \right]^{\frac{1}{a-1}} \\ &= \log \lim_{a\to\infty} \left[\sum_{x\in \mathrm{supp}(P)} p(x) \left(\frac{p(x)}{q(x)} \right)^{a-1} \right]^{\frac{1}{a-1}} \\ &= \log z_\infty = \log \max_{x\in \mathrm{supp}(P)} \frac{p(x)}{q(x)}. \end{aligned}$$

Now we will reduce the general case to the finite case. Recall $a > 1$,

$$\begin{aligned} R_\infty(P||Q) &= \lim_{a\uparrow\infty} R_a(P||Q) \\ &= \lim_{a\uparrow\infty} \sup_{\mathcal{P}} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})}) && \text{(by Theorem 4.7)} \\ &= \sup_{a<\infty} \sup_{\mathcal{P}} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})}) && \text{(by Theorem 4.11)} \\ &= \sup_{\mathcal{P}} \sup_{a<\infty} R_a(P_{|\sigma(\mathcal{P})}||Q_{|\sigma(\mathcal{P})}) \\ &= \sup_{\mathcal{P}} \log \max_{A\in\mathcal{P}} \frac{P(A)}{Q(A)} && \text{(finite case)} \\ &= \log \sup_{A\subset \mathrm{supp}(P)} \frac{P(A)}{Q(A)}, \end{aligned}$$

where $\mathcal{P}$ ranges over all finite partitions of $\mathrm{supp}(P)$.

It is obvious that

$$\sup_{A\subset \mathrm{supp}(P)} \frac{P(A)}{Q(A)} \geq \sup_{x\in \mathrm{supp}(P)} \frac{p(x)}{q(x)},$$

because for every $x \in \mathrm{supp}(P)$, $\{x\} \subset \mathrm{supp}(P)$. Next, let us show equality holds.

$$\begin{aligned} P(A) &= \sum_{y\in A} p(y) = \sum_{y\in A} \frac{p(y)}{q(y)} q(y) \\ &\leq \sum_{y\in A} \sup_{x\in \mathrm{supp}(P)} \left(\frac{p(x)}{q(x)} \right) q(y) = \sup_{x\in \mathrm{supp}(P)} \left(\frac{p(x)}{q(x)} \right) Q(A). \end{aligned}$$

So for all A,

$$\frac{P(A)}{Q(A)} \leq \sup_{x \in \text{supp}(P)} \frac{p(x)}{q(x)}.$$

That is

$$\sup_{A \subset \text{supp}(P)} \frac{P(A)}{Q(A)} \leq \sup_{x \in \text{supp}(P)} \frac{p(x)}{q(x)}.$$

Thus it follows,

$$\sup_{A \subset \text{supp}(P)} \frac{P(A)}{Q(A)} = \sup_{x \in \text{supp}(P)} \frac{p(x)}{q(x)}.$$

Then we obtain

$$R_\infty(P||Q) = \log \sup_{A \subset \text{supp}(P)} \frac{P(A)}{Q(A)} = \log \sup_{x \in \text{supp}(P)} \frac{p(x)}{q(x)}.$$

□

Theorem 4.19. *Let P, Q be two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$. The Rényi divergence $R_a(P||Q)$ is continuous in a on $\{a \in [0,1]\} \cup \{a \in [0,\infty] \mid R_a(P||Q) < \infty\}$.*

Proof. Continuity at a simple order follows from Lemma 4.12 and 4.13. By definition it is continuous on the extended orders 0 and ∞ and by Theorem 4.16 and 4.17 also at the extended order 1. □

Theorem 4.20. *Let P, Q be two discrete probability distributions such that $\text{supp}(P) \subset \text{supp}(Q)$. For any order $a \in [0,\infty]$*

$$R_a(P||Q) \geq 0.$$

For $a > 0$, $R_a(P||Q) = 0 \iff P = Q$. For $a = 0$, $R_a(P||Q) = 0 \iff \text{supp}(P) = \text{supp}(Q)$.

Proof. First consider $a = 0$.

$$R_0(P||Q) = -\log \sum_{x \in \text{supp}(P)} q(x) \geq 0.$$

where the inequality follows from $\sum_{x \in \text{supp}(P)} q(x) \leq 1$. The inequality holds with equality if and only if $\text{supp}(Q) \subset \text{supp}(P)$, thus $\text{supp}(Q) = \text{supp}(P)$.

Now suppose a is a simple order. By Jensen's inequality

$$\begin{aligned} R_a(P||Q) &= \frac{1}{a-1}\log \mathbb{E}_P\left[\left(\frac{p(x)}{q(x)}\right)^{a-1}\right] \\ &= \frac{1}{a-1}\log \mathbb{E}_P\left[\left(\frac{q(x)}{p(x)}\right)^{1-a}\right] \\ &\geq \frac{1}{a-1}\log\left(\mathbb{E}_P\left[\frac{q(x)}{p(x)}\right]\right)^{1-a} \\ &= -\log \mathbb{E}_P\left[\frac{q(x)}{p(x)}\right] \\ &= -\log \sum_{x\in \mathrm{supp}(P)} p(x)\frac{q(x)}{p(x)} \\ &= -\log \sum_{x\in \mathrm{supp}(P)} q(x) \geq 0. \end{aligned}$$

The second inequality holds with equality if and only if $\mathrm{supp}(Q) \subset \mathrm{supp}(P)$. The first inequality holds with equality if and only if $\frac{q(x)}{p(x)}$ is constant on the support of P. Together this is equivalent to $P = Q$.

By noting that $R_a(P||Q) = \sup_{b<a} R_b(P||Q)$ the result extends to orders $a \in \{1, \infty\}$.

□

Theorem 4.21. *(Data processing inequality) Let P, Q be two discrete probability distributions such that $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$. For any order $a \in [0, \infty]$ and any partition $\mathcal{P} \subset \mathrm{supp}(P)$,*

$$R_a(P||Q) \geq R_a(P_{|\mathcal{P}}||Q_{|\mathcal{P}}).$$

Proof. By Theorem 4.6 it holds for simple orders. Let $a_n \to a$ be any sequence of simple orders that converges to a from above if $a = 0$ and from below if $a \in 1, \infty$. Then

$$R_a(P||Q) = \lim_{a_n \to a} R_{a_n}(P||Q) \geq \lim_{a_n \to a} R_{a_n}(P_{|\mathcal{P}}||Q_{|\mathcal{P}}) = R_a(P_{|\mathcal{P}}||Q_{|\mathcal{P}}).$$

□

Lemma 4.22. *(Probability preservation property) Let P, Q be two discrete probability distributions such that $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$. Let $E \subset \mathrm{supp}(Q)$ be*

an arbitrary event. Then we have for $a \in (1, \infty)$,

$$Q(E) \geq \frac{P(E)^{\frac{a}{a-1}}}{\exp(R_a(P||Q))}.$$

Proof. Hölder's inequality states for $s, t \in (1, \infty)$ with $1/s + 1/t = 1$ we have

$$\sum_{x \in E} |f(x)g(x)| \leq \left(\sum_{x \in E} |f(x)|^s \right)^{\frac{1}{s}} \left(\sum_{x \in E} |g(x)|^t \right)^{\frac{1}{t}}.$$

Take $f(x) = \frac{p(x)}{q(x)^{\frac{1}{t}}}$ and $g(x) = q(x)^{\frac{1}{t}}$ for $\frac{1}{t} = 1 - \frac{1}{s}$. Then

$$\sum_{x \in E} p(x) \leq \left(\sum_{x \in E} \frac{p(x)^s}{q(x)^{s-1}} \right)^{\frac{1}{s}} \left(\sum_{x \in E} q(x) \right)^{1-\frac{1}{s}}.$$

Substitute $s = a$,

$$P(E) \leq \left(\sum_{x \in E} p(x)^a q(x)^{1-a} \right)^{\frac{1}{a}} Q(E)^{1-\frac{1}{a}}.$$

Rearranging gives

$$Q(E) \geq \frac{P(E)^{\frac{a}{a-1}}}{\left(\sum_{x \in E} p(x)^a q(x)^{1-a} \right)^{\frac{1}{a-1}}}.$$

Using

$$\exp(R_a(P||Q)) = \left(\sum_{x \in \mathrm{supp}(P)} p(x)^a q(x)^{1-a} \right)^{\frac{1}{a-1}} \geq \left(\sum_{x \in E} p(x)^a q(x)^{1-a} \right)^{\frac{1}{a-1}},$$

we get the result

$$Q(E) \geq \frac{P(E)^{\frac{a}{a-1}}}{\exp(R_a(P||Q))}.$$

□

The probability preservation property in the case of Rényi divergence of order ∞ follows by a straightforward argument. We provide the proof for completeness.

Lemma 4.23. *(Probability preservation property) Let P, Q be two discrete probability distributions such that* $\mathrm{supp}(P) \subset \mathrm{supp}(Q)$*. Let $E \subset \mathrm{supp}(Q)$ be an arbitrary event. Then we have*

$$Q(E) \geq \frac{P(E)}{\exp(R_\infty(P||Q))}.$$

Proof.

$$\begin{aligned} P(E) &= \sum_{x \in E} p(x) = \sum_{x \in E} \frac{p(x)}{q(x)} q(x) \\ &\leq \sum_{x \in E} \left(\sup_{x \in E} \frac{p(x)}{q(x)} \right) q(x) = \left(\sup_{x \in E} \frac{p(x)}{q(x)} \right) \sum_{x \in E} q(x) = \left(\sup_{x \in E} \frac{p(x)}{q(x)} \right) Q(E) \\ &\leq \left(\sup_{x \in \mathrm{supp}(P)} \frac{p(x)}{q(x)} \right) Q(E) = \exp(R_\infty(P||Q)) Q(E) \end{aligned}$$

It immediately follows

$$Q(E) \geq \frac{P(E)}{\exp(R_\infty(P||Q))}.$$

□

Chapter 5

Digital Signature Scheme

Our main reference this chapter is Lyubashevsky's paper [7]. Let us first in general outline the signature scheme we will discuss in this chapter. Let q be a prime number and k, m, n, κ are parameters of the scheme. The secret key $S \in \mathbb{Z}_q^{m \times k}$ is a matrix with small coefficients and the public key consists of a pair of matrices (A, T), where $A \in \mathbb{Z}_q^{n \times m}$ is chosen uniformly at random and $T = AS \bmod q$. $H : \{0,1\}^* \to \{x \in \{-1,0,1\}^k \mid ||x||_1 \leq \kappa\}$ is a hash function.

Given a message μ, the signing algorithm Σ generates a vector $y \in \mathbb{Z}^m$ according to some distribution D. Next, Σ computes $c = H(Ay \bmod q, \mu)$ and $z = Sc + y \in \mathbb{Z}^m$. Note the last computation is not performed modulo q. S, c and y will be lifted to the integers in the range $[-\frac{q-1}{2}, \frac{q-1}{2}]$ and then z is computed. The result (z, c) will be output as the signature with a certain probability, such that its distribution will be independent of S. Let us first focus on what this probability should be, we will cover the verification algorithm later.

