NTRU+Sign: Compact NTRU-Based Signatures Using Bimodal Distributions

Joo Woo[1], Jonghyun Kim[1], Ga Hee Hong[1], Seungwoo Lee[1],
Minkyu Kim[2], Hochang Lee[2], and Jong Hwan Park[3]

[1]Korea University, Korea, {woojoo0121,yoswuk,hongh,kak5175}@korea.ac.kr
[2]The Affiliated Institute of ETRI, Korea, {mkkim,lhc254}@nsr.re.kr
[3]Sangmyung University, Korea, jhpark@smu.ac.kr

January 23, 2025

Abstract

We present a new lattice-based signature scheme, called 'NTRU+Sign', using the Fiat-Shamir with Aborts framework. The proposed scheme is designed based on a novel NTRU-based key structure that fits well with bimodal distributions, enabling efficiency improvements compared to its predecessor, BLISS. The novel NTRU-based key structure is characterized by: (1) effectively changing a modulus from $2q$ to q, which is different from the existing usage of $2q$ for bimodal distributions, and (2) drastically reducing the magnitude of a secret key, which directly leads to compactness of signature sizes. We provide two concrete parameter sets for NTRU+Sign, supporting 93-bit and 211-bit security levels. Using the technique from GALACTICS (that was suggested as the constant-time implementation of BLISS), our analysis shows that NTRU+Sign achieves a good balance between computational efficiency and signature compactness, with constant-time implementation. For instance, at the NIST-3 security level, NTRU+Sign produces signatures that are significantly smaller than Dilithium and HAETAE, while providing faster verification speeds. These advantages position NTRU+Sign as a competitive and practical solution for real-world deployments.

1 Introduction

Lyubashevsky [30, 32] proposed lattice-based signature schemes based on the Fiat-Shamir with Aborts (FSwA) framework. Their underlying lattice-based identification scheme involves a publicly shared matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a public key $\mathbf{T} = \mathbf{AS} \bmod q$, where the secret key $\mathbf{S} \in \mathbb{Z}^{m\times k}$ consists of small entries. In response to a challenge $\mathbf{c} \in \mathbb{Z}^k$, a response $\mathbf{z}$ is computed as $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$, where $\mathbf{y} \in \mathbb{Z}^m$ is a small masking vector sampled from a certain distribution. A verifier checks if the relation $\mathbf{Ay} = \mathbf{Az} - \mathbf{Tc} \bmod q$ holds. Unlike the Schnorr identification scheme, the distribution of $\mathbf{z}$ remains inherently biased by $\mathbf{Sc}$, leading to a potential leakage of information about the secret key $\mathbf{S}$. To address this, Lyubashevsky [30, 32] employed the technique of *rejection sampling* which outputs a candidate value of $\mathbf{z}$ with probability $f(\mathbf{z})/(Mg(\mathbf{z}))$, where f indicates a target distribution, g indicates a source distribution, and M is a constant. Using rejection sampling, the source distribution of $\mathbf{z}$ is forced to align with the target distribution, effectively removing any dependency on the secret key $\mathbf{S}$ from $\mathbf{z}$.

In 2013, Ducas *et al.* [17] introduced a novel lattice-based signature scheme, BLISS, which leveraged the so-called *bimodal distributions*, achieving significantly more compact signature sizes. In BLISS, the bimodal distributions are characterized by using a random bit b such that $\mathbf{z} = \mathbf{y} + (-1)^b\mathbf{Sc}$, where $\mathbf{y}$ is sampled from a discrete Gaussian distribution D_σ^m with standard deviation σ. This construction allowed the source distribution of $\mathbf{z}$ to better align with the target distribution of $\mathbf{y}$, improving the efficiency of the rejection sampling process. To ensure correctness regarding the usage of $(-1)^b$, BLISS incorporates arithmetic modifications, such as replacing the modulus q with $2q$ and changing the key generation algorithm to satisfy the condition $\mathbf{AS} = q\mathbf{I} \bmod 2q$. These modifications guarantee that $\mathbf{Az}-q\mathbf{c} = \mathbf{Ay}+(-1)^b\mathbf{ASc}-q\mathbf{c} = \mathbf{Ay} \bmod 2q$, regardless of whether $b = 0$ or $b = 1$. While BLISS was considered one of the most efficient lattice-based signature scheme, its practical implementations revealed significant vulnerabilities to side-channel timing attacks [9, 11, 21, 35]. Such attacks mainly exploited flaws (from a side-channel attack perspective) in the implementation of transcendental functions that are necessary for both discrete Gaussian samplings for $\mathbf{y}$ and the rejection sampling process involved in computing $f(\mathbf{z})/(Mg(\mathbf{z}))$.

To address these vulnerabilities, Barthe et al. [7] proposed GALACTICS, a secure implementation of BLISS that mitigates the side-channel vulnerabilities in [9, 11, 21, 35], while maintaining efficiency comparable to the original implementation. In GALACTICS, polynomial approximations of transcendental functions ensure that all operations related to discrete Gaussian sampling and the computation of $f(\mathbf{z})/(Mg(\mathbf{z}))$ are executed in constant time. Following the basic structure of BLISS, another approach has been made by HAETAE [13], a module lattice-based signature scheme that uses a uniform hyperball sampling [15]. HAETAE simplifies the rejection sampling process by simply checking if the Euclidean norms of $\mathbf{z}$ are sufficiently small within a preset bound, rather than computing $f(\mathbf{z})/(Mg(\mathbf{z}))$ as in BLISS. However, uniform hyperball sampling requires a large number of discrete Gaussian samplings with a large standard deviation and high-precision arithmetic, using the polynomial approximation of the exponential function e^{-x} essentially similar to GALACTICS, and involves additional computations such as inverse of square roots and the ℓ_2 norm of a vector, adding computational complexity.

Ducas *et al.* [19] proposed Dilithium, a lattice-based signature scheme that goes back to the form of $\mathbf{z} = \mathbf{y}+\mathbf{Sc}$ (that is, avoiding bimodal distributions). Instead of using discrete Gaussian samplings, Dilithium uses uniform sampling in hypercubes to generate $\mathbf{y}$, significantly simplifying the sampling part. Additionally, the rejection sampling process is also simplified by checking whether the infinite norm of $\mathbf{z}$ is sufficiently small within a preset bound. Dilithium was later selected by NIST (FIPS 204) as the module-lattice-based digital signature standard. Although uniform sampling simplifies implementation and makes it easier to protect against side-channel timing attacks, it results in less compact signatures than BLISS. In theory, it has been justified by prior work [15], which shows that bimodal Guassian distributions provide better signature compactness than uniform distributions in hypercubes.

1.1 Our Contribution

The goal of this paper is to revisit BLISS [17] so as to achieve its full potential. We suggest a novel lattice-based signature scheme, called NTRU+Sign, building upon BLISS by relying on the NTRU problem [26, 27]. At a similar security level, NTRU+Sign achieves more compact signatures and faster verification speeds than Dilithium, HAETAE, and re-parametrized BLISS. To achieve our goal, we made several new changes to BLISS.

First, we introduce a novel NTRU-based key structure suitable for bimodal distributions, which eliminates the need for modulus $2q$ operations by transitioning to modulus q operations. Let $R = \mathbb{Z}[x]/(x^n + 1)$ be a polynomial-based ring for the NTRU-based setting. For a public key $\mathbf{A} \in R^{1\times 2}$ and its relevant signing key $\mathbf{S} \in R^{2\times 1}$, BLISS satisfies the condition of $\mathbf{AS} = q \bmod 2q$, whereas NTRU+Sign satis-

fies the condition of $\mathbf{AS} = \hat{q} \bmod q$, where $\hat{q} = 2^{-1} \bmod q$. In both schemes, signatures are generated as $(\mathbf{z} = \mathbf{y} + (-1)^b \mathbf{Sc}, \mathbf{c})$, leveraging bimodal discrete Gaussian distributions. The commitment (corresponding to the identification scheme) uses only the high-order bits of $\mathbf{Ay}$. The key distinction lies in the verification process. BLISS checks if the high-order bits of $\mathbf{Az} + \mathbf{c}q \bmod 2q$ match the commitment, and NTRU+Sign checks if the high-order bits of $\mathbf{Az} + \mathbf{c}\hat{q} \bmod q$ match the commitment. The verification of NTRU+Sign succeeds from the following observation. When $b = 1$, the value $\mathbf{z} = \mathbf{y} - \mathbf{Sc}$ results in $\mathbf{Az} + \mathbf{c}\hat{q} = \mathbf{Ay} \bmod q$, which satisfies the verification equation. Also, when $b = 0$, the value $\mathbf{z} = \mathbf{y} + \mathbf{Sc}$ results in $\mathbf{Az} + \mathbf{c}\hat{q} = \mathbf{Ay} + 2\mathbf{c}\hat{q} = \mathbf{Ay} + \mathbf{c} \bmod q$. Since the coefficients of the challenge $\mathbf{c} \in R$ are 0 or 1, the high-order bits of $\mathbf{Ay}$ and $\mathbf{Ay} + \mathbf{c}$ are equal with high probability. Furthermore, the signing process in NTRU+Sign incorporates an equality check in advance, ensuring this condition holds without requiring modulus $2q$ operations.

One of the main challenges in achieving small signature sizes within the FSwA framework is to make the upper bound of the Euclidean norm of $\mathbf{Sc}$ as small as possible. This is crucial because a smaller norm of $\mathbf{Sc}$ results in a source distribution that more closely resembles the target distribution, allowing for a more compact signature size under a fixed rejection rate. We address this challenge through two key approaches: (1) Compared to BLISS, the NTRU-based key structure in NTRU+Sign significantly reduces the magnitude of $\mathbf{S}$ itself. For small polynomials $\mathbf{f}$ and $\mathbf{g}$ in R, the public and signing key pairs of BLISS are $\mathbf{A} = (2((2\mathbf{f}+1)/\mathbf{g} \bmod q), q-2)$ and $\mathbf{S} = (\mathbf{g}, 2\mathbf{f}+1)^T$, and this structure has been made in the process of harmonizing bimodal structure and the classical NTRU problem *over* q. But, NTRU+Sign can avoid such artificial structure by setting the public key as $\mathbf{A} = ((\mathbf{f} + \hat{q})/\mathbf{g} \bmod q, 1)$, and so the corresponding signing key $\mathbf{S} = (\mathbf{g}, -\mathbf{f})^T$ has smaller norm than that of BLISS. For example, for polynomials $\mathbf{f}$ and $\mathbf{g}$ with coefficients sampled from $\{-1, 0, 1\}$, the coefficients of $2\mathbf{f} + 1$ in BLISS belong to $\{-1, 1, 3\}$, leading to a higher upper bound on $\|\mathbf{S}\|$ and, consequently, $\|\mathbf{Sc}\|$. In contrast, NTRU+Sign ensures that the coefficients of $\mathbf{S} = (\mathbf{g}, -\mathbf{f})^T$ still belong to $\{-1, 0, 1\}$. (2) We further tighten the bound on $\|\mathbf{Sc}\|$ by leveraging the canonical embedding into $\mathbb{C}^n$, as studied in HAETAE [13]. Compared to BLISS, this significantly improves the estimation of $\|\mathbf{Sc}\|$, further contributing to the compactness of the signature. Combining these improvements, NTRU+Sign achieves the smallest upper bound on $\|\mathbf{Sc}\|$ among comparable schemes. At the 180-bit security level, for example, the upper bound is 341 for NTRU+Sign, but 1223 for BLISS and 516 for HAETAE. Separately, compared to HAETAE, NTRU+Sign benefits from a more moderate increase in the Hamming weight τ of $\mathbf{c}$ as the security parameter λ increases. Since τ directly affects the upper bound on $\|\mathbf{Sc}\|$, this moderate increase is significant. In HAETAE, $\mathbf{c}$ is a polynomial with the fixed degree of 256, and thus τ is required to satisfy $\binom{256}{\tau} \approx 2^\lambda$. In NTRU+Sign, however, τ is determined by $\binom{n}{\tau} \approx 2^\lambda$, where the polynomial degree n grows with λ. For example, HAETAE requires $\tau = 80$ for $\lambda = 180$, while NTRU + Sign requires only $\tau = 36$ for $\lambda = 211$ when $n = 1024$.