Let Z denote the distribution of $z = Sc + y$ with probability density function g. Here S is the secret key, c is the output of H and y is chosen according to distribution D. The goal is to find a distribution D and a distribution X, independent of S, with probability density function f, such that $\forall x \in \text{supp}(f), f(x) \leq Mg(x)$ for a small constant M.

Then if we sample z from Z and output z as signature with probability $f(z)/(Mg(z))$, the resulting distribution is exactly X. We obtain this result from the well known Rejection Sampling Theorem. We provide a proof for

completeness.

Theorem 5.1. Rejection Sampling Theorem. *Let X be a random variable distributed according to cumulative distribution function F and let Y be a random variable distributed according to cumulative distribution function G. Suppose the probability density functions f, g exist for X, Y, respectively. Let $M \in \mathbb{R}$ be such that $\forall x \in \text{supp}(f), f(x) \leq Mg(x)$. Then generating a sample Y according to G and accepting with probability $f(Y)/(Mg(Y))$ results exactly in a sample X distributed according to F.*

Proof. We will first prove the probability of outputting a sample. Let U denote a standard uniform random variable. Then the probability that the sample Y is accepted is given by

$$\begin{aligned}
\mathbb{P}[Y \text{ is accepted}] &= \mathbb{P}\left[U \leq \frac{f(Y)}{Mg(Y)}\right] = \mathbb{E}\left[\mathbb{P}\left[U \leq \frac{f(Y)}{Mg(Y)} \mid Y\right]\right] \\
&= \mathbb{E}\left[\frac{f(Y)}{Mg(Y)}\right] = \int_{-\infty}^{\infty} \frac{f(y)}{Mg(y)} g(y) dt \\
&= \frac{1}{M} \int_{-\infty}^{\infty} f(y) dt = \frac{1}{M}.
\end{aligned}$$

Next, let us show the sample Y has the desired distribution. We must show that Y conditioned on being accepted is distributed according to cumulative distribution function F.

$$\begin{aligned}
\mathbb{P}[Y \leq y \mid Y \text{ is accepted}] &= \frac{\mathbb{P}\left[Y \text{ is accepted} \mid Y \leq y\right] \cdot \mathbb{P}[Y \leq y]}{\mathbb{P}[Y \text{ is accepted}]} \\
&= \mathbb{P}\left[Y \text{ is accepted} \mid Y \leq y\right] \cdot \frac{\mathbb{P}[Y \leq y]}{1/M} \\
&= \frac{\mathbb{P}\left[Y \text{ is accepted} \cap Y \leq y\right]}{\mathbb{P}[Y \leq y]} \cdot \frac{\mathbb{P}[Y \leq y]}{1/M} \\
&= M \cdot \mathbb{P}\left[Y \text{ is accepted} \cap Y \leq y\right] \\
&= M \cdot \int_{-\infty}^{y} \mathbb{P}[Y \text{ is accepted} \mid Y = t] g(t) dt \\
&= M \cdot \int_{-\infty}^{y} \frac{f(t)}{Mg(t)} g(t) dt \\
&= \int_{-\infty}^{y} f(t) dt = F(y).
\end{aligned}$$

This completes the proof. □

Remark. The discrete case of the Rejection Sampling Theorem is analogous to the continuous case when we replace the probability density function by the probability mass function. In the proof we can replace the integrals by sums.

Lemma 5.2. *The rejection sampling described in Theorem 5.1 requires an expected number of M samples from Y in order to output a sample from X.*

Proof. The rejection sampling outputs a sample from X with probability $1/M$. Thus the number of samples required from Y follows a geometric distribution with probability $1/M$, which has mean M. □

5.1 Uniform distribution

In this section we will first explore the option where the vector y is sampled according to a uniform distribution to familiarize ourselves with the signature scheme. In the next section we will sample the vector y from a Gaussian distribution in order to obtain better (practical) performance results.

We pick $A \in \mathbb{Z}_q^{n \times m}$ uniformly at random, pick $S \in \{-b, ..., 0, ..., b\}^{m \times k}$ uniformly at random for some small $b \in \mathbb{N}$ and set $T = AS \bmod q$. We have $c \in \{x \in \{-1, 0, 1\}^k \mid ||x||_1 \leq \kappa\}$. Then we set $R = \sqrt{m \cdot (\kappa b)^2} = \kappa b \sqrt{m}$ so that $\forall S \in \{-b, ..., 0, ..., b\}^{m \times k}$ and $\forall c \in \{x \in \{-1, 0, 1\}^k \mid ||x||_1 \leq \kappa\}$ we have $||Sc|| \leq R$.

Choose y uniformly from an m-dimensional discrete ball $\{x \in \mathbb{Z}^m \mid ||x|| \leq r + R\}$ of radius $r + R$, where r is some real number and R is an upper bound on the length of Sc as defined above.

We compute $c = H(Ay \bmod q, \mu)$ for a hash function H and $z = Sc + y$. For X we choose the uniform distribution over an m-dimensional discrete ball $\{x \in \mathbb{Z}^m \mid ||x|| \leq r\}$ of radius r. We will output (z, c) as the signature if $z \in \{x \in \mathbb{Z}^m \mid ||x|| \leq r\}$.

We set

$$M = \frac{|\{x \in \mathbb{Z}^m \mid ||x|| \leq r + R\}|}{|\{x \in \mathbb{Z}^m \mid ||x|| \leq r\}|}.$$

Then by the rejection sampling theorem (Theorem 5.1), z conditioned on being output, is exactly distributed as X.

We want to ensure M is small so we can sign a message efficiently. We approximate M by ratio of the volume of two (continuous) balls.

$$M \approx \frac{\mathrm{vol}(\mathcal{B}(r+R))}{\mathrm{vol}(\mathcal{B}(r))} = \frac{\frac{\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2}+1)}(r+R)^m}{\frac{\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2}+1)}r^m} = \left(\frac{r+R}{r}\right)^m = \left(1+\frac{1}{r/R}\right)^m.$$

If $r/R \geq m$, we have:

$$M \approx \left(1+\frac{1}{r/R}\right)^m \leq \left(1+\frac{1}{m}\right)^m \leq e.$$

Therefore we have to pick $r \geq mR = \tilde{O}(m^{1.5})$ to ensure M is small.

A verifier accepts a signature $\sigma = (z, c)$ on a message μ if $c = H(Az - Tc \bmod q, \mu)$ and $||z|| \leq r$.

Lemma 5.3. *A valid signature (z, c) is accepted by the verifier.*

Proof. We have $z \in \mathcal{B}(r)$, so $||z|| \leq r$ and

$$\begin{aligned} H(Az - Tc \bmod q, \mu) &= H(A(Sc+y) - ASc \bmod q, \mu) \\ &= H(Ay \bmod q, \mu) = c \end{aligned}$$

□

Lemma 5.4. *A valid signature (z, c) is zero-knowledge in the random oracle model.*

Proof. Because of the rejection sampling step, the signature z is distributed as X, which is independent of the secret key. So to simulate a signature, generate a sample z from X and a uniformly random c and output (z, c). To make the signature valid, program the oracle H as follows: if $H(Az - Tc \bmod q, \mu)$ already has a value assigned not equal to c, start over; else program H such that $c = H(Az - Tc \bmod q, \mu)$. □

We will now prove a lemma that we use in the security proof to replace a column of secret key S.

Lemma 5.5. *Let $l \in \mathbb{Z}$. For any $A \in \mathbb{Z}_q^{n \times m}$ with $m > l \cdot \frac{\log 2}{\log(2b+1)} + n \cdot \frac{\log q}{\log(2b+1)}$ and $s \in \{-b, ..., 0, ..., b\}^m$ chosen uniformly random, there exists with probability at least $1 - 2^{-l}$ an $s' \in \{-b, ..., 0, ..., b\}^m$, $s' \neq s$ such that $As = As'$.*

Proof. The range of the corresponding matrix transformation of A has size q^n, so there can be at most q^n elements $s \in \{-b, ..., 0, ..., b\}^m$ that do not collide. Thus the probability of picking such an element is at most

$$\frac{q^n}{(2b+1)^m} = \frac{e^{n \log q}}{e^{m \log(2b+1)}} \leq e^{n \log q - n \log q - l \log 2} = e^{-l \log 2} = 2^{-l}.$$

□

Theorem 5.6. *Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most s sign queries to the signer Σ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability ϵ. Moreover, assume the image of the random oracle H is large such that H is pre-image and collision resistant. Then there exists an algorithm of the same time complexity that can solve the $SIS(m, n, q, \beta)$ problem for $\beta = 2r + 2b\kappa\sqrt{m} = \tilde{O}(n^{1.5})$ with probability $\approx \frac{\epsilon^2}{2(s+h)}$.*

Proof. Let A be the matrix in the SIS instance and run the forger $\mathcal{F}$ on public key (A, T) for $T = AS$ with $S \in \{-b, ..., 0, ..., b\}^{m \times k}$ uniformly at random for some small $b \in \mathbb{N}$. Let $t = s + h$ and pick $r_1, ..., r_t$ uniformly random from $\{x \in \{-1, 0, 1\}^k \mid ||x||_1 \leq \kappa\}$, i.e. the image of the random oracle H which we will denote by $\mathrm{Im}(H)$. Whenever $\mathcal{F}$ queries H or $\mathcal{F}$ requests a signature and Σ queries H, the response of H will be the first r_i that has not yet been used. If the query has been made before, H will reply with the same r_j. So in particular H keeps a list of queries and outputs. Note we simulate Σ to generate a signature as described in the proof of Lemma 5.4.

$\mathcal{F}$ will output, with probability ϵ, a message μ and its signature (z, c) such that $||z|| \leq r$ and $c = H(Az - Tc \bmod q, \mu)$. If $\mathcal{F}$ or Σ has not queried H on $(Az - Tc \bmod q, \mu)$, $\mathcal{F}$ cannot know $H(Az - Tc \bmod q, \mu)$ and could at best obtain $c = H(Az - Tc \bmod q, \mu)$ with probability $1/|\mathrm{Im}(H)|$. Therefore the probability that $\mathcal{F}$ outputs a forgery (z, c) with c being one of the r_i's is at least $\epsilon - 1/|\mathrm{Im}(H)|$. Let j be such that $c = r_j$.

Now r_j was a response to a random oracle query made by $\mathcal{F}$ or it was programmed during signing by Σ.

Suppose $c = H(Az' - Tc \bmod q, \mu')$ was programmed during the signing of a message μ' by Σ. If $\mathcal{F}$ produces a valid forgery for the message μ, we have $H(Az' - Tc \bmod q, \mu') = H(Az - Tc \bmod q, \mu)$. So if $Az' - Tc \neq$

$Az - Tc \bmod q$ or $\mu' \neq \mu$, then $\mathcal{F}$ has found a second preimage of the random oracle H, which is negligible given our assumption on H. Therefore we can assume $Az' - Tc = Az - Tc \bmod q$ and $\mu' = \mu$. Then $Az' - Az = 0 \bmod q$, thus $A(z' - z) = 0 \bmod q$. We know $z' \neq z$, because $\mu' = \mu$ and $\mathcal{F}$ output a forgery. Therefore $z' - z \neq 0$ and $||z' - z|| \leq ||z'|| + ||z|| \leq r + r = 2r$.