Lastly, we provide two parameter sets: NTRU+Sign-$\{512, 1024\}$ based on the rings $\mathbb{Z}_q[x]/(x^n+1)$, targeting 93 bits and 211 bits of security levels, respectively. Table 1 presents a comparison between Dilithium, HAETAE, and NTRU+Sign in terms of public key sizes, signature sizes, and performance evaluations. Even at higher security levels, Table 1 demonstrates that NTRU+Sign achieves the shortest signature sizes, compared to Dilithium and HAETAE. For instance, at NIST security level 3 (aiming at 192-bit classical security), the signature size of NTRU+Sign-1024 is approximately 55% smaller than Dilithium and 35% smaller than HAETAE. Furthermore, when considering the combined size of the signature and public key, NTRU+Sign-1024 is about 40% smaller than Dilithium and 15% smaller than HAETAE. All parameter sets for NTRU+Sign are designed to support efficient Number Theoretic Transform (NTT) operations in their underlying rings. Our reference implementation, based on the open source[1] of GALACTICS, ensures constant-

[1] https://github.com/espitau/GALACTICS

Table 1: Comparison between Dilithium, HAETAE and NTRU+Sign

	Classical Security	|Sig| (bytes)	$\|pk\|$ (bytes)	|Sig|+$\|pk\|$ (bytes)	KeyGen (K cycle)	Sign (K cycle)	Verify (K cycle)
Dilithium-2	123	$2,420$	$1,312$	$3,732$	278	$1,290$	302
Dilithium-3	182	$3,293$	$1,952$	$5,245$	506	$2,076$	520
HAETAE-120	119	$1,474$	992	$2,466$	$1,731$	$7,785$	317
HAETAE-180	180	$2,349$	$1,472$	$3,821$	$2,686$	$9,831$	606
NTRU+Sign-512	93	751	768	$1,519$	975	$2,418$	118
NTRU+Sign-1024	211	$1,551$	$1,664$	$3,215$	$1,886$	$6,005$	204

time implementations of Gaussian sampling and rejection sampling, mitigating potential vulnerabilities to side-channel attacks. The implementation demonstrates that NTRU+Sign-1024 outperforms HAETAE-180, achieving 1.5 times faster key generation and signing speeds, and verification speeds approximately 3 times faster. In terms of signature compactness, we give another comparison with other lattice-based signature schemes, including re-parametrized BLISS[2], G + G [16], Patronus [5] and Falcon [23] in Section 4.3. Our results highlight NTRU+Sign as a highly efficient and compact lattice-based signature scheme, establishing a new benchmark for post-quantum cryptographic applications. Its compactness and computational efficiency make it a strong candidate for real-world implementation in high-security environments.

1.2 Related Works

Lyubashevsky introduced the first efficient lattice-based signature scheme, using rejection sampling with uniform distribution in hypercubes [30]. Since then, extensive research has been conducted on lattice-based signatures under the FSwA framework, focusing on optimizing sampling methods to both achieve signature compactness and simplify rejection processes. Signature schemes with uniform sampling were refined in [25, 3], achieving reduced signature sizes. Further progress was achieved by Dilithium [19], which optimized public key sizes through a truncation technique. While uniform sampling over hypercubes simplifies rejection sampling by utilizing infinite norm bounds, maintaining low rejection rates often necessitates larger parameters, leading to increased public key and signature sizes.

Gaussian-based signature schemes have also been extensively studied for achieving more compact signatures. Following Lyubashevsky's FSwA-based digital signature scheme [32], Ducas *et al.* presented Dilithium-G, a variation described in the first ePrint version of [19]. Dilithium-G employed a unimodal Gaussian distribution. This approach was later optimized by [15] through the analysis of imperfect rejection sampling using Rényi divergence instead of statistical distance. The adoption of bimodal Gaussian distributions in BLISS [17] significantly reduced signature sizes. However, the original security analysis of BLISS resulted in parameter sets that did not meet the claimed security levels under more rigorous analyzes.

Devevey *et al.* [15] derived generic optimal bounds for two critical metrics: minimal rejection rate and proof-of-knowledge size compactness. Their results demonstrated that both Gaussian distributions and uniform distributions within hyperballs achieve optimal bounds in the bimodal setting. Using this observation, Cheon *et al.* [13] proposed HAETAE, a signature scheme based on a bimodal hyperball distribution.

[2]The original BLISS paper employed a different security analysis, resulting in parameter sets that fail to meet the claimed security levels under the analysis presented in this paper. To achieve at least 128-bit security, we recalibrated the parameters, doubling the dimension to 1024 to ensure alignment with modern security standards. Additionally, we conducted an analysis of the rANS encoding process to estimate the signature size under the revised parameters.

HAETAE simplifies rejection sampling to a straightforward Euclidean norm check, achieving compact signature sizes. However, its hyperball sampling still depends on discrete Gaussian sampling as a subroutine, which requires high-precision fixed-point arithmetic. This dependency, coupled with additional operations like inverse square root and ℓ_2 norm calculations, introduces notable computational overhead.

An alternative approach to prevent secret value leakage without relying on rejection sampling was introduced by G + G [16]. This scheme generates signatures following a centered spherical Gaussian distribution, achieved through the convolution of two distinct Gaussian distributions, each with covariance dependent on the secret key. The primary advantage of G + G is its ability to generate signatures without requiring rejection sampling while ensuring no leakage of secret key information. However, its dependence on Gaussian sampling related to the secret key can cause potential vulnerabilities against side-channel attacks, and also need complexity to secure implementation.

Recently, Bambury *et al.* [5] proposed a new sampler based on uniform sampling within polytopes, leading to a novel signature scheme Patronus. This approach maintains the simplicity of uniform sampling while reducing proof-of-knowledge size, thereby decreasing overall signature sizes. Furthermore, the rejection sampling process is simplified by relying solely on checks involving ℓ_1 norm, infinite norm, and ℓ_2 norm in optimized versions, eliminating the need for transcendental function computations. However, Patronus suffers from significant computational overhead due to the high cost of uniform sampling within polytopes, and its signature sizes still remain relatively large.

2 Preliminaries

Let $\{0,1\}^*$ denote the set of all binary strings. Let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denote the quotient ring of integers modulo q. Define R and R_q as the rings $\mathbb{Z}[x]/(x^n+1)$ and $\mathbb{Z}_q[x]/(x^n+1)$, respectively, where q is a prime and n is a power of two. Elements in R_q are denoted in bold lowercase. Occasionally, elements of R_q that are either 0 or have all coefficients equal to 0 except for a nonzero constant term, such as 1 or $\hat{q}$, are written in lowercase rather than bold lowercase. Let $R_{n,\tau}$ denote the set of elements in R_q that have all zero coefficients except for τ out of n coefficients, which are 1. We have $|R_{n,\tau}| = \binom{n}{\tau}$. For a positive integer k, the centered binomial distribution (CBD), denoted by ψ_k, is a distribution over the integers $\mathbb{Z}$, which outputs $\sum_{i=1}^{k}(b_i - b'_i)$ using uniformly and independently sampled bits b_i, b'_i. For $\mathbf{f} \in R_q$, we use the notation '$\mathbf{f} \leftarrow \psi_k^n$' to represent that each coefficient of $\mathbf{f}$ is sampled from ψ_k. In general, for a probability distribution $\mathcal{D}$, $a \leftarrow \mathcal{D}$ denotes that a is sampled according to $\mathcal{D}$. For a set S, $a \leftarrow S$ denotes that a is sampled uniformly at random from S.

For any integer x within the range $[0, q)$ and any positive integer d, x can be uniquely written as $x = [x]_d \cdot 2^d + [x \bmod 2^d]$, where $[x \bmod 2^d] \in [-2^{d-1}, 2^{d-1})$ and $[x]_d = (x - [x \bmod 2^d])/2^d$ to drop the d least significant bits. These functions can be extended to polynomials, i.e., $[\mathbf{f}]_d = \sum_{i=0}^{n-1}[f_i]_d x^i$ for $\mathbf{f} = \sum_{i=0}^{n-1} f_i x^i$. We define the infinite norm of a polynomial $\mathbf{f}$ as $\|\mathbf{f}\|_\infty = \max_{0\le i\le n-1}|f_i|$. We define $\hat{q} = 2^{-1} \bmod q$ for an odd prime q, and $p = (q-1)/2^d$ for an integer d satisfying $q \equiv 1 \bmod 2^d$. Specifically, given the modulus p, and an integer $k \in \mathbb{Z}$, define $k' = k \bmod p$ as the unique element k' such that $-p/2 \le k' < p/2$.

Lemma 2.1 (Rejection Sampling [17])**.** Let V be an arbitrary set, and let $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. If $g_v : \mathbb{Z}^m \to \mathbb{R}$ is a family of probability distributions indexed by $v \in V$ with the property that there exists a constant $M \in \mathbb{R}$ such that

$$\forall v \in V, \forall z \in \mathbb{Z}^m, \Pr[M \cdot g_v(z) \ge f(z) \mid z \leftarrow f] \ge 1 - \epsilon,$$

then the output distributions of the following two algorithms are within a statistical distance of ϵ/M:

1. $v \leftarrow h, z \leftarrow g_v$, output (z, v) with probability $f(z)/(M \cdot g_v(z))$.

2. $v \leftarrow h, z \leftarrow f$, output (z, v) with probability $1/M$.

2.1 Discrete Gaussian Distribution

The Gaussian distribution with standard deviation $\sigma \in \mathbb{R}$ and center $c \in \mathbb{R}$ evaluated at $x \in \mathbb{R}$ is defined by $\rho_{c,\sigma}(x) = \exp(-(x-c)^2/(2\sigma^2))$, and more generally by $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp(-\|\mathbf{x}-\mathbf{c}\|^2/(2\sigma^2))$ for $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$. The discrete Gaussian distribution over $\mathbb{Z}^n$ centered at $\mathbf{c}$ is defined by $D^n_{\mathbf{c},\sigma}(\mathbf{x}) = \rho_{c,\sigma}(\mathbf{x})/\rho_{\mathbf{c},\sigma}(\mathbb{Z}^n)$. When the center $\mathbf{c}$ is $\mathbf{0}$ or $n = 1$, we generally omit it from the notation and simply write $\rho_\sigma(\mathbf{x})$, $D^n_\sigma(\mathbf{x})$, $D_\sigma(x)$. Identifying R with $\mathbb{Z}^n$ via coefficient embedding, $D^n_{\mathbf{c},\sigma}$ also denotes a probability distribution on R.

2.2 Cryptographic Definitions

We recall the definitions of an identification scheme and a signature scheme.

Definition 2.2. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets. A canonical identification scheme ID for an NP relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is a 3-round interactive proof system between a prover P and a verifier V, with a commitment set $\mathcal{W}$, challenge set $\mathcal{C}$, and response set $\mathcal{Z}$. The prover holds a pair $(x, y) \in R$, while the verifier only has x, where the pair (x, y) is generated by a PPT algorithm Gen, called the instance generator. The event "$z = \perp$" is called an abort, and its probability β is called the probability of aborting. The prover is written as $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$, and the verifier as $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2)$, with the following specifications:

- P_1: $(x, y) \to (w, st)$ is a PPT algorithm that takes as input a pair of strings in $\mathcal{X} \times \mathcal{Y}$ and outputs a commitment $w \in \mathcal{W}$ and a state $st \in \{0,1\}^*$.
- V_1: $(x, w) \to c$ is a PPT algorithm that takes as input a string x and a commitment w, and outputs a challenge $c \in \mathcal{C}$.
- P_2: $(x, y, w, c, st) \to z$ is a PPT algorithm that takes as input a pair of strings (x, y), a commitment w, a challenge c, and a state st, and outputs a response $z \in \mathcal{Z} \cup \{\perp\}$ (we say that P_2 aborts if it outputs $\perp$).
- V_2: $(x, w, c, z) \to b \in \{0, 1\}$ is a deterministic polynomial-time algorithm that takes as inputs a string x, a commitment w, a challenge c, and a response z, and outputs a bit b, which represents acceptance (1) or rejection (0); in the case that $z = \perp$, it returns 0.

We then recall the two definitions of statistical properties of an identification scheme, which are perfect accepting honest-verifier zero-knowledge (paHVZK) and the min-entropy of the commitment.

Definition 2.3 (Perfect Accepting Honest-verifier Zero-knowledge [6])**.** An identification scheme is said to be paHVZK if there exists a poly-time algorithm Sim that, when given the public key pk, outputs (w, c, z) with a distribution that is identical to the distribution of a transcript (w, c, z) produced by an honest execution of the protocol conditioned on $z \neq \perp$.

Definition 2.4 (Commitment Min-Entropy [29])**.** For $\alpha \geq 0$, we say that an identification scheme $((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ with an instance generator Gen has commitment min-entropy α if $H_\infty[w|(w, st) \leftarrow \mathsf{P}_1(x, y)] \geq \alpha$, for all $(x, y) \leftarrow \mathsf{Gen}(1^\lambda)$.

Definition 2.5. A digital signature (DS) scheme for a message space $\mathcal{M}$ consists of three algorithms: KeyGen, Sign, and Verify, which are defined as follows:

- $\mathsf{KeyGen}(\lambda)$: The key generation algorithm takes as input a security parameter λ and outputs a public key and a secret key (pk, sk).
- $\mathsf{Sign}(sk, \mu)$: The signing algorithm takes as input the secret key sk and a message $\mu \in \mathcal{M}$, and then outputs a signature σ.
- $\mathsf{Verify}(pk, \mu, \sigma)$: The verification algorithm takes as input the public key pk, a message μ, and a signature σ, and then outputs 1 if the signature is valid or 0 otherwise.

We say that a signature scheme is $(1-\eta)$-*correct* if the following condition holds: for all $(pk, sk) \in \mathsf{KeyGen}(\lambda)$ and all messages $\mu \in \mathcal{M}$,

$$\Pr\left[\mathsf{Verify}(pk, \mu, \sigma) = 1 \,|\, (pk, sk) \leftarrow \mathsf{KeyGen}(\lambda); \sigma \leftarrow \mathsf{Sign}(sk, \mu)\right] > 1 - \eta(\lambda)$$

where η is a negligible function for the security parameter λ.

Definition 2.6 (Unforgeability)**.** Let $\mathsf{DS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme. The unforgeability against chosen-message attacks (UF-CMA) is defined via the following experiment $\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{DS}}(\lambda)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

1. $\mathcal{C}$ runs $(pk, sk) \leftarrow \mathsf{KeyGen}(\lambda)$ and gives pk to $\mathcal{A}$.
2. $\mathcal{A}$ queries the signing oracle $\mathsf{Sign}(sk, \mu)$ with a message μ.
3. Finally, $\mathcal{A}$ outputs a signature σ^* and a message μ^* that was not previously queried on the signing oracle. $\mathcal{C}$ returns 1 if $\mathsf{Verify}(pk, \mu^*, \sigma^*) = 1$, and otherwise returns 0 as the output of the game.

The advantage of $\mathcal{A}$ in breaking the UF-CMA security of DS is defined as $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{DS}}(\mathcal{A}) = \Pr[\mathsf{UF\text{-}CMA}^{\mathcal{A}}_{\mathsf{DS}} \Rightarrow 1]$. We say that a signature scheme is UF-CMA secure if, for any polynomial-time adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{DS}}(\mathcal{A}) \leq \epsilon(\lambda)$, where ϵ is a function of the security parameter λ.

The unforgeability against no-message attack, denoted by UF-NMA is defined similarly except that the adversary is not allowed to query any signature per message.

2.3 Hardness Assumptions

The security of our construction is based on the hardness of two lattice problems, namely the NTRU problem and the BimodalSelftargetRSIS problem.

Definition 2.7 ($\mathsf{NTRU}_{n,q,\psi}$)**.** Let ψ be a distribution over R_q and let $\hat{q} = 2^{-1} \bmod q$. An adversary $\mathcal{A}$ solving the $\mathsf{NTRU}_{n,q,\psi}$ problem has advantage

$$\mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\psi}(\mathcal{A}) := \left| \begin{array}{c} \Pr\left[b = 1 |\, \mathbf{u} \leftarrow R_q; b \leftarrow \mathcal{A}(\mathbf{u})\right] - \\ \Pr\left[b = 1 |\, \mathbf{f}, \mathbf{g} \leftarrow \psi; b \leftarrow \mathcal{A}((\mathbf{f} + \hat{q})/\mathbf{g})\right] \end{array} \right|.$$

When setting $\mathbf{h} = (\mathbf{f} + \hat{q})/\mathbf{g}$, the above definition of the NTRU problem involves $2\mathbf{h} = (2\mathbf{f} + 1)/\mathbf{g}$. Recent results [20, 37] show that the inverse of $2\mathbf{h}$, that is, $\mathbf{g}/(2\mathbf{f} + 1)$, is also indistinguishable from a random element in R_q. Thus, it can be inferred that $2\mathbf{h} = (2\mathbf{f} + 1)/\mathbf{g}$ is also indistinguishable from a random element in R_q, and so is $\mathbf{h}$, because 2 is coprime to q.

Before introducing a new lattice-based problem, we define the ring version of Short Integer Solution problem (RSIS) first.

Definition 2.8 ($\mathsf{RSIS}_{n,q,\beta}$)**.** Let $\beta > 0$ and q be a positive modulus. An adversary $\mathcal{A}$ solving the $\mathsf{RSIS}_{n,q,\beta}$ problem has advantage

$$\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,\beta}(\mathcal{A}) := \Pr\big[[1 \,|\, \mathbf{a}] \cdot \mathbf{y} = 0 \ \wedge \ 0 < \|\mathbf{y}\| \leq \beta \mid \mathbf{a} \leftarrow R_q; \mathbf{y} \in R_q^2 \leftarrow \mathcal{A}(\mathbf{a})\big].$$

Next, we introduce a new lattice-based problem similar to the SelfTargetMSIS problem [29], which allows us to directly prove the UF-NMA security of our signature scheme in the Quantum Random Oracle Model (QROM).

Definition 2.9 ($\mathsf{BiomodalSelfTargetRSIS}_{H,n,q,\beta}$)**.** Let $H : \{0,1\}^* \to R_{n,\tau}$ be a cryptographic hash function, and let $q > 0$ be an odd modulus. An adversary $\mathcal{A}$ solving the $\mathsf{BiomodalSelfTargetRSIS}_{H,n,q,\beta}$ problem has advantage

$$\mathsf{Adv}^{\mathsf{BiomodalSelfTargetRSIS}}_{H,n,q,\beta}(\mathcal{A}) := \Pr\left[\begin{array}{c} \|\mathbf{Y}\| < \beta \ \wedge \ \|\mathbf{Y}\|_\infty < (q-2)/4 \\ \wedge \, H([1 \,|\, \mathbf{a}] \cdot \mathbf{Y} + \mathbf{c}\hat{q}, \mu) = \mathbf{c} \end{array} \;\middle|\; \mathbf{a} \leftarrow R_q; \left(\mathbf{Y} := \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \mathbf{c}, \mu\right) \leftarrow \mathcal{A}^{|H\rangle}(\mathbf{a}) \right]$$

where $\hat{q} = 2^{-1} \bmod q$, $(\mathbf{y}_1, \mathbf{y}_2) \in R_q^2$, $\mathbf{c} \in R_{n,\tau}$, and $\mu \in \{0,1\}^*$.

Now, we will show that the $\mathsf{BiomodalSelfTargetRSIS}$ problem is reduced to the RSIS, assuming that H is a random oracle. If $\mathcal{A}$ only has classical access to H, then there is a reduction, using the forking lemma [8], to prove that $\mathsf{Adv}^{\mathsf{BiomodalSelfTargetRSIS}}_{H,n,q,\beta}(\mathcal{A}) \approx \sqrt{\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,4\beta+2\sqrt{\tau}}(\mathcal{B})/Q_h}$, where Q_h is the number of classical queries to H. We give its proof sketch below.

Assume that $\mathcal{A}$ is a solver for the $\mathsf{BiomodalSelfTargetRSIS}_{H,n,q,\beta}$ problem. Then, $\mathcal{B}$ passes the polynomial $\mathbf{a}$ from its $\mathsf{RSIS}_{n,q,4\beta+2\sqrt{\tau}}$ instance to $\mathcal{A}$ and replies to $\mathcal{A}$'s queries $H(\mathbf{w}, \mu)$ with a uniformly random $\mathbf{c} \in R_{n,\tau}$. If $\mathcal{A}$ returns a solution $(\mathbf{Y} = [\mathbf{y}_1 \,|\, \mathbf{y}_2]^T, \mathbf{c}, \mu)$ to $\mathsf{BiomodalSelfTargetRSIS}_{H,n,q,\beta}$, then $\mathcal{B}$ reprograms the "winning" query $H([1|\mathbf{a}] \cdot \mathbf{Y} + \mathbf{c}\hat{q}, \mu)$ to a different random element $\mathbf{c}'$. The forking lemma states that $\mathcal{A}$ outputs another solution $(\mathbf{Y}' = [\mathbf{y}_1' | \mathbf{y}_2']^T, \mathbf{c}', \mu)$ with probability $1/Q_h$. Now, we have

$$\begin{cases} [1 \,|\, \mathbf{a}] \cdot \mathbf{Y} + \mathbf{c}\hat{q} = \mathbf{w} = [1 \,|\, \mathbf{a}] \cdot \mathbf{Y}' + \mathbf{c}'\hat{q} \\ \|\mathbf{Y}\| < \beta, \|\mathbf{Y}'\| < \beta, \|\mathbf{Y}\|_\infty < (q-2)/4, \|\mathbf{Y}'\|_\infty < (q-2)/4. \end{cases}$$

Thus, we obtain $[1 \,|\, \mathbf{a}] \cdot (\mathbf{Y} - \mathbf{Y}') + (\mathbf{c} - \mathbf{c}')\hat{q} = 0$. Since $\hat{q}$ satisfies $2\hat{q} = 1 \bmod q$ by definition, we derive $2[1 \,|\, \mathbf{a}] \cdot (\mathbf{Y} - \mathbf{Y}') + \mathbf{c} - \mathbf{c}' = 0$, following that $[1 \,|\, \mathbf{a}] \cdot \begin{bmatrix} 2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}' \\ 2\mathbf{y}_2 - 2\mathbf{y}_2' \end{bmatrix} = 0$. The condition $\mathbf{c} \neq \mathbf{c}' \bmod 2$ implies that $2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}'$ is non-zero over R. Since $\|\mathbf{Y}\|_\infty, \|\mathbf{Y}'\|_\infty < (q-2)/4$, we obtain $\|2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}'\|_\infty < q$. Therefore, $2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}' \neq 0$. Moreover, we have $0 < \|(2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}', 2\mathbf{y}_2 - 2\mathbf{y}_2')\| \leq \|(2\mathbf{y}_1 - 2\mathbf{y}_1', 2\mathbf{y}_2 - 2\mathbf{y}_2')\| + \|(\mathbf{c} - \mathbf{c}', 0)\| \leq \|2\mathbf{Y} - 2\mathbf{Y}'\| + 2\sqrt{\tau} < 4\beta + 2\sqrt{\tau}$.

This provides an $\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,4\beta+2\sqrt{\tau}}$ solution for $[1 \,|\, \mathbf{a}]$, which is a polynomial vector $[2\mathbf{y}_1 - 2\mathbf{y}_1' + \mathbf{c} - \mathbf{c}' \,|\, 2\mathbf{y}_2 - 2\mathbf{y}_2']^T$.

3 Our NTRU+Sign Signature Scheme

3.1 Construction

We present the NTRU+Sign scheme in Figure 1.

$\underline{\mathsf{KeyGen}(1^\lambda)}$

$\mathbf{f}, \mathbf{g} \leftarrow \psi_1^n$
if $\mathbf{g}$ is not invertible in R_q, **then** restart
$\mathbf{S} := (\mathbf{g}, -\mathbf{f})$
if $\mathcal{N}(\mathbf{S}) > \gamma^2 n$, **then** restart
$\mathbf{a} := (\mathbf{f} + \hat{q})/\mathbf{g} \in R_q$ // $\hat{q} = 2^{-1} \bmod q$
$pk := \mathbf{a}$
$sk := \mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$
return (pk, sk)

$\underline{\mathsf{Sign}(sk = \mathbf{S}, \mu)}$

$\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \leftarrow \mathcal{D}_\sigma^n \times \mathcal{D}_\sigma^n$
$\mathbf{u} := \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2 \in R_q$
$\mathbf{c} := H([\mathbf{u}]_d \bmod p, \mu) \in R_{n,\tau}$ // $p = (q-1)/2^d$
Choose a random bit $b \leftarrow \{0, 1\}$
$\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b \mathbf{S}\mathbf{c}$ // $\mathbf{z}_i = \mathbf{y}_i + (-1)^b \mathbf{s}_i \mathbf{c}$
Continue with probability $1/(M\exp(-\frac{\|\mathbf{S}\mathbf{c}\|^2}{2\sigma^2})\cosh(\frac{\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle}{\sigma^2}))$ // rejection sampling
  otherwise restart
if $[\mathbf{u}]_d \neq [\mathbf{u} + (-1)^b \mathbf{c}]_d$, **then** restart // equality check
$\mathbf{h} := [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p$
if $\|(\mathbf{z}_1, 2^d \mathbf{h})\| > B_2$, **then** restart
if $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty > B_\infty$, **then** restart
return $(\mathbf{z}_1, \mathbf{h}, \mathbf{c})$

$\underline{\mathsf{Verify}(pk, \mu, (\mathbf{z}_1, \mathbf{h}, \mathbf{c}))}$

if $\|(\mathbf{z}_1, 2^d \mathbf{h})\| > B_2$ **then** Reject
if $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty > B_\infty$ **then** Reject
Accept if $H([\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q} \bmod q]_d + \mathbf{h} \bmod p, \mu) = \mathbf{c}$

Figure 1: Description of NTRU+Sign

3.2 Correctness

The Verify algorithm will accept the signature $(\mathbf{z}_1, \mathbf{h}, \mathbf{c})$ for a message $\mu \in \{0,1\}^*$ to be signed if the following three conditions hold:

1. $\|(\mathbf{z}_1, 2^d\mathbf{h})\| \leq B_2$,
2. $\|(\mathbf{z}_1, 2^d\mathbf{h})\|_\infty \leq B_\infty$,
3. $\mathbf{c} = H([\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} \bmod p, \mu)$.