Suppose $c = r_j$ was set as a response to a random oracle query made by $\mathcal{F}$. We store the signature (z, c), rewind $\mathcal{F}$ to after the selection of message μ and regenerate random elements $r'_j, .., r'_t$ uniformly random from the range of the random oracle H. By the General Forking Lemma [21] we obtain that the probability that $\mathcal{F}$ forges again on r'_j with $r'_j \neq r_j$ is at least

$$(\epsilon - 1/|\operatorname{Im}(H)|)\left(\frac{\epsilon - 1/|\operatorname{Im}(H)|}{t} - 1/|\operatorname{Im}(H)|\right) \approx \frac{\epsilon^2}{t}.$$

Thus we obtain a signature (z', c') with $c' = r'_j$ on the message μ with the above probability. $Az - Tc = Ay = Az' - Tc' \bmod q$, so we obtain:

$$A\left(z - z' + S(c' - c)\right) = 0 \bmod q.$$

$||z||, ||z'|| \leq r$, $Sc, Sc' \leq b\kappa\sqrt{m}$, so $||z - z' + S(c' - c)|| \leq 2r + 2b\kappa\sqrt{m}$.

It remains to be shown $z - z' + S(c' - c) \neq 0$. We know $c \neq c'$, so let i be a position such that $c_i \neq c'_i$. By Lemma 5.5 it follows that with probability $1 - 2^{-l}$ we can replace column S_i of S by another vector S'_i with $AS = AS'$, where S' is the matrix S with replaced column i. If $z - z' + S(c' - c) = 0$, then $z - z' + S'(c' - c) \neq 0$. By Lemma 5.4 $\mathcal{F}$ cannot know whether we are using secret key S or S'. Both are equally likely, so we get $z - z' + S(c' - c) \neq 0$ with probability at least $\frac{1}{2}$. □

5.2 Gaussian distribution

In the previous section we saw the signature vectors z have length $\tilde{O}(m^{1.5})$. In this section we obtain signature vectors z with length $\sigma\sqrt{m} = \tilde{O}(m)$ by replacing the uniform distribution with the discrete Gaussian distribution.

Definition 5.7. [7] The *Gaussian distribution on* $\mathbb{R}^n$ *centered at* $v \in \mathbb{R}^n$ *with standard deviation* $\sigma \in \mathbb{R}_{>0}$ is defined by:

$$\forall x \in \mathbb{R}^n, \rho^n_{v,\sigma}(x) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \exp\left(\frac{-||x - v||^2}{2\sigma^2}\right).$$

When the subscript v is omitted, the function is taken to be centered around 0.

Definition 5.8. [7] For any mean $v \in \mathbb{Z}^n$, parameter $\sigma \in \mathbb{R}_{\geq 0}$, we define the *discrete Gaussian distribution* over $\mathbb{Z}^n$ as:

$$\forall x \in \mathbb{Z}^n, D^n_{v,\sigma}(x) = \frac{\rho^n_{v,\sigma}(x)}{\rho^n_{v,\sigma}(\mathbb{Z}^n)} = \frac{\rho^n_{v,\sigma}(x)}{\sum_{x\in\mathbb{Z}^n} \rho^n_{v,\sigma}(x)}.$$

Note that $\rho^n_{v,\sigma}(\mathbb{Z}^n) = \rho^n_{\sigma}(\mathbb{Z}^n)$ for all $v \in \mathbb{Z}^n$. Thus the normalization factor is the same for all v. Moreover note that σ is a parameter close to the standard deviation of the distribution, but not equal [22].

Remark. Some authors define the discrete Gaussian distribution over $\mathbb{Z}^n$ with the Gaussian function $\rho'_{s,c}$ defined in definition 3.22. This is equivalent for $s = \sigma\sqrt{2\pi}$.

Recall we choose $A \in \mathbb{Z}_q^{n\times m}$ uniformly at random, choose $S \in \{-b, ..., 0, ..., b\}^{m\times k}$ uniformly at random for some small $b \in \mathbb{N}$, set $T = AS \bmod q$ and the image of the hash function $\mathrm{Im}(H) = \{x \in \{-1, 0, 1\}^k \mid ||x||_1 \leq \kappa\}$.

Choose y from the discrete Gaussian distribution D^m_σ over $\mathbb{Z}^m$ centered around 0 for some parameter σ, compute $c = H(Ay \bmod q, \mu)$ and compute $z = Sc + y$. We will output (z, c) as the signature with probability $\min\left(\frac{D^m_\sigma(z)}{MD^m_{v,\sigma}(z)}, 1\right)$ where $v = Sc$ and M has yet be determined to match criteria outlined in the following theorem (Theorem 5.9).

A signature will be accepted if $||z|| \leq 2\sigma\sqrt{m}$ and $c = H(Az - Tc \bmod q, \mu)$.

This signature scheme with rejection sampling that is not perfect is similar to Lyubashevsky's scheme [7], but instead we will use the Rényi divergence in the security proof. We will show later that this Rényi divergence analysis leads to better parameters for the signature scheme.

Theorem 5.9. *Let X be a discrete random variable on $\mathbb{Z}^m$ with probability mass function f. Let h specify the uniform probability mass function on an arbitrary countable set V. Let g_v be a family of probability mass functions on $\mathbb{Z}^m$ indexed with $v \in V$. Let $M, \epsilon \in \mathbb{R}$ be such that for all v, $\Pr[f(X) \geq Mg_v(X)] \leq \epsilon$.*

Then the output distribution of the following algorithm $\mathcal{S}_1$:

1. *Sample v according to h.*

2. *Sample z according to g_v.*

3. *With probability $\min\left(f(z)/(Mg_v(z)), 1\right)$ output (z, v). Else return to step 1.*

is within $\frac{\epsilon}{1-\epsilon}$ Rényi divergence of order $a \in [0, \infty]$ with respect to the output distribution of algorithm $\mathcal{S}_2$:

1. *Sample v according to h.*

2. *Sample z according to f.*

3. *Output (z, v).*

Moreover, the probability that $\mathcal{S}_1$ terminates during the current iteration is at least $(1-\epsilon)/M$.

Proof. Let $E_v = \{z \in \mathbb{Z}^m \mid f(z) \leq Mg_v(z)\}$ be the set of points on which f is dominated by Mg_v. We will write $E_v^C = \mathbb{Z}^m \backslash E_v$ for the complement of E_v in $\mathbb{Z}^m$. If $z \in E_v$, then $\min\left(f(z)/(Mg_v(z)), 1\right) = f(z)/(Mg_v(z))$ and if $z \in E_v^C$, then $\min\left(f(z)/(Mg_v(z)), 1\right) = 1$. We now determine upper and lower bounds on the probability that $\mathcal{S}_1$ terminates in the current iteration. We denote this event by $\mathcal{T}$.

$$\begin{aligned}
\mathbb{P}[\mathcal{T}] &= \sum_{v \in V} h(v) \left(\sum_{z \in E_v} g_v(z) \frac{f(z)}{Mg_v(z)} + \sum_{z \in E_v^C} g_v(z) \right) \\
&\geq \sum_{v \in V} h(v) \sum_{z \in E_v} \frac{f(z)}{M} \geq \frac{1-\epsilon}{M}, \\
\mathbb{P}[\mathcal{T}] &= \sum_{v \in V} h(v) \left(\sum_{z \in E_v} g_v(z) \frac{f(z)}{Mg_v(z)} + \sum_{z \in E_v^C} g_v(z) \right) \\
&\leq \sum_{v \in V} h(v) \left(\sum_{z \in E_v} \frac{f(z)}{M} + \sum_{z \in E_v^C} \frac{f(z)}{M} \right) = \frac{1}{M}.
\end{aligned}$$

In order to compute the Rényi divergence, we must first know for all $z \in \mathbb{Z}^m$, $v \in V$ what the probability is of $\mathcal{S}_1$ and $\mathcal{S}_2$ outputting (z, v). In the case of $\mathcal{S}_2$ this probability is $h(v)f(z)$. In the case of $\mathcal{S}_1$ we are required to do some more work. The probability that $\mathcal{S}_1$ outputs a sample (z, v) in iteration i conditioned on not having output a sample in iteration $1, ..., i-1$

is $h(v)g_v(z)\min(f(z)/(Mg_v(z)),1)$. It is thus clear that the probability of outputting a sample in the current iteration does not depend on the iteration number, nor on the previous iterations. Therefore we can define $N_{\mathcal{S}_1}$ as the probability that $\mathcal{S}_1$ does not output a sample (z,v) during the current iteration and compute $(N_{\mathcal{S}_1})^i$ as the probability that $\mathcal{S}_1$ has failed to produce output after i iterations. Then by the law of total probability we obtain

$$\mathbb{P}[\mathcal{S}_1 \text{ outputs } (z,v)] = \sum_{i=0}^{\infty}\left((N_{\mathcal{S}_1})^i \cdot h(v)g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)\right).$$

Now to compute the Rényi divergence of order ∞ we use Theorem 4.18 to get

$$\begin{aligned}
R_\infty(\mathcal{S}_1||\mathcal{S}_2) &= \log\sup_{(z,v)}\frac{\sum_{i=0}^{\infty}\left((N_{\mathcal{S}_1})^i h(v)g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)\right)}{h(v)f(z)}\\
&= \log\sup_{(z,v)}\frac{g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)\cdot\sum_{i=0}^{\infty}(N_{\mathcal{S}_1})^i}{f(z)}\\
&= \log\frac{1}{1-N_{\mathcal{S}_1}}\sup_{(z,v)}\frac{g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)}{f(z)}.
\end{aligned}$$

Let us analyze the supremum.

$$\begin{aligned}
&\sup_{(z,v)}\frac{g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)}{f(z)}\\
&= \max\left(\sup_{(z,v),z\in E_v}\frac{g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)}{f(z)}, \sup_{(z,v),z\notin E_v}\frac{g_v(z)\min(\frac{f(z)}{Mg_v(z)},1)}{f(z)}\right)\\
&= \max\left(\sup_{(z,v),z\in E_v}\frac{g_v(z)\frac{f(z)}{Mg_v(z)}}{f(z)}, \sup_{(z,v),z\notin E_v}\frac{g_v(z)}{f(z)}\right)\\
&= \max\left(\frac{1}{M}, \sup_{(z,v),z\notin E_v}\frac{g_v(z)}{f(z)}\right)\\
&= \frac{1}{M}.
\end{aligned}$$

So

$$R_\infty(\mathcal{S}_1||\mathcal{S}_2) = \log\frac{1}{M(1-N_{\mathcal{S}_1})}.$$

And

$$1 = \frac{1}{M(1-(1-\frac{1}{M}))} \leq \frac{1}{M(1-N_{\mathcal{S}_1})} \leq \frac{1}{M(1-(1-\frac{1-\epsilon}{M}))} = 1 + \frac{\epsilon}{1-\epsilon}.$$

Thus,

$$R_\infty(\mathcal{S}_1||\mathcal{S}_2) = \log \frac{1}{M(1-N_{\mathcal{S}_1})} \leq \log\left(1 + \frac{\epsilon}{1-\epsilon}\right) \leq \frac{\epsilon}{1-\epsilon}.$$

By Theorem 4.11 the result follows for the Rényi divergence of all orders $a \in [0, \infty]$. □

Remark. Multiple runs of the algorithms are independent, so running n times would give

$$R_\infty(\mathcal{S}_1^n||\mathcal{S}_2^n) = \log\left[\left(\frac{1}{M(1-N_{\mathcal{S}_1})}\right)^n\right] = n \cdot R_\infty(\mathcal{S}_1||\mathcal{S}_2).$$

Compared to [7] it should be noted that in addition to performing a Rényi divergence analysis we have also chosen to iterate algorithm $\mathcal{S}_1$ until it produces output as opposed to letting $\mathcal{S}_2$ output a sample with probability $1/M$. This in turn will make the security analysis easier, as we do not have to consider a distinguisher based on whether a sample was output or not, and produce better results.

We will use the following lemmas from [7] to bound probabilities regarding events of a random variable distributed according to the discrete Gaussian distribution.

Lemma 5.10. *Let Z be a random variable distributed according to the discrete Gaussian distribution D_σ over $\mathbb{Z}$. Then for any $j > 0$,*

$$\mathbb{P}[|Z| > j\sigma] \leq 2e^{\frac{-j^2}{2}} = 2^{\frac{-j^2}{2\log 2}+1}.$$

Proof. See Lemma 4.4 in full version of [7]. □

Lemma 5.11. *Let Z be a random variable distributed according to the discrete Gaussian distribution D_σ^m over $\mathbb{Z}^m$. Then for any $j > 0$,*

$$\mathbb{P}[||Z|| > j\sigma\sqrt{m}] < j^m e^{\frac{m}{2}(1-j^2)}.$$

Proof. See Lemma 4.4 in full version of [7]. □

Corollary 5.12. *Let Z be a random variable distributed according to the discrete Gaussian distribution D_σ^m over $\mathbb{Z}^m$. Then,*

$$\mathbb{P}[||Z|| > 2\sigma\sqrt{m}] < 2^{-m}.$$

Proof. Apply Lemma 5.11 and simplify $j = 2$. □

Lemma 5.13. *Let Z be a random variable distributed according to the discrete Gaussian distribution D_σ^m over $\mathbb{Z}^m$. Then for any vector $v \in \mathbb{R}^m$ and any $\sigma, r > 0$,*

$$\mathbb{P}[|\langle Z, v\rangle| > r] \leq 2e^{-\frac{r^2}{2||v||^2\sigma^2}}.$$

Proof. See Lemma 4.3 in full version of [7]. □

We are now ready to show the existence of a constant $M \in \mathbb{R}$ as in Theorem 5.9 where $\mathcal{S}_1$ and $\mathcal{S}_2$ sample according to discrete Gaussian distributions. Hence we prove a bound on the Rényi divergence of $\mathcal{S}_1$ from $\mathcal{S}_2$ by applying Theorem 5.9.

Theorem 5.14. *Let X be a discrete Gaussian distributed random variable centered around 0 for some standard deviation parameter $\sigma = \alpha R$ with probability density function D_σ^m for $\alpha, R > 0$. Let h specify the uniform probability distribution on $\mathrm{Im}(H)$, $S \in \{-b, ..., 0, ..., b\}^{m\times k}$ and $R = \max_{c\in\mathrm{Im}(H)} ||Sc||$. Let $D_{v,\sigma}^m$ be a family of probability density functions of the discrete Gaussian distribution for parameter σ indexed with mean $v \in \{Sc \in \mathbb{Z}^m \mid c \in \mathrm{Im}(H)\}$. Then for all $\epsilon > 0$ there exists a constant $M \in \mathbb{R}$ such that for all $v \in \{Sc \in \mathbb{Z}^m \mid c \in \mathrm{Im}(H)\}$ we have $\Pr[D_\sigma^m(X) \geq M D_{v,\sigma}^m(X)] \leq \epsilon$.*

Then the output distribution of the following algorithm $\mathcal{S}_1$:

1. *Sample c according to h.*
2. *Set $v = Sc$.*
3. *Sample z according to $D_{v,\sigma}^m$.*
4. *With probability $\min\left(D_\sigma(z)/(MD_{v,\sigma}(z)), 1\right)$ output (z, c) and program $H(Az - Tc \bmod q, \mu) = c$. Else return to step 1.*

is within Rényi divergence $\frac{\epsilon}{1-\epsilon}$ with respect to the output distribution of $\mathcal{S}_2$:

1. *Sample c according to h.*
2. *Sample z according to D_σ^m.*

3. *Output* (z, c) *and program* $H(Az - Tc \bmod q, \mu) = c$.

Moreover, the probability that $\mathcal{S}_1$ *terminates during the current iteration is at least* $(1 - \epsilon)/M$.

Proof. Let α be such that $\sigma = \alpha R$. We have

$$\begin{aligned}\frac{D_\sigma^m(z)}{D_{v,\sigma}^m(z)} = \frac{\rho_\sigma^m(z)}{\rho_{v,\sigma}^m(z)} &= \frac{\exp\left(\frac{-||z||^2}{2\sigma^2}\right)}{\exp\left(\frac{-||z-v||^2}{2\sigma^2}\right)} \\ &= \frac{\exp\left(\frac{-||z||^2}{2\sigma^2}\right)}{\exp\left(\frac{-(||z||^2+||v||^2-2\langle z,v\rangle)}{2\sigma^2}\right)} \\ &= \exp\left(\frac{||v||^2}{2\sigma^2}\right)\exp\left(\frac{-\langle z,v\rangle}{\sigma^2}\right).\end{aligned}$$

By Lemma 5.13 applied with $r = j\sigma||v||$ we have with probability less than $\epsilon = 2e^{\frac{-j^2}{2}} = 2^{\frac{-j^2}{2\log 2}+1}$ that $|\langle z, v\rangle| > j\sigma||v||$. Therefore with probability at least $1 - \epsilon$,

$$\frac{D_\sigma^m(z)}{D_{v,\sigma}^m(z)} = \exp\left(\frac{||v||^2}{2\sigma^2}\right)\exp\left(\frac{-\langle z,v\rangle}{\sigma^2}\right) < \exp\left(\frac{||v||^2}{2\sigma^2}\right)\exp\left(\frac{j\sigma||v||}{\sigma^2}\right).$$

Recall $\sigma = \alpha R \geq \alpha||v||$, so

$$\begin{aligned}\frac{D_\sigma^m(z)}{D_{v,\sigma}^m(z)} &< \exp\left(\frac{||v||^2}{2\sigma^2}\right)\exp\left(\frac{j||v||}{\sigma}\right) \\ &\leq \exp\left(\frac{||v||^2}{2\alpha^2||v||^2}\right)\exp\left(\frac{j||v||}{\alpha||v||}\right) \\ &= \exp\left(\frac{1}{2\alpha^2}\right)\exp\left(\frac{j}{\alpha}\right) = \exp\left(\frac{j}{\alpha}+\frac{1}{2\alpha^2}\right).\end{aligned}$$

So one can pick $M = \exp\left(\frac{j}{\alpha} + \frac{1}{2\alpha^2}\right)$.

Apply Theorem 5.9 with $f = D_\sigma^m$ and $g_v = D_{v,\sigma}^m$ to complete the proof. □

Lemma 5.15. *A valid signature* (z, c) *is accepted by the verifier with overwhelming probability.*

Proof. By the probability preservation property (Lemma 4.23) and Corollary 5.12 we know

$$\mathbb{P}[||z|| > 2\sigma\sqrt{m}] \leq \left(1 + \frac{\epsilon}{1-\epsilon}\right) 2^{-m},$$

which is negligible, and it always holds

$$\begin{aligned} H(Az - Tc \bmod q, \mu) &= H(A(Sc + y) - ASc \bmod q, \mu) \\ &= H(Ay \bmod q, \mu) = c. \end{aligned}$$

□

Lemma 5.16. *Let $q \geq 2m$ and $m \geq 2n$. Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most s sign queries to the signer Σ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability τ. Then $\mathcal{F}$ succeeds in producing a forgery with probability at least $\tau - s(s+h)2^{-n+1}$ when Σ is replaced by $\mathcal{S}_1$ from Theorem 5.14.*

Proof. Recall the signer Σ:

1. Sample y according to D_σ^m.
2. Set $c = H(Ay \bmod q, \mu)$.
3. Set $z = Sc + y$.
4. With probability $\min\left(D_\sigma(z)/(MD_{v,\sigma}(z)), 1\right)$ output (z, c). Else return to step 1.

The "simulator" $\mathcal{S}_1$ from Theorem 5.14 is equivalent to:

1. Sample y according to D_σ^m.
2. Sample c according to h.
3. Set $z = Sc + y$.
4. With probability $\min\left(D_\sigma(z)/(MD_{v,\sigma}(z)), 1\right)$ output (z, c) and Program $H(Az - Tc \bmod q, \mu) = c$. Else return to step 1.

We note that in the random oracle model the only difference between the actual signing algorithm Σ and "simulator" $\mathcal{S}_1$ is that in $\mathcal{S}_1$ it is not checked whether the value for $H(Ay \bmod q, \mu) = H(Az - Tc \bmod q, \mu)$ was already set.

Since $\mathcal{F}$ makes at most s sign queries and h hash queries, at most $(s+h)$ values of (Ay, μ) will be set. Lemma 5.3 in [7] shows that for a random sample y from D_σ^m

$$\mathbb{P}[Ay = t] \leq 2^{-n+1}$$

for any $t \in \mathbb{Z}_q^n$. Then the probability of a collision each time is at most $(s+h)2^{-n+1}$. So the probability of a collision after s queries is at most $s(s+h)2^{-n+1}$. □

Remark. Note n will be set ≥ 384. Thus this probability is small enough for our purposes.

Recall from Theorem 5.14 we have $M = \exp(\frac{j}{\alpha} + \frac{1}{2\alpha^2})$ and $\epsilon = 2\exp(\frac{-j^2}{2})$. We would like to pick α small because α determines the signature size. Moreover, we would like M to be small for efficiency, because M is the expected number of times we need to generate a sample to produce a signature. So in order for M to be small, we can pick $\alpha = j$. In section 5.5 we will explore the option to pick α small.

Lyubashevsky [7] requires $\epsilon < 2^{-100}$, because he bounds the statistical distance between $\mathcal{S}_1$ and $\mathcal{S}_2$ by $s \cdot \epsilon$, where s is the number of signatures the forger can query. Therefore he must pick $j = 12$ and consequently $\sigma = 12R$. Using the Rényi divergence, we will show that $j = 9$ and thus $\sigma = 9R$ suffices for up to 2^{56} signature queries.