The first two conditions are guaranteed, because these conditions are already checked by the Sign algorithm in the same manner. The third one is also guaranteed by the following equations:

$$\begin{aligned}
\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q} &= \mathbf{a}(\mathbf{y}_1 + (-1)^b\mathbf{s}_1\mathbf{c}) + (\mathbf{y}_2 + (-1)^b\mathbf{s}_2\mathbf{c}) + \mathbf{c}\hat{q} - \mathbf{z}_2 \\
&= \mathbf{u} + (-1)^b\mathbf{c}(\mathbf{a}\mathbf{s}_1 + \mathbf{s}_2) + \mathbf{c}\hat{q} - \mathbf{z}_2 \qquad (\because\ \mathbf{u} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2) \\
&= \mathbf{u} + (-1)^b\mathbf{c}\hat{q} + \mathbf{c}\hat{q} - \mathbf{z}_2 \qquad (\because\ \mathbf{a}\mathbf{s}_1 + \mathbf{s}_2 = \hat{q}) \\
&= \mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}.
\end{aligned}$$

The last equality can be verified by the fact that: if $b = 0$, $(-1)^b\hat{q} + \hat{q}$ becomes $2\hat{q} = 1$ in R_q, and if $b = 1$, $(-1)^b\hat{q} + \hat{q}$ becomes 0 in R_q, which is equal to $1 - b$ for the same bit $b \in \{0,1\}$. Now, we can consider the hint $\mathbf{h}$ as being $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p = [\mathbf{u}]_d - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d \bmod p$, using the above last equality. Then, it holds that

$$\begin{aligned}
[\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} &= [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + [\mathbf{u}]_d - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d \bmod p \\
&= [\mathbf{u}]_d \bmod p,
\end{aligned}$$

as required in line 3 of the Verify algorithm.

3.3 Equality Check

Unlike BLISS, the Sign algorithm of NTRU+Sign includes an additional equality check in line 8, which checks whether or not $[\mathbf{u}]_d = [\mathbf{u}+(-1)^b\mathbf{c}]_d$. The purpose of this equality check is to conceal the information of the bit b from the hint $\mathbf{h}$, ensuring that the distribution of $\mathbf{h}$ remains the same regardless of the value of b. More concretely, $\mathbf{h}$ is calculated as $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p$; when $b = 0$, $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + \mathbf{c}]_d \bmod p$, and when $b = 1$, $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2]_d \bmod p$. This creates a difference in the distributions of $\mathbf{h}$ based on the choice of b. However, if the condition $[\mathbf{u}]_d = [\mathbf{u} + \mathbf{c}]_d$ is met for $b = 0$, then it holds that $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + \mathbf{c}]_d = [\mathbf{u} + \mathbf{c}]_d - [\mathbf{u} + \mathbf{c} - \mathbf{z}_2]_d$, which has the same distribution as in the case when $b = 1$, by setting $\mathbf{u}' = \mathbf{u} + \mathbf{c}$.

Technically, the equality check increases the expected number of the repetition rate M (heuristically) by $(2^d/(2^d-1))^\tau$ times, where $(2^d-1)/2^d$ is the probability that the equality holds for each coefficient of $\mathbf{u}$ [3]. Since $\mathbf{c} \in R_{n,\tau}$, only τ coefficients are considered, resulting in a probability of $((2^d-1)/2^d)^\tau$ for passing the equality check. Indeed, this probability does not significantly affect the repetition rate. For example, at the (classical) 211-bit security level (in case of NTRU+Sign-1024), with parameters $d = 8$ and $\tau = 36$, the probability is approximately 0.86, leading to an increase in the repetition rate M by a factor of about 1.15.

[3] It is assumed that the low d bits of $\mathbf{u}$ is uniformly distributed.

3.4 Rejection Sampling

The rejection sampling of NTRU+Sign, detailed in line 6 of the Sign algorithm, follows the approach used in BLISS [17]. For the discrete Gaussian distribution D_σ^{2n} with standard deviation σ, the source distribution is $g_{\mathbf{Sc}} = \frac{1}{2}D^{2n}_{\mathbf{Sc},\sigma} + \frac{1}{2}D^{2n}_{-\mathbf{Sc},\sigma}$ where $\mathbf{Sc}$ is computed in line 5 of the Sign algorithm over the space of all $(b, \mathbf{y})$ where $b \leftarrow \{0,1\}$ and $\mathbf{y} \leftarrow D_\sigma^{2n}$, whereas the target distribution is the centered discrete Gaussian distribution D_σ^{2n}, that is, $f = D_\sigma^{2n}$. Then, Lemma 2.1 shows that, for a fixed $\mathbf{Sc}$, the computed $\mathbf{z}$ in line 5 of the Sign algorithm should be accepted with probability:

$$P_{\mathbf{z}} = \frac{f(\mathbf{z})}{M g_{\mathbf{Sc}}(\mathbf{z})} = 1 \Big/ \Big(M \exp\Big(- \frac{\|\mathbf{Sc}\|^2}{2\sigma^2} \Big) \cosh\Big(\frac{\langle \mathbf{z}, \mathbf{Sc} \rangle}{\sigma^2} \Big) \Big),$$

where M is a predetermined positive real constant such that, for all possible values of $\mathbf{Sc}$, the inequality $f \leq M g_{\mathbf{Sc}}$ holds with probability 1 (i.e., $\epsilon = 0$ in Lemma 2.1). Indeed, M is the number of repetition rate, more specifically, repeating from line 1 to 6 in the Sign algorithm. According to [17], it suffices to set $M = \exp(1/2\hat{\alpha}^2)$, where $\hat{\alpha} \leq \sigma / \|\mathbf{Sc}\|$. To set an appropriate value M, [17] (and also NTRU+Sign) first selects an appropriate value $\hat{\alpha}$ that makes M acceptable, for instance, $M = 3.25$ or $M = 4.28$, and next sets σ to be $\hat{\alpha} \cdot \|\mathbf{Sc}\|$, using the equality of $\sigma = \hat{\alpha} \cdot \|\mathbf{Sc}\|$. This indicates that, for a fixed $\hat{\alpha}$ (and thus M), reducing the bound on $\|\mathbf{Sc}\|$ lowers σ, leading to a more compact signature.

3.5 Secret Key Constraint

To bound on $\|\mathbf{Sc}\|$ tightly, we follow the method of HAETAE [13] based on the *canonical embedding* into $\mathbb{C}^n$, which is a classical concept from algebraic number theory. In the canonical embedding, both addition and multiplication in R_q correspond to their coordinate-wise counterparts in $\mathbb{C}^n$, yielding tight bounds on geometric quantities such as Euclidean norms and inner products [33]. Under the canonical embedding, $\|\mathbf{Sc}\|^2$ can be represented as $\|\mathbf{Sc}\|^2 = (\sum_{j=1}^{n} \|\mathbf{S}(w_j)\|^2 \cdot \|\mathbf{c}(w_j)\|^2)/n$, where w_j is the $2n$-th primitive root of unity for $1 \leq j \leq n$. To achieve a tight bound, [13] uses m-largest values of $\|\mathbf{S}(w_j)\|^2$ where $m = \lfloor n/\tau \rfloor$, rather than $\max_j(\|\mathbf{S}(w_j)\|^2)$. Additionally, to ensure that the bound holds regardless of the specific choice of $\mathbf{c}$ (i.e., to eliminate the terms related to $\mathbf{c}$), the right-hand side of the equation above is bounded in a worst-case manner.

Lemma 3.1 ([13])**.** For any $\mathbf{c} \in \{0,1\}^n$ with Hamming weight τ and a secret $\mathbf{S} \in R$, the value $\|\mathbf{Sc}\|^2$ is upper bounded by

$$\frac{\tau}{n} \cdot \mathcal{N}(\mathbf{S}) = \frac{\tau}{n} \Big(\sum_{i=0}^{m} \max_{0 \leq j < n}^{i\text{-}th} \|\mathbf{S}(w_j)\|^2 + r \cdot \max_{0 \leq j < n}^{(m+1)\text{-}th} \|\mathbf{S}(w_j)\|^2 \Big)$$

where $m = \lfloor n/\tau \rfloor, r = n \mod \tau$, and w_j's are the $2n$-th primitive roots of unity.

Using this lemma, we obtain a direct upper-bound on $\|\mathbf{Sc}\|$ by simply fixing $\mathcal{N}(\mathbf{S}) \leq \gamma^2 n$ in line 4 of the KeyGen algorithm for some positive real number γ.

3.6 Security Proof

Theorem 3.2 (UF-CMA Security [6])**.** Assume that NTRU+ident $= ((\mathsf{P}_1, \mathsf{P}_2), (\mathsf{V}_1, \mathsf{V}_2))$ is a paHVZK public-coin identification protocol with probability of aborting β and that the commitment message of the prover has min-entropy α, described in Figure 2. Let $\mathcal{A}$ be any arbitrary adversary against UF-CMA security of DS = FS[NTRU+ident, H] that issues at most Q_h queries to the random oracle H and Q_s classical queries

$\underline{\mathsf{P}_1(sk = (\mathbf{s}_1, \mathbf{s}_2))}$

1: $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \leftarrow \mathcal{D}_\sigma^n \times \mathcal{D}_\sigma^n$
2: $\mathbf{u} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$
3: $\mathbf{w} = [\mathbf{u}]_d$
4: **return** $(W = \mathbf{w}, st = (\mathbf{u}, \mathbf{y}))$

$\underline{\mathsf{V}_1(pk = \mathbf{a}, \mathbf{w})}$

1: $\mathbf{c} \leftarrow U(\mathcal{C})$
2: **return** $\mathbf{c}$

$\underline{\mathsf{P}_2(sk = \mathbf{S}, \mathbf{w}, \mathbf{c}, st)}$

1: $b \leftarrow U(\{0, 1\})$
2: $\mathbf{z} := (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b \mathbf{S}\mathbf{c}$
3: With prob. $1/(M\exp(-\frac{\|\mathbf{S}\mathbf{c}\|^2}{2\sigma^2})\cosh(\frac{\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle}{\sigma^2}))$:
4: **if** $[\mathbf{u}]_d = [\mathbf{u} + (-1)^b \mathbf{c}]_d$
5: $\mathbf{h} := [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1 - b)\mathbf{c}]_d \bmod p$
6: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\| \leq B_2$
7: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty \leq B_\infty$
8: **return** $(\mathbf{z}_1, \mathbf{h})$
9: **else return** $\perp$

$\underline{\mathsf{V}_2(pk, (\mathbf{w}, \mathbf{c}, \mathbf{z}_1, \mathbf{h}))}$

1: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\| \leq B_2$, $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty \leq B_\infty$
2: **if** $[\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} = \mathbf{w} \bmod p$
3: **return** Accept
4: **return** Reject

Figure 2: NTRU+ident protocol

to the signing oracle. Then there exists an adversary $\mathcal{B}$ against UF-NMA security of DS with running time $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + Q_s T_{\mathsf{Sim}}$ such that:

$$\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{DS}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{DS}}(\mathcal{B}) + \frac{2^{-\alpha/2+1} Q_s}{1-\beta} \sqrt{Q_h + 1 + \frac{Q_s}{1-\beta}}$$
$$+ 2^{-\alpha/2+1}(Q_h + 1)\sqrt{\frac{Qs}{1-\beta}} + Q_s \epsilon_{zk}.$$

We recall that an identification scheme ID can be transformed into a digital signature via the Fiat-Shamir transform, and let $\mathsf{FS}[\mathsf{ID}, H]$ denote the resulting signature scheme (see [16] for more details). Based on the analysis of [14], which reduces UF-CMA security to UF-NMA security, we need to show that the commitment min-entropy α is high and the underlying identification scheme is paHVZK. The underlying identification, with repetition parameter $M \geq 1$, Euclidean norm bound B_2 and infinite norm bound B_∞ is given in Figure 2.

3.6.1 paHVZK

In this section, we show that the underlying NTRU+ident scheme from Figure 2 satisfies the paHVZK property in the non-aborting case. Specifically, we demonstrate that the statistical distance ϵ_{zk} between the output $(w, c, z) = (\mathbf{w}, \mathbf{c}, (\mathbf{z}_1, \mathbf{h}))$ distributions of Trans and Sim from Figure 3 is 0, following the approach used in previous works such as [19, 13, 5].