Lemma 5.17. *Let $\mathcal{S}_1$, $\mathcal{S}_2$ be as in Theorem 5.14. Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most $s \leq 2^{56}$ sign queries to the signer using $\mathcal{S}_1$ with $\epsilon = \frac{2}{\exp(81/2)}$ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability τ. Then $\mathcal{F}$ succeeds in producing a forgery with probability $2\tau/3$ when $\mathcal{S}_1$ is replaced by $\mathcal{S}_2$.*

Proof. Let E_s be the collection of sets of s signatures for which an attacker outputs a valid signature. We write $\mathcal{S}_i^s$ to denote the distribution of s signatures sampled independently from $\mathcal{S}_i$. Then by the probability preservation property (Lemma 4.23)

$$\mathcal{S}_2^s(E_s) \geq \mathcal{S}_1^s(E_s)/\exp(R_\infty(\mathcal{S}_1^s||\mathcal{S}_2^s)) = \mathcal{S}_1^s(E_s)/\exp(s \cdot R_\infty(\mathcal{S}_1||\mathcal{S}_2)),$$

where the equality follows per the remark in Theorem 5.9. By Theorem 5.14

$$\exp(s \cdot R_\infty(\mathcal{S}_1||\mathcal{S}_2)) \leq \exp\left(s \cdot \frac{\epsilon}{1-\epsilon}\right) \leq \exp(0.4) \leq 1.5,$$

using $\epsilon = \frac{2}{\exp(81/2)}$ and $s \leq 2^{56}$. □

Theorem 5.18. *Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most s sign queries to the simulator $\mathcal{S}_2$ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability τ. Moreover, assume the image of the random oracle H is large such that H is pre-image and collision resistant. Then there exists an algorithm of the same time complexity that can solve the $SIS(m, n, q, \beta)$ problem for $\beta = (4\sigma + 2b\kappa)\sqrt{m} = \tilde{O}(n)$ with probability $\approx \frac{\tau^2}{2(s+h)}$.*

Proof. Let A be the matrix in the SIS instance and run the forger $\mathcal{F}$ on public key (A, T) for $T = AS$ with $S \in \{-b, ..., 0, ..., b\}^{m \times k}$ uniformly at random for some small $b \in \mathbb{N}$. Let $t = s + h$ and pick $r_1, ..., r_t$ uniformly random from $\mathrm{Im}(H)$. Whenever $\mathcal{F}$ queries H or $\mathcal{F}$ requests a signature and $\mathcal{S}_2$ programs H, the response of H will be the first r_i that has not yet been used. If the query has been made before, H will reply with the same r_j. So in particular H keeps a list of queries and outputs.

$\mathcal{F}$ will output, with probability τ, a message μ and its signature (z, c) such that $||z|| \leq 2\sigma\sqrt{m}$ and $c = H(Az - Tc \bmod q, \mu)$. If $\mathcal{F}$ or $\mathcal{S}_2$ has not queried H on $(Az - Tc \bmod q, \mu)$, $\mathcal{F}$ cannot know $H(Az - Tc \bmod q, \mu)$ and could at best obtain $c = H(Az - Tc \bmod q, \mu)$ with probability $1/|\mathrm{Im}(H)|$. Therefore the probability that $\mathcal{F}$ outputs a forgery (z, c) with c being one of the r_i's is at least $\tau - 1/|\mathrm{Im}(H)|$. Let j be such that $c = r_j$.

Now r_j was a response to a random oracle query made by $\mathcal{F}$ or it was programmed during signing by $\mathcal{S}_2$.

Suppose $c = H(Az' - Tc \bmod q, \mu')$ was programmed during the signing of a message μ' by $\mathcal{S}_2$. If $\mathcal{F}$ produces a valid forgery for the message μ, we have $H(Az' - Tc \bmod q, \mu') = H(Az - Tc \bmod q, \mu)$. So if $Az' - Tc \neq Az - Tc \bmod q$ or $\mu' \neq \mu$, then $\mathcal{F}$ has found a second preimage of the random oracle H, which is negligible given our assumption on H. Therefore we can assume $Az' - Tc = Az - Tc \bmod q$ and $\mu' = \mu$. Then $Az' - Az = 0 \bmod q$, thus $A(z' - z) = 0 \bmod q$. We know $z' \neq z$, because $\mu' = \mu$ and $\mathcal{F}$ output a forgery. Therefore $z' - z \neq 0$ and $||z' - z|| \leq ||z'|| + ||z|| \leq 2\sigma\sqrt{m} + 2\sigma\sqrt{m} = 4\sigma\sqrt{m}$.

Suppose $c = r_j$ was set as a response to a random oracle query made by $\mathcal{F}$. We store the signature (z, c), rewind $\mathcal{F}$ to after the selection of message μ and regenerate random elements $r'_j, .., r'_t$ uniformly random from the range

of the random oracle H. By the General Forking Lemma [21] we obtain that the probability that $\mathcal{F}$ forges again on r'_j with $r'_j \neq r_j$ is at least

$$(\tau - 1/|\operatorname{Im}(H)|)\left(\frac{\tau - 1/|\operatorname{Im}(H)|}{t} - 1/|\operatorname{Im}(H)|\right) \approx \frac{\tau^2}{t}.$$

Thus we obtain a signature (z', c') with $c' = r'_j$ on the message μ with the above probability. $Az - Tc = Ay = Az' - Tc' \bmod q$, so we obtain:

$$A\left(z - z' + S(c' - c)\right) = 0 \bmod q.$$

$||z||, ||z'|| \leq 2\sigma\sqrt{m}$, $||Sc||, ||Sc'|| \leq b\kappa\sqrt{m}$, so $||z - z' + S(c' - c)|| \leq (4\sigma + 2b\kappa)\sqrt{m}$.

It remains to be shown $z - z' + S(c' - c) \neq 0$. We know $c \neq c'$, so let i be a position such that $c_i \neq c'_i$. By Lemma 5.5 it follows that with probability $1 - 2^{-l}$ we can replace column S_i of S by another vector S'_i with $AS = AS'$, where S' is the matrix S with replaced column i. If $z - z' + S(c' - c) = 0$, then $z - z' + S'(c' - c) \neq 0$. Because $\mathcal{S}_2$ does not use the secret to generate signatures, $\mathcal{F}$ cannot know whether we are "using" secret key S or S'. Both are equally likely, so we get $z - z' + S(c' - c) \neq 0$ with probability at least $\frac{1}{2}$. □

We wish to be able to replace a column of S with overwhelming probability, so we set $l = 100$ in Lemma 5.5.

Corollary 5.19. *Let $m \geq \max(2n, (100 + n\log_2 q)/\log_2(2b + 1))$ and $q \geq 2m$. Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most $s \leq 2^{56}$ sign queries to the signer Σ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability τ. Then there exists an algorithm of the same time complexity that can solve the $SIS(m, n, q, \beta)$ problem for $\beta = (4\sigma + 2b\kappa)\sqrt{m} = \tilde{O}(n)$ with probability $\approx \frac{2\tau^2}{9(s+h)}$.*

Proof. Apply Lemma 5.16, Lemma 5.17 and Theorem 5.18 in sequence. □

5.3 Parameter selection

Setting the parameters is difficult for lattice-based cryptography as it is unclear how the running time of reduction algorithms depends on the dimension of the lattice. In order to do so we rely on the experiments made

by Gama and Nguyen [23]. They observed that the shortest length of a vector that can be found for m-dimensional lattices with the best known algorithms is about $\det(\mathcal{L})^{1/m}\delta^m$. The parameter δ depends on the algorithm used.

Even though their experiments were performed on a different distribution of lattices, Micciancio and Regev [13] observed the same behaviour in their experiments on random q-ary lattices, with the exception that the trivial vector of length q can always be found. Therefore the shortest length vector of the SIS problem that can be found is about $\min(q, \det(\Lambda_q^\perp(A))^{1/m}\delta^m)$.

We have $m \geq 2n$ and q prime, so with high probability the matrix A has n linearly independent columns over $\mathbb{Z}_q$. We have seen in section 3.2 that the lattice basis of $\Lambda_q^\perp(A)$ is given by

$$\begin{pmatrix} qI_n & -A_1^{-1}A_2 \\ 0 & I_{m-n} \end{pmatrix}.$$

The determinant of this upper triangular matrix is then given by the product of the diagonal entries, so we easily obtain $\det(\Lambda_q^\perp(A)) = q^n$. Therefore the shortest length of a vector of the SIS problem that can be found becomes $\min(q, q^{n/m}\delta^m)$. Next Micciancio and Regev reason that increasing m cannot make the problem harder, as we can always delete some columns from A, effectively setting some coordinates of y to 0. For large m the high dimension prevents lattice reduction algorithms to find short vectors, in such cases it is thus better to delete some columns of A in order to reduce the dimension. Micciancio and Regev show the optimal $m = \sqrt{n \log q / \log \delta}$ and obtain $\min(q, 2^{2\sqrt{n \log_2 q \log_2 \delta}})$ for the length of the shortest vector that can be found.

They note that slower algorithms can provide $\delta \approx 1.012$ or even $\delta \approx 1.011$. However when the dimension of the lattice becomes several hundreds, it becomes unavoidable to use faster algorithms and δ becomes about 1.013. In any case it seems to be impossible to reach values lower than 1.01 with our current understanding of lattice reductions [13].

We will use this analysis to estimate the shortest vector that can be found for the SIS problem and then set the parameters of the signature scheme such that we can extract a shorter vector for the SIS problem from the forger. To determine the security level, Lyubashevsky [7] sets $\delta = 1.007$ and that offers a large margin to the current 1.013. It is difficult to predict what δ can be achieved in the future, but fortunately for signature schemes long term

security is less important than it is for encryption schemes. In any case we will use the same δ in order to make a fair comparison.

We will take a look at the parameters of the signature scheme of Lyubashevsky based on the hardness of the SIS(m, n, q, β) problem and compare this to what we can achieve based on this same hardness assumption. This concerns columns 1, 2 and 3 of figure 2 in [7], which we replicate in Table 5.1.

	I	II	III	III*
n	512	512	512	512
q	2^{27}	2^{25}	2^{33}	2^{33}
b	1	1	31	31
k	80	512	512	512
$m \approx 64 + n\frac{\log_2 q}{\log_2(2b+1)}$	8786	8139	3253	2891
κ s.t. $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$	28	14	14	14
$\sigma \approx 12 \cdot b \cdot \kappa\sqrt{m}$	31 495	15 157	300 926	280 024
sig size (bits) $\approx m \cdot \log_2(12\sigma)$	163 000	142 300	73 000	62 678
sk size (bits) $\approx m \cdot k \cdot \log_2(2b+1)$	2^{20}	$2^{22.5}$	2^{23}	2^{23}
pk size (bits) $\approx n \cdot k \cdot \log_2(q)$	2^{20}	$2^{22.5}$	2^{23}	2^{23}
shortest $\|\|v\|\|$ $\approx \min\left(q, 2^{2\sqrt{n \log_2 q \log_2 \delta}}\right)$	12 625 688	6 810 608	70 914 144	70 914 144
forger $\|\|v\|\|$ $\approx (4\sigma + 2b\kappa)\sqrt{m}$	11 813 633	5 471 934	67 816 002	60 271 983
ratio	0.94	0.80	0.96	0.85

Table 5.1: **Signature Scheme Parameters.** The parameters are based on the hardness of SIS(m, n, q, β) with $\beta = (4\sigma + 2b\kappa)\sqrt{m}$. The security level is for $\delta = 1.007$. Columns I-III correspond to Fig. 2 in [7].