First, it is straightforward to observe that $\mathbf{z}_1$ in the Trans algorithm and $\mathbf{z}_1$ in the Sim algorithm are statistically identical, based on Lemma 2.1 and the analysis in Chapter 3.4. Next, we show that the distribution of the commitment $\mathbf{w}$ in the Trans algorithm and the distribution of $\mathbf{w}$ in the Sim algorithm are identical. To achieve this, we show that the condition in line 7 in the Trans algorithm corresponds to the condition in line

$\underline{\mathsf{Trans}(sk = (\mathbf{s}_1, \mathbf{s}_2), \mathbf{c})}$

1: $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2) \leftarrow \mathcal{D}_\sigma^n \times \mathcal{D}_\sigma^n$
2: $\mathbf{u} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$
3: $\mathbf{w} = [\mathbf{u}]_d$
4: $b \leftarrow \{0, 1\}$
5: $\mathbf{z} := (\mathbf{z}_1, \mathbf{z}_2) = \mathbf{y} + (-1)^b \mathbf{S}\mathbf{c}$
6: With prob. $1/(M\exp(-\frac{\|\mathbf{S}\mathbf{c}\|^2}{2\sigma^2})\cosh(\frac{\langle \mathbf{z}, \mathbf{S}\mathbf{c}\rangle}{\sigma^2}))$:
7: **if** $[\mathbf{u}]_d = [\mathbf{u} + (-1)^b \mathbf{c}]_d$
8: $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p$
9: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\| \leq B_2$
10: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty \leq B_\infty$
11: **return** $(\mathbf{w}, \mathbf{z}_1, \mathbf{h})$
12: **else return** $\perp$

$\underline{\mathsf{Sim}(pk, \mathbf{c})}$

1: With probability $1/M$:
2: $\mathbf{z}_1, \mathbf{z}_2 \leftarrow \mathcal{D}_\sigma^n$
3: $\mathbf{v} = \mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}$
4: **if** $[\mathbf{v}]_d = [\mathbf{v} - \mathbf{c}]_d$
5: $\mathbf{h} = [\mathbf{v}]_d - [\mathbf{v} - \mathbf{z}_2]_d \bmod p$
6: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\| \leq B_2$
7: **if** $\|(\mathbf{z}_1, 2^d \mathbf{h})\|_\infty \leq B_\infty$
8: **return** $(\mathbf{w} = [\mathbf{v}]_d, \mathbf{z}_1, \mathbf{h})$
9: **else return** $\perp$

Figure 3: NTRU+ident protocol simulator

5 in the Sim algorithm from Figure 3. Recall that

$$\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q} = \begin{cases} \mathbf{u} + \mathbf{c} & (\text{if } b = 0) \\ \mathbf{u} & (\text{if } b = 1) \end{cases} \tag{1}$$

where $\mathbf{u} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$. Notably, checking if $[\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q} - \mathbf{c}]_d = [\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}]_d \bmod p$ is equivalent to checking if $[\mathbf{u}]_d = [\mathbf{u} + \mathbf{c}]_d \bmod p$ in case that $b = 0$ and $[\mathbf{u} - \mathbf{c}]_d = [\mathbf{u}]_d \bmod p$ in case that $b = 1$. In other words, the equality check simplifies to checking whether $[\mathbf{u}]_d = [\mathbf{u} + (-1)^b \mathbf{c}]_d \bmod p$. Therefore, through this equality check, we guarantee that $\mathbf{w}$ in the Trans algorithm and $\mathbf{w}$ in Sim algorithm are identical, as it always holds that $[\mathbf{u}]_d = [\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}]_d \bmod p$ for both cases that $b = 0$ or 1.

Lastly, we need to prove that the distribution of $\mathbf{h}$ in the Trans algorithm is identical to that in the Sim. To do so, we show that the hint $\mathbf{h}$ in line 8 of the Trans algorithm corresponds to the hint $\mathbf{h}$ in line 5 of the Sim algorithm (as shown in Figure 3). For $b = 1$, we observe that $[\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}]_d - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2]_d \bmod p$ by equation (1). For $b = 0$, it follows that $[\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}]_d - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d = [\mathbf{u} + \mathbf{c}]_d - [\mathbf{u} - \mathbf{z}_2 + \mathbf{c}]_d \bmod p$ again by equation (1). Given the equality $[\mathbf{u}]_d = [\mathbf{u} + \mathbf{c}]_d \bmod p$ when $b = 0$, the latter expression can be rewritten as $[\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + \mathbf{c}]_d \bmod p$. In summary, we conclude $[\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{c}\hat{q}]_d - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p$. This completes the proof, confirming that the distributions of $\mathbf{h}$ in both algorithms are identical.

3.6.2 Commitment Min-Entropy

We claim that the underlying identification scheme has large commitment min-entropy by showing that, for a given public polynomial $\mathbf{a} \in R_q$,

$$\Pr\left[[\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2]_d = \mathbf{w} \,|\, \mathbf{y}_1, \mathbf{y}_2 \leftarrow D_\sigma^n\right] \leq \left(\frac{2^d}{\sqrt{2\pi}\sigma - 1}\right)^n \quad \forall \mathbf{w} \in \mathcal{W}.$$

Define T to be the set containing all the elements $\mathbf{u}$ such that $[\mathbf{u}]_d = \mathbf{w}$. By the definition of the function $[\cdot]_d$, the size of T is at most 2^{dn}. We can rewrite the above probability as

$$\Pr\left[\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2 \in T \,|\, \mathbf{y}_1, \mathbf{y}_2 \leftarrow D_\sigma^n\right] = \Pr\left[\mathbf{y}_2 \in (T - \mathbf{a}\mathbf{y}_1) \,|\, \mathbf{y}_2 \leftarrow D_\sigma^n\right]$$

where the equality follows from the fact that the size of the set $(T - \mathbf{a}\mathbf{y}_1)$ is the same as the size of T. Finally, using $\sum_{x\in\mathbb{Z}} \rho_\sigma(x) \geq \int_{-\infty}^{\infty} \rho_\sigma(x)dx - 1 = \sqrt{2\pi}\sigma - 1$ [31, Lemma 4.4], we can derive the inequality

$$\Pr\left[\mathbf{y}_2 = \mathbf{t} \,|\, \mathbf{y}_2 \leftarrow D_\sigma^n\right] \leq \left(\frac{1}{\sqrt{2\pi}\sigma - 1}\right)^n \quad \text{for any } \mathbf{t} \in R_q,$$

which in turn implies the claim. For example, the commitment min-entropy α is at least 989 for NTRU+Sign-1024.

3.6.3 UF-NMA Security

We prove that NTRU+Sign is UF-NMA secure if the NTRU assumption and BimodalSelfTargetRSIS assumption hold.

Theorem 3.3 (UF-NMA Security)**.** For any quantum adversary $\mathcal{A}$ against the UF-NMA security of NTRU+Sign making at most Q_h quantum hash queries, there exists an adversary $\mathcal{B}$ and $\mathcal{C}$ such that:

$$\mathsf{Adv}_{\mathsf{DS}}^{\mathsf{UF\text{-}NMA}}(\mathcal{A}) \leq \mathsf{Adv}_{n,q,\psi_1}^{\mathsf{NTRU}}(\mathcal{B}) + \mathsf{Adv}_{H,n,q,B_2+(2^{d-1}+1)\sqrt{n}}^{\mathsf{BimodalSelfTargetRSIS}}(\mathcal{C}). \tag{2}$$

Proof. Given an element $\mathbf{a} \in R_q$, the adversary $\mathcal{C}$ sets $\mathbf{a}$ as the public key of the signature scheme and sends it to $\mathcal{A}$. If the public key pk generated by KeyGen is indistinguishable from uniform over R_q (i.e., if the $\mathsf{NTRU}_{n,q,\psi_1}$ problem is hard), then with probability $\mathsf{Adv}_{\mathsf{DS}}^{\mathsf{UF\text{-}NMA}}(\mathcal{A})$, $\mathcal{A}$ will return a signature $(\mathbf{z}_1, \mathbf{h}, \mathbf{c})$ of some message μ such that $\|(\mathbf{z}_1|2^d\mathbf{h})\| \leq B_2$ satisfying the verification equation

$$\mathbf{c} = H([\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} \bmod p, \mu).$$

For simplicity, we define the function H' such that $H(w_1, \mu) = H'(w_1 \cdot 2^d, \mu)$ to accommodate the compression that does not appear in the BimodalSelfTargetRSIS problem. Notice that the difference between H and H' is just a change in the format of the input. The above equality can be rewritten as:

$$\mathbf{c} = H'(\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q} + 2^d\mathbf{h} + \mathbf{e} + \mathbf{k} \bmod q, \mu) \tag{3}$$

where $\mathbf{e} = \mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q} - [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d$, $\|\mathbf{e}\|_\infty \leq 2^{d-1}$, and $\|\mathbf{k}\|_\infty = 1$. The term $\mathbf{k}$ is added for the following reason. Define $\mathbf{v}_1 = [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} \bmod p$. It can be shown that there exists $\mathbf{k}$ such that the following equation holds over R:

$$\mathbf{v}_1 = [\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} + p\mathbf{k}$$

where all coefficients of $\mathbf{k}$ are in $\{0, \pm 1\}$. Then we obtain $2^d\mathbf{v}_1 = 2^d([\mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q}]_d + \mathbf{h} + p\mathbf{k}) = \mathbf{a}\mathbf{z}_1 + \mathbf{c}\hat{q} + 2^d\mathbf{h} + \mathbf{e} + 2^d \cdot p\mathbf{k}$. Note that it holds that $2^d \cdot p = q - 1$. Applying the modulo q operation to both sides, we derive equation (3).

Equation (3) provides a polynomial vector $\mathbf{Y} = [2^d\mathbf{h} + \overline{\mathbf{e}}|\mathbf{z}_1]^T$ such that $H'([1|\mathbf{a}] \cdot \mathbf{Y} + \mathbf{c}\hat{q}, \mu) = \mathbf{c}$ where $\overline{\mathbf{e}} = \mathbf{e} + \mathbf{k}$. We can show that $\|\mathbf{Y}\|_\infty \leq (q-2)/4$ since the condition $\|(\mathbf{z}_1, 2^d\mathbf{h})\|_\infty \leq B_\infty$ is enforced in the Sign algorithm, where $B_\infty \leq (q-2)/4 - 2^{d-1} - 1$. Next, we note that $\|\mathbf{Y}\| = \|(2^d\mathbf{h} + \overline{\mathbf{e}} + \mathbf{k}, \mathbf{z}_1)\| \leq \|(2^d\mathbf{h}, \mathbf{z}_1)\| + \|(\overline{\mathbf{e}}, 0)\| \leq B_2 + (2^{d-1} + 1)\sqrt{n}$. This completes the proof. □

Table 2: Requirements for Parameter Selection

Parameter Requirements		
(1)	$n = 2^a$ for $a \in \mathbb{N}$	Degree of a polynomial
(2)	$q \equiv 1 \pmod{n/2}$	Modulus for NTT
(3)	$p = (q-1)/2^d$	Modulus for hint generation
(4)	$(2^d/(\sqrt{2\pi}\sigma - 1))^n \ll 2^{-2\lambda}$	Commitment entropy
(5)	$\binom{n}{\tau} \geq 2^\lambda$	Challenge space

4 Parameter Settings

4.1 Concrete Parameters

Table 2 presents several conditions necessary for choosing the parameters for NTRU+Sign. We provide concrete parameter sets achieving 93 and 211 security bits, with the latter corresponding to NIST level 3, as shown in Table 3. Both parameter sets meet all the requirements outlined in Table 2.

The sizes of the public keys for NTRU+Sign-{512, 1024} are 768 and 1664 bytes, respectively, calculated as $(n\lceil \log q \rceil)/8$ bytes. A signature consists of $(\mathbf{z}_1, \mathbf{h}, \mathbf{c})$, where the high bits of $\mathbf{z}_1$ and $\mathbf{h}$ are encoded using range Asymmetric Numeral System (rANS) encoding [22]. Then, the resulting signature size is approximately $(256 + nd + n\log_2(2\pi e(\sigma/2^d)^2)/8$ bytes, where 256 represents a hash value related to $\mathbf{c}$, nd represents low bits of $\mathbf{z}_1$, and $n\log_2(2\pi e(\sigma/2^d)^2$ represents rANS-encoded values of the high bits of $\mathbf{z}_1$ and $\mathbf{h}$. The sizes of the secret keys are 2336 and 5024 bytes for NTRU+Sign-{512, 1024}, respectively. These are calculated as $(3n\lceil \log q \rceil + 256)/8$ bytes, based on the storage of the secret key in NTT format. If the secret key is not stored in NTT format, the sizes would be 1056 and 2208 bytes for NTRU+Sign-{512, 1024}, respectively, computed as $(n\lceil \log q \rceil + 4n + 256)/8$ bytes.

In the KeyGen algorithm, the process restarts if the generated secret key $\mathbf{S}$ satisfies $\mathcal{N}(\mathbf{S}) > \gamma^2 n$, where γ is a pre-defined constant. The constant γ is chosen to ensure that $\mathbf{S}$ is accepted with the sk acceptance rate in Table 3. Specifically, γ is determined in advance by calculating $\{\mathcal{N}(\mathbf{S})\}$ for randomly sampled secret keys $\{\mathbf{S}\}$, sorting $\{\mathcal{N}(\mathbf{S})\}$, and selecting a threshold value $\mathcal{N}(\mathbf{S}^*)$ such that $\mathcal{N}(\mathbf{S}^*)$ among the sorted $\{\mathcal{N}(\mathbf{S})\}$ corresponds to the desired sk acceptance rate. Once $\mathcal{N}(\mathbf{S}^*)$ is selected, γ is computed from the equation of $\mathcal{N}(\mathbf{S}^*) = \gamma^2 n$. Using Lemma 3.1, we can establish an upper bound for $\|\mathbf{Sc}\|$ for secret keys satisfying $\mathcal{N}(\mathbf{S}) \leq \gamma^2 n$. Specifically, Lemma 3.1 states that for any $\mathbf{c}$ with Hamming weight τ, the bound $\|\mathbf{Sc}\|^2 \leq (\tau/n) \cdot \mathcal{N}(\mathbf{S}) \leq \tau\gamma^2$ holds. Thus, the upper bound for $\|\mathbf{Sc}\|$ is computed as $\gamma\sqrt{\tau}$.

4.2 Concrete Security Analysis

As shown above, the security of NTRU+Sign is based on the RSIS problem and the NTRU problem, specifically relative to $\mathsf{Adv}^{\mathsf{BimodalSelfTargetRSIS}}_{H',n,q,B_2+(2^{d-1}+1)\sqrt{n}}$[4] and $\mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\psi_1}$. Note that while the Euclidean norm bound, $B = B_2 + (2^{d-1}+1)\sqrt{n}$, for RSIS solutions exceeds the modulus q, the infinite norm bound remains below q, specifically at $(q-2)/4$. This ensures that trivial solutions are avoided. To analyze the concrete security of the above RSIS and NTRU problems, we adapt the concrete security analysis conducted in [13]. The concrete security strength of the NTRU problem is expected to be similar to the case of Ring Learning with

[4]By the reduction as shown in Def. 2.9, the security is further reduced to the RSIS problem relative to $\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,4(B_2+(2^{d-1}+1)\sqrt{n})+2\sqrt{\tau}}$, but we follow the analysis in [19] to analyze the security of RSIS, i.e., we analyze the concrete security of $\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,B_2+(2^{d-1}+1)\sqrt{n}}$. Here, $2\sqrt{\tau}$ is dropped since $2\sqrt{\tau}$ is relatively very small compared to $4(B_2+(2^{d-1}+1)\sqrt{n})$.

Table 3: Parameter sets for NTRU+Sign-{512, 1024}

Parameter sets		1	2
n	Degree of a polynomial	512	1,024
q	Modulus	3,329	7,681
τ	Hamming weight of $\mathbf{c}$	20	36
$\lfloor \log_2 \lvert\mathcal{C}\rvert \rfloor$	$\lfloor \log_2 \binom{n}{\tau} \rfloor$	118	221
d	# of dropped bits in the commitment	7	8
M	Expected # of repetitions for rejection sampling	3.25	4.28
M_{Eq}	Expected # of repetitions for equality check	1.17	1.15
M_{total}	$= M \times M_{Eq}$	3.80	4.92
$B_{\mathbf{Sc}}$	Upper-bound for Euclidean norm of $\mathbf{Sc}$	169	341
γ	sk rejection parameter	37.77	56.71
	sk acceptance rate	0.25	0.25
σ	Standard deviation of D_σ	110	200
$\hat{\alpha}$	$= \sigma / B_{\mathbf{Sc}}$	0.61	0.58
α	Min-entropy of the commitment	564	989
B_2	Verification threshold in Euclidean norm	4,000	10,000
B_∞	Verification threshold in infinite norm	766	1,790
NTRU Hardness (Core-SVP)			
BKZ block-size b		339	724
Classical Core-SVP		98	211
Quantum Core-SVP		88	186
SIS Hardness (Core-SVP)			
BKZ block-size b		320	767
Classical Core-SVP		93	224
Quantum Core-SVP		82	197

Error (LWE) problems, following the methodology outlined in the script of Kyber[10]. To analyze the concrete security of the above RSIS and NTRU problems, we employ the BKZ lattice reduction algorithm [12], which represents the most effective known lattice attack. Various approaches exist to estimate the running time of BKZ [1, 2, 12]. In general, a Shortest Vector Problem (SVP) solver is the main building block of the BKZ algorithm. To estimate the number of gates required to solve the RSIS and NTRU problems, as in [13], we follow the established Core-SVP methodology, which assumes that an SVP oracle is required only once in a conservative model, regarding the number of SVP oracle calls that the BKZ algorithm makes.

Table 3 presents the concrete security levels for each set of parameters of NTRU+Sign. Our estimates for the RSIS problem are slightly conservative in that those estimates are calculated under the Euclidean norm bound $B_2 + (2^{d-1} + 1)\sqrt{n}$, not considering the additional infinite norm bound $(q-2)/4$. As noted in [28], such cryptanalysis likely underestimates the true complexity of the RSIS problem. Because the modulus q of NTRU+Sign is relatively small compared to the dimension n and the RSIS bound $B_2 + (2^{d-1} + 1)\sqrt{n}$, we can also use the recent estimator proposed by Ducas *et al.* [18]. The estimated results are 93 and 214 bits of security for the RSIS problem, which slightly reduces the RSIS hardness, but has no impact on the overall target security level of NTRU+Sign-$\{512, 1024\}$.

Table 4: Comparison to previous lattice-based signature schemes

	Classical Security	|Sig| (bytes)	$\|pk\|$ (bytes)	|Sig|+$\|pk\|$ (bytes)	Sampling Distribution
Dilithium-2	123	2,420	1,312	3,732	Hypercube
Dilithium-G-2[1]	118	1,921	800	2,721	Gaussian
HAETAE-120	119	1,474	992	2,466	Hyperball
G + G-120[2]	121	1,677	1,472	3,149	(Convolved) Gaussian
G + G-512[3]	85	1,021	992	2,013	(Convolved) Gaussian
Patronus-120	120	2,070	832	2,902	Polytope
Falcon-512	120	666	897	1,563	-
BLISS-512[4]	87	831	896	1,727	Gaussian
NTRU+Sign-512	93	751	768	1,519	Gaussian
Dilithium-3	182	3,293	1,952	5,245	Hypercube
Dilithium-G-3[1]	183	2,462	1,184	3,646	Gaussian
HAETAE-180	180	2,349	1,472	3,821	Hyperball
G + G-180[2]	178	2,143	1,952	4,095	(Convolved) Gaussian
G + G-1024[3]	178	1,769	2,080	3,849	(Convolved) Gaussian
Patronus-180	182	2,575	1,152	3,727	Polytope
Falcon-1024	273	1,280	1,793	3,073	-
BLISS-1024[4]	178	1,836	1,792	3,628	Gaussian
NTRU+Sign-1024	211	1,551	1,664	3,215	Gaussian

[1] Gaussian version of Dilithium optimized in [15] [2] Module-LWE version of G + G
[3] NTRU version of G + G [4] BLISS with updated security analysis

4.3 Comparison

Table 4 provides a detailed comparison of signature sizes among recent lattice-based signature schemes. For comparison with NTRU+Sign, we include FSwA-based schemes [19, 13, 16, 5, 17], including the re-parametrized BLISS, as well as Falcon [23], which is based on the Hash-and-Sign framework. The results show that NTRU+Sign achieves the smallest signature size among the FSwA-based schemes at a comparable security level. Notably, NTRU+Sign-1024 achieves this compactness despite targeting a higher classical security level than most other schemes (excluding Falcon-1024). For example, the signature size of NTRU+Sign-1024 is approximately 55% smaller than Dilithium-3 and 35% smaller than HAETAE-180. Moreover, NTRU+Sign-1024 achieves the smallest combined size of the signature and public key among FSwA-based schemes. Compared to Dilithium-3, the combined size is approximately 40% smaller, and compared to HAETAE-180, it is approximately 15% smaller. A direct comparison with the re-parametrized BLISS highlights the significant impact of the modulus change and the novel key structure in NTRU+Sign.

The performance comparison between Dilithium, HAETAE, and NTRU+Sign is summarized in Table 1. Compared to HAETAE-180, NTRU+Sign-1024 achieves 1.4 times faster key generation and 1.6 times faster signing speeds. However, compared to Dilithium, NTRU+Sign-1024 is approximately 3.5 times slower for key generation and 2.8 times slower for signing. This is primarily due to the additional key rejection process in the KeyGen algorithm of NTRU+Sign, designed to reduce the norm $\|\mathbf{Sc}\|$ and optimize rejection sampling. The performance difference also stems from the efficiency of Dilithium's uniform hypercube sampling and its simplified rejection sampling, which relies solely on infinite norm checks.

Notably, the Verify algorithm of NTRU+Sign is 2.5 times faster than Dilithium and 3 times faster than

HAETAE. While the verification processes in all three schemes share a similar conceptual foundation, several factors contribute to the superior efficiency of NTRU+Sign. Firstly, NTRU+Sign-1024 uses a smaller modulus, $q = 7681 (\approx 2^{13})$, compared to $q = 8380417 (\approx 2^{23})$ in Dilithium and $q = 64513 (\approx 2^{16})$ in HAETAE. The smaller modulus enables faster polynomial multiplications. Secondly, unlike Dilithium and HAETAE, NTRU+Sign does not require re-generating the public matrix $\mathbf{A} \in R_q^{k \times \ell}$ from a public seed during verification, further reducing computational overhead.

Regarding the Sign algorithm, NTRU+Sign achieves computational improvements over HAETAE primarily due to its use of Gaussian sampling instead of hyperball sampling. Hyperball sampling, while simplifying rejection sampling, inherently relies on discrete Gaussian sampling with extremely large standard deviations (e.g., 2^{72}), high-precision arithmetic, and additional operations such as inverse square root and ℓ_2 norm calculations, all of which introduce significant computational overhead. For example, constant-time Gaussian sampling in NTRU+Sign-1024 achieves a 43% speedup compared to hyperball sampling in HAETAE-180. Also, contrary to what is expected, the rejection sampling process of NTRU+Sign-1024 is about 1.6 times faster than that of HAETAE-180. To perform the rejection sampling process, NTRU+Sign needs to calculate transcendental function approximations based on GALATICS, whereas HAETAE needs to compute Euclidean norm. Fortunately, the coefficients of approximated polynomials are precomputed, so that the rejection sampling process of NTRU+Sign can be done by simple scalar multiplications and shift operations, using the precomputed coefficients.

5 Performance Analysis

5.1 Reference Implementation

We evaluate the performance of NTRU+Sign-$\{512, 1024\}$, using our reference code on a 3.7GHz Intel Core i7-8700k running Ubuntu 20.04 LTS. Table 1 shows the performance of the KeyGen, Sign, and Verify algorithms of NTRU+Sign-$\{512, 1024\}$ in comparison to the reference code implementations of Dilithium[19] and HAETAE [13]. The cycle counts presented in Table 1 are averaged over 10,000 executions of each respective algorithm. The reference code for NTRU+Sign is publicly available[5].

5.2 Implementation Details

5.2.1 Gaussian Sampling

In line 1 of the Sign algorithm in Figure 1, we produce $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2)$, where $\mathbf{y}_1, \mathbf{y}_2$ are polynomials with coefficients sampled according to a centered discrete Gaussian distribution of relatively large standard deviation ($\sigma = 200$ for NTRU+Sign-1024). In essence, we start by sampling x from a discrete Gaussian distribution with a smaller standard deviation $\sigma_1 = \sigma/k$ where $k = 2^{\lfloor \log_2 \sigma \rfloor}$ using a Cumulative Distribution Table (CDT), and next we sample a uniform $y \in \{0, ..., k-1\}$ and accept the candidate Gaussian sample $r = xk + y$ with probability $\exp(-y(y + 2xk)/(2\sigma^2))$, following the technique of BLISS [17]. For instance, we use a CDT with $\sigma_1 = 200/2^7 \approx 1.57$ where $n = 1024, \sigma = 200$, and $k = 2^7$, which only requires 15 table entries. This CDT implementation stores the cumulative probabilities with 86 bits of precision. For the rejection sampling with probability $\exp(-y(y + 2xk)/(2\sigma^2))$, we follow the way of GALATICS to generate a polynomial approximation of $\exp(x/(2\sigma^2))$. Then, under a sufficiently large precision, an approximated polynomial $P_{\exp}^{I_1}(x)$ over the interval $I_1 = [-\lfloor 2\sigma^2 \rfloor, 0]$ can be used to evaluate a

[5] https://github.com/KU-Cryptographic-Protocol-Lab/ntruplus_sign

function value at a point x in constant time. Further details on how the polynomial approximation is constructed and how closely the approximated polynomial matches the real exponential function can be found in the appendix A.1.