Note the m in column I and II perfectly follow the formula, but the m in column III seems to be set arbitrarily larger than necessary for the assumptions. We have appended column III* with the same parameters n, q, b, k and $m = 64 + 512 \cdot 33/\log_2(63) = 2891$. This m is still much larger than the optimal $m = \sqrt{512 \cdot 33/\log_2(1.007)} = 1296$ for the lattice reduction algorithms according to Micciancio and Regev [13].

The secret key is an $m \times k$ matrix with entries bounded by b, so it will require $m \cdot k \cdot \lceil \log_2(2b+1) \rceil$ bits. The part of the public key that is individual for each user (T) is an $n \times k$ matrix, so it will require $n \cdot k \cdot \lceil \log_2(q) \rceil$ bits. The size of the signature is dominated by z, as c is just a low weight bit string. The vector z is distributed according to D_σ^m and by Lemma 5.10 we know that each entry of z has length at most 12σ with probability $1 - 2^{-100}$. Thus z can be represented by $m \cdot \lceil \log_2(12\sigma) \rceil$ bits.

We now determine some improved parameters, based on our security proof using the Rényi divergence instead of the statistical distance.

The reduction of σ going from $12R$ to $9R$ barely has a direct influence on the signature length in bits, each vector entry would be at most 1 bit shorter. However, because we can reduce σ by 25%, the forger will also have to find a vector z with length that is 25% smaller. This makes the problem significantly harder. We note the signature size depends mainly on m, therefore we aim to lower the dimension. The smaller vector z required allows us to maintain the same hardness while lowering the dimension. We want to be able to extract a vector from the forger that is smaller than we can expect to obtain from a lattice reduction algorithm with $\delta = 1.007$.

We observe we can gain the most from our reduction of σ from $12R$ to $9R$ when σ is large, in other words, when b is large. In particular a larger b means that the shortest vector we can extract from the forger becomes larger. For security we still require that this extracted vector is smaller than the shortest vector that can be found for the SIS problem directly. Therefore we also set q much higher to increase the length of the shortest vector that can be found for the SIS problem per the analysis in [13]. Then in table 5.2 we see in column VI we can halve the secret and public key size while maintaining the same signature length as in column III*. Alternatively, in column VII we decrease the signature length by about 9,000 bits or approximately 15%.

5.4 SIS decision problem

We have seen that the signature size is approximately $m \cdot \log_2(12\sigma)$ and hence it is most affected by the parameter m. Moreover the chosen values of m are well above the optimal value for m as shown by Micciancio and Regev [13] and thus do not make the problem any harder. Therefore it appears

	VI	VII
n	384	384
q	2^{40}	2^{50}
b	18	181
k	320	320
$m \approx (100 + n \log_2 q)/\log_2(2b+1)$	2969	2270
κ s.t. $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$	15	15
$\sigma \approx 9 \cdot b \cdot \kappa\sqrt{m}$	132 408	1 164 194
sig size (bits) $\approx m \cdot \log_2(12\sigma)$	61 161	53 881
sk size (bits) $\approx m \cdot k \cdot \log_2(2b+1)$	2^{22}	$2^{22.5}$
pk size (bits) $\approx n \cdot k \cdot \log_2(q)$	2^{22}	$2^{22.5}$
shortest $\|\|v\|\| \approx \min(q, 2^{2\sqrt{n \log_2 q \log_2 \delta}})$	30 575 959	233 834 913
forger $\|\|v\|\| \approx (4\sigma + 2b\kappa)\sqrt{m}$	28 888 104	222 128 510
ratio	0.94	0.95

Table 5.2: **Signature Scheme Parameters.** The parameters are based on the hardness of SIS(m, n, q, β) with $\beta = (4\sigma + 2b\kappa)\sqrt{m}$. The security level is for $\delta = 1.007$.

this higher value of m is redundant in some sense. Indeed, Lyubashevsky [7] shows we can use $m = 2n$.

If we still were to satisfy the equation in Lemma 5.5 with $l = 100$, we would need to pick a very large b. This would make the problem a lot easier, since larger vectors will now suffice for the forger to break the signature scheme. Therefore we will no longer require the equation in Lemma 5.5 to hold. Hence we cannot be sure there exists a second secret key $S' \neq S$ that satifies $AS' = T$. In fact, with extremely high probability there will only be one S for which $AS = T$ [7]. However, Theorem 5.14 and Lemma 5.16 still hold. So the real signer is still indistinguishable from the simulator $\mathcal{S}_2$.

Then, for a given A, we can use an S with small coefficients bounded by b in the actual signature, but an S' with large coefficients bounded by b' for the simulator such that there exists an $S'' \neq S'$ with $AS' = AS''$. If the distribution of the public key (A, AS) is computationally indistinguishable from the distribution of (A, AS'), the forger will not be able to tell he is given an invalid key pair. Since $\mathcal{S}_2$ never uses the secret key to generate signatures, the forger will not behave any differently in that case.

The security of the signature scheme is now based on the hardness of both

the original $SIS(m, n, q, \beta)$ problem with $\beta = (4\sigma + 2b'\kappa)\sqrt{m}$ and the distinguishing problem. In order to formalize the hardness of the distinguishing problem we will define the $SIS(m, n, q, b)$ distribution and the $SIS(m, n, q, b)$ decision problem.

Definition 5.20. [7] Choose a uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$ and a vector $s \in \{-b, ..., 0, ..., b\}^m$. Then the pair (A, As) is generated from the $SIS(m, n, q, b)$ *distribution.*

Definition 5.21. [7] *Short Integer Solution decision problem* denoted by SIS(m, n, q, b). Given a pair (A, t), decide with non-negligible advantage whether it is generated from the $SIS(m, n, q, b)$ distribution or generated uniformly at random from $\mathbb{Z}_q^{n\times m} \times Z_q^n$.

The hardness of the distinguishing problem is based on the hardness of the SIS(m, n, q, b) decision problem and the SIS(m, n, q, b') decision problem. Certainly if one cannot even distinguish either from the uniformly random distribution, then one also cannot tell them apart. See Lemma 6.1 in [7] for a formal treatment.

In fact, the hardness of the distinguishing problem is solely based on the hardness of the SIS(m, n, q, b) decision problem as this problem reduces to the SIS(m, n, q, b') decision problem for appropriately chosen b'. It is quite intuitive the problem becomes harder when b increases as the $SIS(m, n, q, b)$ distribution will be statistically close to the uniformly random distribution for very large b.

Lemma 5.22. *[7] For any $\theta \in \mathbb{N}$ such that* $\gcd(2\theta + 1, q) = 1$*, there is a polynomial time reduction from the SIS(m, n, q, b) decision problem to the SIS$(m, n, q, (2\theta + 1)b + \theta)$ decision problem*

Proof. Given (A, t), generate a uniformly random vector $r \in \{-\theta, ..., 0, ..., \theta\}^m$ and output $(A, (2\theta + 1)t + Ar)$. Since $\gcd(2\theta + 1, q) = 1$, the uniform distribution will be mapped to itself. If (A, t) came from the SIS(m, n, q, b) distribution, then $(2\theta + 1)t + Ar = A((2\theta + 1)s + r)$. Vector s was chosen uniformly random from $\{-b, ..., 0, ..., b\}^m$, so $(2\theta + 1)s + r$ is uniformly random in $\{-(2\theta + 1)b - \theta, ..., 0, ..., (2\theta + 1)b + \theta\}^m$. Thus the transformation maps the SIS(m, n, q, b) distribution to the SIS$(m, n, q, (2\theta + 1)b + \theta)$ distribution and the uniform distribution over $\mathbb{Z}^{n\times m} \times \mathbb{Z}^n$ to itself. □

Moreover, the SIS(m, n, q, b) decision problem reduces to the SIS(m, n, q, β)

problem and we end up with only our original computational hardness assumption, for different parameters.

Lemma 5.23. *If $m = 2n$ and $4b\beta < q$, then there is a polynomial time reduction from the SIS(m, n, q, b) decision problem to the SIS(m, n, q, β) problem.*

Proof. Lemma 3.7 in [7]. □

Corollary 5.24. *Let $m = 2n$, $m \geq (100 + n \log_2 q)/\log_2(2b' + 1)$, $q \geq 2m$, $q > 4b\beta$ and $b' = (2\theta + 1)b + \theta$ for some $\theta \in \mathbb{N}$. Suppose there exists a probabilistic polynomial time forger $\mathcal{F}$ in the random oracle model who makes at most $s \leq 2^{56}$ sign queries to the signer Σ and at most h hash queries to the random oracle and succeeds in producing a forgery with probability δ. Then there exists an algorithm of the same time complexity that can solve the $SIS(m, n, q, \beta)$ problem for $\beta = (4\sigma + 2b'k)\sqrt{m} = \tilde{O}(n)$ with probability $\approx \frac{2\delta^2}{9(s+h)}$.*

Proof. The proof is the same as Corollary 5.19 with b replaced by b'. □

Earlier we remarked our analysis has a bigger impact for larger bounds b. For the setting in this section picking bound $b = 1$ is optimal, hence we notice our Rényi divergence analysis now has a lesser impact on the signature size. Moreover, since b, κ and m are fixed, the only possibility to change the signature size is by adjusting the dimension n. As we can extract slightly shorter vectors from the forger, we are able to lower the dimension n a bit further while maintaining security. This allows us a decrease in the signature length by about 2,000 bits or approximately 12%. In table 5.3 we compare column IV from Figure 2 in [7] versus these suggested parameters in column VIII.

5.5 Beating the MTU

The internet protocol (IP) provides for transmitting blocks of data called packets from sources to destinations. The maximum sized package that can be transmitted through the network is called the maximum transmission unit (MTU) [24]. Almost all IP over ethernet implementations use Ethernet v2 and the MTU for Ethernet v2 is 1500 bytes [25]. Therefore it would be especially interesting if we can further reduce our signature size from 14400

	IV	VIII	IX
n	512	464	452
q	2^{24}	2^{25}	2^{22}
b	1	1	1
b'	2206	3124	1108
k	512	448	448
$m = 2n$	1024	928	904
κ s.t. $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$	14	14	14
$\sigma_{SD} \approx 12 \cdot b \cdot \kappa\sqrt{m}$	5376	-	-
$\sigma_{RD} \approx 9 \cdot b \cdot \kappa\sqrt{m}$	-	3839	-
$\sigma_{RD} \approx 1.96 \cdot b \cdot \kappa\sqrt{m}$	-	-	825
M	2.7	2.7	112.4
sig size (bits) $\approx m \cdot \log_2(12\sigma)$	16 500	14 377	11 999
sk size (bits) $\approx m \cdot k \cdot \log_2(2b+1)$	$2^{19.5}$	$2^{19.3}$	$2^{19.3}$
pk size (bits) $\approx n \cdot k \cdot \log_2(q)$	$2^{22.5}$	$2^{22.3}$	$2^{22.1}$
shortest $\|\|v\|\| \approx \min(q, 2^{2\sqrt{n \log_2 q \log_2 \delta}})$	4 955 956	3 198 889	1 053 914
forger $\|\|v\|\| \approx (4\sigma + 2b'\kappa)\sqrt{m}$	2 664 704	3 132 380	1 031 958
ratio	0.54	0.98	0.98

Table 5.3: **Signature Scheme Parameters.** The parameters are based on the hardness of SIS(m, n, q, β) with $\beta = (4\sigma + 2b'\kappa)\sqrt{m}$. The security level is for $\delta = 1.007$. Column IV corresponds to Fig. 2 in [7].

bits (1800 bytes) to 12000 bits (1500 bytes) in order to avoid negative effects such as fragmentation of packets.