5.2.2 Rejection Sampling

Explicitly in line 6 of the Sign algorithm in Figure 1, we need to implement the rejection sampling in constant time. Specifically, we need to sample a bit $\tilde{b}$ from the Bernoulli distribution $\mathcal{B}_\psi$ with parameter $\psi(\mathbf{z}, \mathbf{v}) = 1/(M \exp(-\|\mathbf{v}\|^2/2\sigma^2) \cosh(\langle \mathbf{z}, \mathbf{v} \rangle)/\sigma^2)$ where $M = \exp(1/2\hat{\alpha}^2)$. According to the bit $\tilde{b}$, the Sign algorithm decides whether or not it proceeds with $\mathbf{z}$. To implement this efficiently, we decompose the Bernoulli distribution $\mathcal{B}_\psi$ into the product of $\mathcal{B}_{\psi_1}$ and $\mathcal{B}_{\psi_2}$ where $\psi_1(\mathbf{v}) = 1/(M \exp(-\|\mathbf{v}\|^2/2\sigma^2))$ and $\psi_2(\mathbf{z}, \mathbf{v}) = 1/\cosh(\langle \mathbf{z}, \mathbf{v} \rangle/\sigma^2)$. Following GALATICS, we approximate the functions $\exp$ and $\cosh$ with sufficiently close polynomials, and given an input $(\mathbf{z}, \mathbf{v})$ we calculate those function values as in the Gaussian sampling. Further details on constructing these polynomial approximations and the theoretical justification are provided in Appendix A.2.

5.2.3 Signature Packing

In line of 12 in the Sign algorithm in Figure 1, a signature consists of $(\mathbf{z}_1, \mathbf{h}, \mathbf{c})$, where $\mathbf{z}_1$ and $\mathbf{h}$ are packed to reduce the signature size. Indeed, $\mathbf{z}_1$ is a polynomial whose coefficients follow a discrete Gaussian distribution with standard deviation σ, and the hint $\mathbf{h} = [\mathbf{u}]_d - [\mathbf{u} - \mathbf{z}_2 + (1-b)\mathbf{c}]_d \bmod p$ also nearly follows a discrete Gaussian distribution that is substantially derived from $\mathbf{z}_2$. That is, $\mathbf{z}_1$ and $\mathbf{h}$ have their coefficients distributed according to probability distributions that are concentrated around elements with small ℓ_2 norms. Hence, it is not optimal to simply represent coefficients with their binary representations. Instead, we use the adaptive arithmetic encoding, referred as rANS as described in [22]. This entropic coding method enables efficient compression by closely matching the signature size to its entropy, while maintaining high performance through finite precision integer arithmetic. Our rANS encoding is based on the implementation from [24], which provides an efficient and practical approach for encoding a signature.

References

[1] Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015)

[2] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In: Holz, T., Savage, S. (eds.) 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016. pp. 327–343. USENIX Association (2016)

[3] Bai, S., Galbraith, S.D.: An improved compression technique for signatures based on learning with errors. In: Benaloh, J. (ed.) Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8366, pp. 28–47. Springer (2014). https://doi.org/10.1007/978-3-319-04852-9_2, https://doi.org/10.1007/978-3-319-04852-9_2

[4] Bai, S., Lepoint, T., Roux-Langlois, A., Sakzad, A., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance. Journal of Cryptology **31**, 610–640 (2018)

[5] Bambury, H., Beguinet, H., Ricosset, T., Sageloli, É.: Polytopes in the fiat-shamir with aborts paradigm. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14920, pp. 339–372. Springer (2024). https://doi.org/10.1007/978-3-031-68376-3_11

[6] Barbosa, M., Barthe, G., Doczkal, C., Don, J., Fehr, S., Grégoire, B., Huang, Y., Hülsing, A., Lee, Y., Wu, X.: Fixing and mechanizing the security proof of fiat-shamir with aborts and dilithium. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14085, pp. 358–389. Springer (2023). https://doi.org/10.1007/978-3-031-38554-4_12

[7] Barthe, G., Belaïd, S., Espitau, T., Fouque, P., Rossi, M., Tibouchi, M.: GALACTICS: gaussian sampling for lattice-based constant- time implementation of cryptographic signatures, revisited. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019. pp. 2147–2164. ACM (2019). https://doi.org/10.1145/3319535.3363223

[8] Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006. pp. 390–399. ACM (2006). https://doi.org/10.1145/1180405.1180453

[9] Bootle, J., Delaplace, C., Espitau, T., Fouque, P., Tibouchi, M.: LWE without modular reduction and improved side-channel attacks against BLISS. In: Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11272, pp. 494–524. Springer (2018). https://doi.org/10.1007/978-3-030-03326-2_17

[10] Bos, J.W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018. pp. 353–367. IEEE (2018). https://doi.org/10.1109/EUROSP.2018.00032

[11] Bruinderink, L.G., Hülsing, A., Lange, T., Yarom, Y.: Flush, gauss, and reload - A cache attack on the BLISS lattice-based signature scheme. In: Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9813, pp. 323–345. Springer (2016). https://doi.org/10.1007/978-3-662-53140-2_16

[12] Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 1–20. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_1

[13] Cheon, J.H., Choe, H., Devevey, J., Güneysu, T., Hong, D., Krausz, M., Land, G., Möller, M., Stehlé, D., Yi, M.: HAETAE: shorter lattice-based fiat-shamir signatures. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2024**(3), 25–75 (2024). https://doi.org/10.46586/TCHES.V2024.I3.25-75

[14] Devevey, J., Fallahpour, P., Passelègue, A., Stehlé, D.: A detailed analysis of fiat-shamir with aborts. In: CRYPTO 2023. Lecture Notes in Computer Science, vol. 14085, pp. 327–357. Springer (2023). https://doi.org/10.1007/978-3-031-38554-4_11

[15] Devevey, J., Fawzi, O., Passelègue, A., Stehlé, D.: On rejection sampling in lyubashevsky's signature scheme. In: ASIACRYPT 2022. Lecture Notes in Computer Science, vol. 13794, pp. 34–64. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_2

[16] Devevey, J., Passelègue, A., Stehlé, D.: G+G: A fiat-shamir lattice signature based on convolved gaussians. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 37–64. Springer (2023). https://doi.org/10.1007/978-981-99-8739-9_2

[17] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: CRYPTO 2013. Lecture Notes in Computer Science, vol. 8042, pp. 40–56. Springer (2013). https://doi.org/10.1007/978-3-642-40041-4_3

[18] Ducas, L., Espitau, T., Postlethwaite, E.W.: Finding short integer solutions when the modulus is small. In: Annual International Cryptology Conference. pp. 150–176. Springer (2023)

[19] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2018**(1), 238–268 (2018). https://doi.org/10.13154/TCHES.V2018.I1.238-268

[20] Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient NTRU instantiations. In: Boldyreva, A., Kolesnikov, V. (eds.) Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13940, pp. 65–94. Springer (2023). https://doi.org/10.1007/978-3-031-31368-4_3

[21] Espitau, T., Fouque, P., Gérard, B., Tibouchi, M.: Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp. 1857–1874. ACM (2017). https://doi.org/10.1145/3133956.3134028

[22] Espitau, T., Tibouchi, M., Wallet, A., Yu, Y.: Shorter hash-and-sign lattice-based signatures. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. Lecture Notes in Computer Science, vol. 13508, pp. 245–275. Springer (2022). https://doi.org/10.1007/978-3-031-15979-4_9

[23] Fouque, P., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over NTRU (2017), available: https://falcon-sign.info/falcon.pdf

[24] Giesen, F.: Interleaved entropy coders. CoRR **abs/1402.3392** (2014), http://arxiv.org/abs/1402.3392

[25] Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: Prouff, E., Schaumont, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7428, pp. 530–547. Springer (2012). https://doi.org/10.1007/978-3-642-33027-8_31, https://doi.org/10.1007/978-3-642-33027-8_31

[26] Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: Ntrusign: Digital signatures using the ntru lattice. In: Cryptographers' track at the RSA conference. pp. 122–140. Springer (2003)

[27] Hoffstein, J., Pipher, J., Silverman, J.H.: Ntru: A ring-based public key cryptosystem. In: Proceedings of the Third International Symposium on Algorithmic Number Theory. pp. 267–288 (1998)

[28] Jeudy, C., Roux-Langlois, A., Sanders, O.: Phoenix: hash-and-sign with aborts from lattice gadgets. In: International Conference on Post-Quantum Cryptography. pp. 265–299. Springer (2024)

[29] Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In: EUROCRYPT 2018. Lecture Notes in Computer Science, vol. 10822, pp. 552–586. Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_18

[30] Lyubashevsky, V.: Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In: ASIACRYPT 2009. Lecture Notes in Computer Science, vol. 5912, pp. 598–616. Springer (2009). https://doi.org/10.1007/978-3-642-10366-7_35

[31] Lyubashevsky, V.: Lattice signatures without trapdoors. Cryptology ePrint Archive, Paper 2011/537 (2011), https://eprint.iacr.org/2011/537

[32] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 738–755. Springer (2012). https://doi.org/10.1007/978-3-642-29011-4_43

[33] Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 35–54. Springer (2013). https://doi.org/10.1007/978-3-642-38348-9_3

[34] Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 465–484. Springer (2011). https://doi.org/10.1007/978-3-642-22792-9_26

[35] Pessl, P., Bruinderink, L.G., Yarom, Y.: To BLISS-B or not to be: Attacking strongswan's implementation of post-quantum signatures. In: Thuraisingham, B., Evans, D., Malkin, T., Xu, D. (eds.) Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017,. pp. 1843–1855. ACM (2017). https://doi.org/10.1145/3133956.3134023

[36] Prest, T.: Sharper bounds in lattice-based cryptography using the rényi divergence. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. pp. 347–374. Springer (2017)

[37] Zhang, J., Feng, D., Yan, D.: NEV: faster and smaller NTRU encryption using vector decoding. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 157–189. Springer (2023). https://doi.org/10.1007/978-981-99-8739-9_6

A Discrete Gaussian and Rejection Sampling

Given a probability distribution $\mathcal{D}$, we denote $\widetilde{\mathcal{D}}$ as a probability distribution similar to $\mathcal{D}$. Let $x \sim \mathcal{D}$ mean that x follows the probability distribution $\mathcal{D}$ and $x \dot{\sim} \mathcal{D}$ indicate that x follows a probability distribution similar to $\mathcal{D}$. The relative error of $\mathcal{D}_1$ from $\mathcal{D}_2$ is defined as $\left|\frac{\mathcal{D}_1}{\mathcal{D}_2} - 1\right| = \max_{x \in \mathrm{supp}(\mathcal{D}_1)} \left|\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)} - 1\right|$. For $I \subseteq \mathbb{Z}$ and $x \in I$, let us define $D_{I,\sigma}(x) = \frac{\rho_\sigma(x)}{\rho_\sigma(I)}$ as discrete Gaussian distribution over I with standard deviation σ and center at 0. For $p \in [0,1]$, let $\mathcal{B}_p$ denote a Bernoulli distribution with probability p. For $a, b \in \mathbb{R}$, we define $[a,b]_\theta = \left\{x \in [a,b] : |x| - \lfloor |x| \rfloor = \sum_{i=1}^{\theta} x_i 2^i, x_i \in \{0,1\}\right\}$. Similarly, $(a,b]_\theta$, $[a,b)_\theta$ and $(a,b)_\theta$ are also defined. For a predicate P, $[\![\mathrm{P}]\!]$ returns 1 if P is true, and 0 otherwise. Lastly, we use the notation $f(\delta) \underset{\delta \to 0}{\sim} c \iff f(\delta) \approx c$ only when $|\delta| \approx 0$[6].

A.1 Discrete Gaussian

A.1.1 Algorithms

To sample from the discrete Gaussian distribution $\mathcal{D}_\sigma$, we employ a constant-time discrete Gaussian sampling algorithm whose output distribution closely approximates the target distribution. The constant-time approximated discrete Gaussian sampler is described in Algorithm 1. Briefly explained, the sampler generates a nonnegative integer y from an approximated distribution of the nonnegative Gaussian, which is described in Algorithm 2, and then determines its sign. Here are some relevant parameters:

$$k = 2^{\lfloor \log_2 \sigma \rfloor}, \quad \sigma_1 = \sigma / k, \quad w_1 = \lfloor \tau_1 \sigma_1 \rfloor.$$

Algorithm 1 $\mathsf{ApproxG}_\sigma$

Input None
Output $y \in [-(w_1+1)k + 1, (w_1+1)k - 1]$

$y \leftarrow \mathsf{ApproxG}^+_\sigma(\,)$ // $y \dot{\sim} D_{\mathbb{Z}_{\geq 0},\sigma}$
if $y = 0$
  restart with probability $\frac{1}{2}$
$b \leftarrow \mathcal{U}(\{0,1\})$
$y \leftarrow (2b-1)y$
return y

The approximated sampler $\mathsf{ApproxG}^+_\sigma$ for the nonnegative Gaussian is represented in Algorithm 2. In broad terms, $\mathsf{ApproxG}^+_\sigma$ samples y_1 from a discrete Gaussian distribution with a smaller standard deviation $\sigma_1 = \sigma/k$ and y_0 from the uniform distribution on $[0, k-1]$, and returns $y = y_1 k + y_0$ conditionally.

[6]The notation is derived from [36], and approximate values are used for brevity in this paper.