The approach we follow will be well suited in cases where the signer is willing to perform some extra work in order to generate smaller signatures. An example could be a signature on software that is distributed to a large number of users or an X.509 certificate .

Recall from Theorem 5.14 we had $M = \exp(\frac{9}{\alpha} + \frac{1}{2\alpha^2})$, where M is the expected number of repetitions in the rejection sampling. In order for M to be small, we picked $\alpha = 9$. This resulted in $\sigma = \alpha R = 9R$. We will now relax the requirement on M to be small and select $\alpha = 1.96$. Therefore we will obtain a smaller $\sigma = 1.96R$ at the cost of a larger $M = 112.4$. This allows us to dive below the 12000 bits limit, while maintaining the same level of security. See column IX in table 5.3 for a detailed parameter description.

In this chapter we have seen a significant improvement in the signature sizes for the desired security level compared to the original signature scheme in [7], further advancing towards practical applicability of lattice-based signature schemes without compromising on the security assumptions.

Chapter 6

BLISS

In this chapter we discuss an alternative approach to improve Lyubashevsky's [7] signature scheme due to Ducas, Durmus, Lepoint and Lyubashevsky [9]. By using samples from a bimodal Gaussian distribution they are able to reduce the standard deviation parameter of the resulting signatures even further. The lattice-based digital signature scheme they construct is named BLISS (Bimodal Lattice Signature Scheme).

6.1 Signature generation

Recall in Lyubashevsky's original scheme [7] a potential signature z was generated by sampling y from the discrete Gaussian distribution D_σ^m over $\mathbb{Z}^m$ centered around 0 for some parameter σ, computing $c = H(Ay \bmod q, \mu)$ and this resulted in $z = Sc + y$. Therefore z is effectively sampled from the discrete Gaussian distribution $D_{v,\sigma}^m$ over $\mathbb{Z}^m$ centered around $v = Sc$ for some parameter σ. Now the authors of BLISS [9] suggested to compute a potential signature as $z = (-1)^a Sc + y$, where a is chosen uniformly random from $\{0, 1\}$. Hence z is now effectively sampled according to the probability distribution function $\frac{1}{2}D_{v,\sigma}^m + \frac{1}{2}D_{-v,\sigma}^m$, where $v = Sc$.

A benefit from using the bimodal Gaussian distribution is the fact that the authors are able to produce exactly the desired distribution with rejection sampling instead of a distribution close to this desired distribution. If we denote the target distribution by f and the bimodal Gaussian distribution by g, there will now exist an M such that $f(x) \leq M \cdot g(x)$ for all x, as opposed

to the case in the original scheme where g was the Gaussian distribution and $f(x) \leq M \cdot g(x)$ held for all values but a fraction with probability measure ϵ.

The main performance gain stems from the fact that the bimodal Gaussian distribution fits the target distribution much better. Keeping M constant, the bimodal Gaussian distribution is able to cover the target distribution completely for a much smaller value of the standard deviation parameter. This will already become clear in the one-dimensional example as seen in figures 6.1 and 6.2.

To perform the rejection sampling in Lyubashevksky's original scheme we required $D_\sigma^m(z) \leq M \cdot D_{v,\sigma}^m(z)$ for all but a negligible fraction of z measured according to D_σ^m. Recall we have computed

$$\frac{D_\sigma^m(z)}{D_{v,\sigma}^m(z)} \leq \exp\left(\frac{j}{\alpha} + \frac{1}{2\alpha^2}\right),$$

where we used $\sigma = \alpha R \geq \alpha||v||$. So we have $M = \exp(\frac{j}{\alpha} + \frac{1}{2\alpha^2})$. Lyubashevsky required $j = 12$ and we required $j = 9$. In both cases we picked $\alpha = j$ in order to obtain $M \approx \exp(1)$. This resulted in $\sigma = 12R$ and $\sigma = 9R$, respectively.

Let us now compute

$$\begin{aligned}
\frac{D_\sigma^m(z)}{\frac{1}{2}D_{v,\sigma}^m(z) + \frac{1}{2}D_{-v,\sigma}^m(z)} &= \frac{\rho_\sigma^m(z)}{\frac{1}{2}\rho_{v,\sigma}^m(z) + \frac{1}{2}\rho_{-v,\sigma}^m(z)} \\
&= \frac{\exp\left(\frac{-||z||^2}{2\sigma^2}\right)}{\frac{1}{2}\exp\left(\frac{-||z-v||^2}{2\sigma^2}\right) + \frac{1}{2}\exp\left(\frac{-||z+v||^2}{2\sigma^2}\right)} \\
&= \frac{\exp\left(\frac{-||z||^2}{2\sigma^2}\right)}{\frac{1}{2}\exp\left(\frac{-||z||^2}{2\sigma^2}\right)\exp\left(\frac{-||v||^2}{2\sigma^2}\right)\left(\exp\left(\frac{\langle z,v\rangle}{\sigma^2}\right) + \exp\left(\frac{-\langle z,v\rangle}{\sigma^2}\right)\right)} \\
&= \frac{1}{\frac{1}{2}\exp\left(\frac{-||v||^2}{2\sigma^2}\right)\left(\exp\left(\frac{\langle z,v\rangle}{\sigma^2}\right) + \exp\left(\frac{-\langle z,v\rangle}{\sigma^2}\right)\right)}.
\end{aligned}$$

We note

$$\frac{1}{2}\left(\exp\left(\frac{\langle z,v\rangle}{\sigma^2}\right) + \exp\left(\frac{-\langle z,v\rangle}{\sigma^2}\right)\right) \geq 1,$$

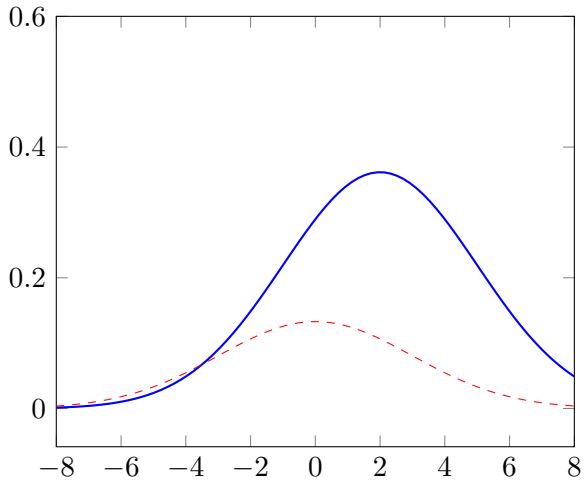

Figure 6.1: Example of Lyubashevsky's original scheme with source distribution (thick, blue) and target distribution (dashed, red)

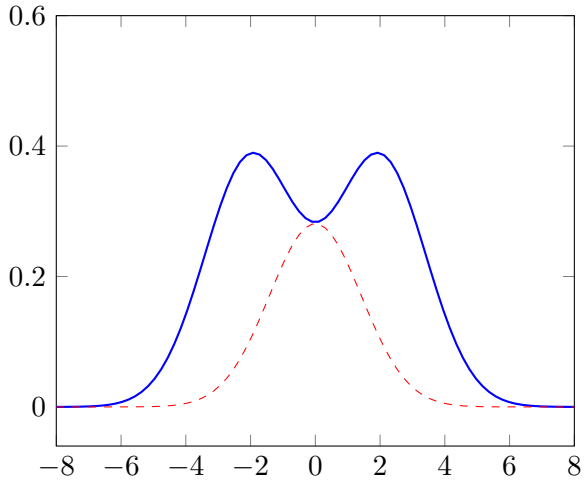

Figure 6.2: Example of BLISS with source distribution (thick, blue) and target distribution (dashed, red)

Therefore we can conclude

$$\frac{D_{\sigma}^{m}(z)}{\frac{1}{2}D_{v,\sigma}^{m}(z)+\frac{1}{2}D_{-v,\sigma}^{m}(z)} \leq \exp\left(\frac{||v||^2}{2\sigma^2}\right) \leq \exp\left(\frac{1}{2\alpha^2}\right),$$

where we again use $\sigma = \alpha R \geq \alpha||v||$.

We now have $M = \exp\left(\frac{1}{2\alpha^2}\right)$ and we are free to choose α. In order to give a fair comparison we will choose α such that $M = \exp(1)$, i.e. $\alpha = \frac{1}{\sqrt{2}}$. This implies $\sigma = \frac{1}{\sqrt{2}}R$, in other words we can choose a much smaller standard deviation parameter compared to $12R$ or even $9R$, which in turn leads to smaller signatures.

The parameter α can be easily increased to favour speed or decreased to favour signature size. In section 5.5 we set $\alpha = 1.96$ in order to obtain signatures sizes below the MTU limit. For BLISS we would in fact be increasing α to 1.96 and hence can expect a speed up compared to the default $M = \exp(1)$. Indeed for $\alpha = 1.96$ the expected number of repetitions $M = 1.14$ is less than 2.72 and a huge performance increase compared to $M = 112.4$ in section 5.5.

Alternatively we could decrease α even further and set $\alpha = 0.5$ in order to obtain even smaller signature sizes. With $\sigma = 0.5R$ we would have $M = 7.39$, which is still reasonable.

6.2 Signature verification

While the approach in section 6.1 clearly generates shorter signatures and is faster, it does break the verification procedure. Therefore it is not as obvious that this technique leads to an improved signature scheme. The verification algorithm checks whether $c = H(Az - Tc \bmod q, \mu)$. This will only hold if $Ay = Az - Tc \bmod q$, hence if $Ay = A\left((-1)^a Sc + y\right) - Tc \bmod q$. Rearranging these terms gives $(-1)^a Tc = Tc \bmod q$. This equation has to hold for $a \in \{0, 1\}$, so we require $-Tc = Tc \bmod q$. However, this never happens for $T \neq 0$ and $q > 2$ prime.

The authors of BLISS [9] propose to work modulo $2q$ and to set $T = qI_n$, where I_n is the $n \times n$ identity matrix, so in particular they set $k = n$. In this case it will hold $-Tc = Tc \bmod 2q$.

However it is no longer obvious how to perform the key generation such that we obtain $AS = qI_n$. The authors of BLISS [9] propose the following

to generate the public and secret keys. First they pick a uniformly random matrix $A' \in \mathbb{Z}_q^{n\times(m-n)}$ and $S' \in \{-n, ..., 0, ..., b\}^{(m-n)\times n}$ for some small $b \in \mathbb{N}$. Compute $A'' = A'S' \bmod q$ and set $A = [2A'|2A'' + qI]$ as public key. The secret key is $S = [S'| - I]^T$ and consists of small entries. Then by construction we have $AS = qI_n \bmod 2q$.

Of course the security proof will also need to be adapted, but it is still very similar to the original security proof in [7]. The dimensions m and n are picked such that the distribution of $[A'|A'' \bmod q]$ is uniformly random in $\mathbb{Z}_q^{n\times m}$. In the security proof we are given a random $B = [A'|A''] \in \mathbb{Z}_q^{n\times m}$ and the authors [9] show that, using a successful forger, we can find a short vector v such that $Bv = 0 \bmod q$, i.e. solve the SIS problem, which we will describe in more detail in a moment. There are however some more technical requirements to ensure the proof works. In particular it is required that for a valid signature z we have $||z||_\infty < q/4$, in addition to the requirement $||z|| \leq 2\sigma\sqrt{m}$. The authors of BLISS note that this condition on the l_∞-norm is usually verified whenever the condition on the l_2-norm is and that it does not restrict them in the manner in which they choose the parameters for the signature scheme [9]. Moreover, as we will see later in the proof, it is now required that $c \neq c'$ implies $c \neq c' \bmod 2$, because in the signature scheme we have modulus $2q$ while in the security proof we would like to reduce to the SIS problem with modulus q prime. Therefore the challenge vector c can no longer be chosen $c \in \{x \in \{-1, 0, 1\}^k \mid ||x||_1 \leq \kappa\}$, but we must choose $c \in \{x \in \{0, 1\}^k \mid ||x||_1 \leq \kappa\}$. In particular this means κ must be set higher to maintain the same level of entropy for the challenge vector. This in turn results in a larger bound $R = \kappa b\sqrt{m}$ on the maximum length of Sc. We have already seen that R plays an important role for both efficiency and signature size as the parameter $\sigma = \alpha R$. To keep the signature size the same for a larger R we must lower α, which decreases efficiency. Alternatively we can keep α the same but then the signature size will increase. In any case, let us first give a more detailed description of the security proof before tackling this newly arisen issue.

From $B = [A'|A''] \in \mathbb{Z}_q^{n\times m}$ we create the public key $A = [2A'|2A'' + qI_n]$ and give it to the forger. Although we do not know a secret key S such that $AS = qI_n \bmod 2q$, we can still provide a signature for any message μ requested by the forger by picking (z, c) from the correct distribution and programming the random oracle accordingly as is done in Theorem 5.18. When the forger provides a successful forgery, we can again use the forking lemma to obtain two equations $Az = Tc \bmod 2q$ and $Az' = Tc' \bmod 2q$. In

this case we have $Az = qc \bmod 2q$ and $Az' = qc' \bmod 2q$. Now we obtain $A(z-z') = q(c-c') \bmod 2q$. So in particular $A(z-z') = 0 \bmod q$ and since $c \neq c' \bmod 2$ we have $q(c-c') \neq 0 \bmod 2q$. Thus $z \neq z' \bmod 2q$. Moreover, $||z - z'||_\infty < q/2$, which implies $z \neq z' \bmod q$. Now $A(z-z') = 0 \bmod q$ in turn implies $2B(z-z') = 0 \bmod q$ and since 2 is invertible modulo q, we have found a $v = z - z' \neq 0 \bmod q$ such that $Bv = 0 \bmod q$, which solves the SIS problem. We would like to refer the reader back to chapter 5 for a formal treatment of the security proof or alternatively to [9] for the security proof in the BLISS setting.

Returning to the issue of the larger bound R in BLISS [9] compared to [7], we draw attention to the work of Ducas [26] which focuses on reducing R. Without changing any other parameters in the scheme, i.e. keeping σ constant, a reduction in R would mean an increase in α. This directly translates into a speedup of the signature generation as the number of repetitions is given by $M = \exp(\frac{1}{2\alpha^2})$. In fact, Ducas is able to reduce R by replacing the bound on $||Sc||$ given in [9] by the bound on $||Sc'||$, for appropriately chosen c'.

Ducas accomplishes this result by choosing a different representation of the binary challenges c. We have noted the challenge vector c must be chosen using coefficients in $\{0, 1\}$ rather than $\{-1, 0, 1\}$. However, this does not restrict us to using the canonical binary representation $c' \in \mathbb{Z}^k$ of $c \in \mathbb{Z}_2^k$. In particular, one may negate individual coordinates of c' in order to obtain a smaller $||Sc'||$ [26]. Ducas proposes an algorithm that efficiently computes an appropriate c' with $c' = c \bmod 2$ such that $||Sc'||$ becomes smaller. This improvement on BLISS is called BLISS-B.

It should be noted that the signature will remain (z, c). So the sign choices made in c', which carry information about the secret key S, are only used to compute $v = Sc'$. By the rejection sampling step v is perfectly hidden.

Only the signature generation algorithm of BLISS-B is changed compared to BLISS and we will briefly show that the verification algorithm indeed still works. We still check the bounds on the length of z as before. Recall it must also be verified that $c = H(Az + qc \bmod 2q, \mu)$. In this new setting we have $z = y + (-1)^a Sc'$ with $c' = c \bmod 2$ and we still have $c = H(Ay \bmod 2q, \mu)$. Then

$$\begin{aligned} Az + qc = A(y + (-1)^a Sc') + qc = Ay + (-1)^a ASc' + qc \\ = Ay + (-1)^a qc' + qc \bmod 2q. \end{aligned}$$

Clearly

$$Ay + (-1)^a qc' + qc = Ay \bmod q,$$

and since we have $c' = c \bmod 2$,

$$Ay + (-1)^a qc' + qc = Ay \bmod 2.$$

Therefore we can conclude by the Chinese remainder theorem that

$$Ay + (-1)^a qc' + qc = Ay \bmod 2q,$$

and thus verification holds. Since both BLISS and BLISS-B have the same verification algorithm, a signature generated by BLISS-B will also be valid for BLISS and vice versa.

We can conclude that BLISS, even without the improvements made in BLISS-B, is faster and generates shorter signatures than our results in chapter 5 based on Lyubashevsky's original scheme [7]. However it should be noted that the security analysis for both schemes is done based on the work of [23] for random lattices. While this applies to Lyubashevsky's original scheme, the lattices used in BLISS are not perfectly random as there exist unusually short vectors in these lattices by construction. As of writing it is unclear whether one can exploit this structure and how hard it is to find those short vectors. With the further advancement in lattice cryptanalysis an attack may be found in the future.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, IEEE Computer Society Press, Nov. 1994.

[2] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, pp. 883–887, Dec. 2001.

[3] N. Xu, J. Zhu, D. Lu, X. Zhou, X. Peng, and J. Du, "Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system," *Phys. Rev. Lett.*, vol. 108, p. 130501, Mar. 2012.

[4] N. S. Dattani and N. Bryans, "Quantum factorization of 56153 with only 4 qubits," *CoRR*, vol. abs/1411.6758, Nov. 2014. https://arxiv.org/abs/1411.6758.

[5] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in Cryptology – CRYPTO'97* (B. S. Kaliski Jr., ed.), vol. 1294 of *Lecture Notes in Computer Science*, pp. 112–131, Springer, Heidelberg, Aug. 1997.

[6] P. Q. Nguyen and O. Regev, "Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures," *Journal of Cryptology*, vol. 22, pp. 139–160, Apr. 2009.

[7] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Advances in Cryptology – EUROCRYPT 2012* (D. Pointcheval and T. Johansson, eds.), vol. 7237 of *Lecture Notes in Computer Science*, pp. 738–755, Springer, Heidelberg, Apr. 2012. Full version available at https://eprint.iacr.org/2011/537.

[8] T. van Erven and P. Harremoës, "Rényi divergence and Kullback-Leibler divergence," *IEEE Trans. Information Theory*, vol. 60, pp. 3797–3820, Jul. 2014.

[9] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Advances in Cryptology – CRYPTO 2013, Part I* (R. Canetti and J. A. Garay, eds.), vol. 8042 of *Lecture Notes in Computer Science*, pp. 40–56, Springer, Heidelberg, Aug. 2013.

[10] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal on Computing*, vol. 17, pp. 281–308, Apr. 1988.

[11] D. Micciancio, "Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions," in *43rd Annual Symposium on Foundations of Computer Science*, pp. 356–365, IEEE Computer Society Press, Nov. 2002.

[12] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *40th Annual ACM Symposium on Theory of Computing* (R. E. Ladner and C. Dwork, eds.), pp. 197–206, ACM Press, May 2008.

[13] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-quantum Cryptography* (D. J. Bernstein, J. Buchmann, and E. Dahmen, eds.), pp. 147–191, Springer, 2008.

[14] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with small parameters," in *Advances in Cryptology – CRYPTO 2013, Part I* (R. Canetti and J. A. Garay, eds.), vol. 8042 of *Lecture Notes in Computer Science*, pp. 21–39, Springer, Heidelberg, Aug. 2013.

[15] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, ACM Press, May 1996.

[16] S. Khot, "Hardness of approximating the shortest vector problem in lattices," in *45th Annual Symposium on Foundations of Computer Science*, pp. 126–135, IEEE Computer Society Press, Oct. 2004.

[17] D. Micciancio and S. Goldwasser, "Complexity of lattice problems: a cryptographic perspective," *The Springer International Series in Engineering and Computer Science*, vol. 671, 2012.

[18] T. Liu, "On basing search SIVP on NP-hardness." Cryptology ePrint Archive, Report 2016/906, 2016. https://eprint.iacr.org/2016/906.

[19] A. Rényi, "On measures of entropy and information," *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1: Contributions to the Theory of Statistics, pp. 547–561, 1961.

[20] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld, "Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance," in *Advances in Cryptology – ASIACRYPT 2015, Part I* (T. Iwata and J. H. Cheon, eds.), vol. 9452 of *Lecture Notes in Computer Science*, pp. 3–24, Springer, Heidelberg, Nov. / Dec. 2015.

[21] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *ACM CCS 06: 13th Conference on Computer and Communications Security* (A. Juels, R. N. Wright, and S. Vimercati, eds.), pp. 390–399, ACM Press, Oct. / Nov. 2006.

[22] N. C. Dwarakanath and S. D. Galbraith, "Sampling from discrete Gaussians for lattice-based cryptography on a constrained device," *Appl. Algebra Eng. Commun. Comput.*, vol. 25, pp. 159–180, Jun. 2014.

[23] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology – EUROCRYPT 2008* (N. P. Smart, ed.), vol. 4965 of *Lecture Notes in Computer Science*, pp. 31–51, Springer, Heidelberg, Apr. 2008.

[24] J. Postel, ed., "Internet protocol," *RFC791*, Sep. 1981. https://tools.ietf.org/html/rfc791.

[25] C. Hornig, "A standard for the transmission of IP datagrams over ethernet networks," *RFC894*, Apr. 1984. https://tools.ietf.org/html/rfc894.

[26] L. Ducas, "Accelerating BLISS: the geometry of ternary polynomials." Cryptology ePrint Archive, Report 2014/874, 2014. https://eprint.iacr.org/2014/874.