Algorithm 2 $\mathsf{ApproxG}^+_\sigma$

Input None
Output $y \in [0, (w_1+1)k-1]$

$y_1 \leftarrow \mathsf{ApproxBaseG}^+_{\sigma_1}(\,)$ // $y_1 \dot{\sim} D_{\mathbb{Z}_{\geq 0},\sigma_1}$
$y_0 \leftarrow \mathcal{U}([0, k-1])$
$x \leftarrow -y_0(y_0 + 2y_1 k)$
$b \leftarrow \mathsf{SampleBernExp}^{I_1}_\sigma(x)$ // $b \dot{\sim} \mathcal{B}_{\exp(x/2\sigma^2)}$
if $b = 0$
  restart
$y \leftarrow y_1 k + y_0$
return y

In Algorithm 2, the subroutine $\mathsf{ApproxBaseG}^+_{\sigma_1}$ outputs an integer y_1 which is distributed according to a probability distribution $\widetilde{D}_{[0,w_1],\sigma_1}$ that can be considered as a θ_1-precision version of $D_{[0,w_1],\sigma_1}$:

$$\widetilde{D}_{[0,w_1],\sigma_1}(x) = \begin{cases} \left\lfloor D_{[0,w_1],\sigma_1}(x) \cdot 2^{\theta_1} \right\rfloor \cdot 2^{-\theta_1} & , (x \in [1, w_1]) \\ 1 - \sum_{i=1}^{w_1} \widetilde{D}_{[0,w_1],\sigma_1}(i) & , (x = 0) \end{cases}$$

Note that for sufficiently large $\tau_1 = w_1/\sigma_1$ the output of Algorithm 3 actually is distributed according to $D_{\mathbb{Z}_{\geq 0},\sigma_1}$. In Algorithm 3, $\mathrm{cdt}[i] = 2^{\theta_1} \cdot \Pr[z < i : z \leftarrow \widetilde{D}_{[0,w_1],\sigma_1}]$ for $i \in [0, w_1]$.

Algorithm 3 $\mathsf{ApproxBaseG}^+_{\sigma_1}$

Input None
Output $y_1 \in [0, w_1]$

$u \leftarrow \mathcal{U}([0, 2^{\theta_1}-1])$
$y_1 \leftarrow 0$
for $i = 0$ **to** $w_1 - 1$ **do**
  $y_1 \leftarrow y_1 + [\![\, u > \mathrm{cdt}[i] \,]\!]$
return y_1

Another subroutine $\mathsf{SampleBernExp}^{I_1}_\sigma(x)$ of Algorithm 2 describes an approximated version of the Bernoulli distribution $\mathcal{B}_{\exp(x/2\sigma^2)}$ for $x \in I_1$. To do this, it uses a polynomial approximation $P^{I_1}_{\exp}(x)$ of $\exp(x/2\sigma^2)$ over the interval I_1. Note that the property of $P^{I_1}_{\exp}(x)$ necessary to ensure security is described in following section. As a result, $\mathsf{SampleBernExp}^{I_1}_\sigma(x)$ returns a bit by comparing a uniform random number u in $[0,1)_{\vartheta_1}$ and the output of $\mathsf{EvaluatePoly}[P^{I_1}_{\exp}, \vartheta_1](x)$, which is an approximation of $P^{I_1}_{\exp}(x)$.

In Algorithm 4, the output of $\mathsf{EvaluatePoly}[P^{I_1}_{\exp}, \vartheta_1](x)$ provides the ϑ_1-bit precision value of $P^{I_1}_{\exp}(x) = \sum_{i=0}^{d} p_i x^i$ where $p_0 \in [0,1)_{\vartheta_1}$ and $p_i = \tilde{p}_i \cdot 2^{-\ell_i}$, $\tilde{p}_i \in [0,1)_{\vartheta_1}$, $\ell_i \in \mathbb{N}$ for $i = 1, \ldots, d$. Basically, $P^{I_1}_{\exp}(x) = \sum_{i=0}^{d} p_i x^i$ is computed by adapting Horner's rule as follows:

$$P^{I_1}_{\exp}(x) = \big(((p_d \cdot x + p_{d-1}) \cdot x) + p_{d-2}) \cdot x + \cdots\big) \cdot x + p_0.$$

On the other hand, since $P^{I_1}_{\exp}$ is an approximation of $\exp(-x)$, its coefficients are decreasing in the sene that $1 = |p_0| \gg |p_1| \gg \cdots \gg |p_d|$, i.e, $\ell_0 < \ell_1 < \ell_2 < \cdots < \ell_d$. So the coefficients of $P^{I_1}_{\exp}$ can be expressed as

$$p_i = \tilde{p}_i \cdot 2^{-\sum_{j=1}^{i} s_j}, \quad \tilde{p}_i \in [0,1)_{\vartheta_1}$$

Algorithm 4 $\mathsf{SampleBernExp}_{\sigma}^{I_1}(x)$

Input $x \in I_1 \cap \mathbb{Z}$
Output $b \in \{0, 1\}$

$\hat{p} \leftarrow \mathsf{EvaluatePoly}[P_{\mathrm{exp}}^{I_1}, \vartheta_1](x)$ // $\hat{p} \in [0, 1]_{\vartheta_1}$
$u \leftarrow \mathcal{U}([0, 1)_{\vartheta_1})$
if $u \leq \hat{p}$
  return 1
return 0

where $s_1 = \ell_1$ and $s_j = \ell_j - \ell_{j-1} \in \mathbb{N}$ for $i = 2, \ldots, d$. Using this representation, the Horner's rule can be written as

$$P_{\mathrm{exp}}^{I_1}(x) = \left(((\tilde{p}_d \cdot 2^{-s_d} \cdot x + \tilde{p}_{d-1}) \cdot 2^{-s_{d-1}} \cdot x) + \tilde{p}_{d-2}) \cdot 2^{-s_{d-2}} \cdot x + \cdots\right) \cdot x \cdot 2^{-s_1} + p_0,$$

and Algorithm 5 adapts this representation as the following sequential execution:

$$y_d = \tilde{p}_d \cdot x \cdot 2^{-s_d},\ y_{d-1} = (y_d + \tilde{p}_{d-1}) \cdot x \cdot 2^{-s_{d-1}},\ \ldots\ ,\ y_1 = (y_2 + \tilde{p}_1) \cdot x \cdot 2^{-s_1},\ y_0 = y_1 + p_0.$$

Actually, Algorithm 5 maintains ϑ_1-bit precision during computation using only the integer arithmetic of a fixed size, which results in computing the approximation of y_i. For simplicity, when a polynomial $P = \sum_{i=0}^{d} p_i x^i$ and ϑ are given as the parameters of Algorithm 5, we assume that two sequences $(\tilde{p}_i)_{0 \leq i \leq d}, (s_i)_{0 \leq i \leq d}$ satisfying $p_i = \tilde{p}_i \cdot 2^{-\sum_{j=1}^{i} s_j}$, $\tilde{p}_i \in [0, 1)_{\vartheta}$ for $i = 0, \ldots, d$ are given.

Algorithm 5 $\mathsf{EvaluatePoly}[P_{\mathrm{exp}}^{I_1}, \vartheta_1](x)$

Input $x \in I_1$
Output $\tilde{y} \in [0, 1)_{\vartheta_1}$

$\tilde{y}_d \leftarrow \left\lfloor (\tilde{p}_d \cdot x) \cdot 2^{\vartheta_1 - s_d} \right\rceil \cdot 2^{-\vartheta_1}$
for $i = d - 1$ **to** 1 **do**
  $\tilde{y}_i \leftarrow \left\lfloor ((\tilde{y}_{i+1} + \tilde{p}_i) \cdot x) \cdot 2^{\vartheta_1 - s_i} \right\rceil \cdot 2^{-\vartheta_1}$
$\tilde{y} \leftarrow \tilde{y}_1 + p_0$
return $\tilde{y}$

A.1.2 Constraints for Parameters

Let $Q_{\mathrm{sign}} = 2^{64}$ be the maximum number of signing queries and define $Q_{\mathrm{gauss}} = M \cdot 2n \cdot Q_{\mathrm{sign}}$. Recall that $k = 2^{\lfloor \log_2 \sigma \rfloor}$, $\sigma_1 = \sigma / k$, and $w_1 = \lfloor \tau_1 \sigma_1 \rfloor$.

τ_1 τ_1 is a value used to restrict the support of a discrete Gaussian distribution with a small standard deviation σ_1, implemented through the cumulative distribution table (CDT). Specifically, τ_1 is set to satisfy the following condition:

$$\frac{1}{1 - 2C(\tau_1)} \leq c_{\mathrm{g}} \left(1 + \frac{1}{16 Q_{\mathrm{gauss}}}\right) \tag{4}$$

where $c_{\mathrm{g}} = \sqrt{1 - 1/(2^5 Q_{\mathrm{gauss}})}$ is a constant used for proper scaling to adjust inequalities and is chosen to be a real number close to $1 - 2^{-47}$.

θ_1 θ_1 represents the precision required when approximating $D_{[0,w_1],\sigma_1}$ using CDT. According to the way CDT is used in Algorithm 4, θ_1 needs to be set for satisfying $R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{[0,w_1],\sigma_1}) \leq 1 + \frac{1}{16Q_{\text{gauss}}}$, which holds when

$$2^{\theta_1} \geq w_1 \cdot \rho_{\sigma_1}([0,w_1]) \cdot \left(c_{\text{g}}(1+\tfrac{1}{16Q_{\text{gauss}}}) - 1\right)^{-1} \tag{5}$$

$P^{I_1}_{\text{exp}}$ $P^{I_1}_{\text{exp}}$ is a polynomial approximation for $\exp\left(-\frac{x}{2\sigma^2}\right)$ over $I_1 = (-\lfloor 2\sigma^2 \rfloor, 0]$. The required properties for the approximation polynomial $P^{I_1}_{\text{exp}}$ are as follows:

$$\left| \frac{P^{I_1}_{\text{exp}}(x)}{\exp(x/(2\sigma^2))} - 1 \right| \leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}}, \tag{6}$$

$$\left| \frac{\exp\left(x/(2\sigma^2)\right) - P^{I_1}_{\text{exp}}(x)}{1-\exp(x/(2\sigma^2))} \right| \leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} \tag{7}$$

for all $x \in I_1$. We find a polynomial satisfying (6) by using the method proposed in GALATICS, and (7) is checked experimentally.

ϑ_1 ϑ_1 represents the precision required for evaluating the polynomial $P^{I_1}_{\text{exp}}$ at $x \in I_1$. It needs to satisfy

$$2^{\vartheta_1} \geq \frac{2}{1-\exp(-1/2\sigma^2)}\sqrt{2\lambda Q_{\text{gauss}}}. \tag{8}$$

A.1.3 Security

The Rényi divergence of the output distribution of $\mathsf{ApproxG}_\sigma$ from $D_{\mathbb{Z},\sigma}$ is demonstrated using that of the output distribution of $\mathsf{ApproxG}^+_\sigma$ from $D_{\mathbb{Z}_{\geq 0},\sigma}$ in Theorem A.1. In simple terms, the divergence is affected by how closely the approximation polynomial and CDT matches its real ones.

Theorem A.1. Let $\widetilde{D}_{\mathbb{Z}_{\geq 0},\sigma}$ denote the output distribution of $\mathsf{ApproxG}^+_\sigma$ in Algorithm 2. If σ satisfies $\exp(-\frac{1}{2\sigma^2}) \leq 1 - \frac{1}{2\sigma^2+1}$ and τ_1, θ_1, $P^{I_1}_{\text{exp}}$, θ_{gauss}, $\theta^{I_1}_{\text{poly}}$ satisfy (4), (5), (6), (7), (8), then

$$R_{2\lambda}(\widetilde{D}_{\mathbb{Z}_{\geq 0},\sigma} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma}) \;\leq\; 1 + \frac{1}{4Q_{\text{gauss}}}.$$

Proof. The distribution $\widetilde{D}_{\mathbb{Z}_{\geq 0},\sigma}$ differs from the distribution $D_{\mathbb{Z}_{\geq 0},\sigma}$ by two aspects due to the operation of $\mathsf{ApproxG}^+_\sigma$. Let $\widehat{D}_{\mathbb{Z}\geq 0,\sigma}$ be the output distribution of Algorithm 2 when replacing $\mathsf{ApproxG}^+_{\sigma_1}$ with $D_{\mathbb{Z}\geq 0,\sigma_1}$, and let $\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}$ be the output distribution of Algorithm 4. Then, by Lemma A.2, the Rényi divergence is bounded as

$$\begin{aligned} R_{2\lambda}(\widetilde{D}_{\mathbb{Z}\geq 0,\sigma} \,|\, D_{\mathbb{Z}\geq 0,\sigma}) &\leq R_{2\lambda}(\widetilde{D}_{\mathbb{Z}\geq 0,\sigma} \,|\, \widehat{D}_{\mathbb{Z}\geq 0,\sigma}) \cdot R_\infty(\widehat{D}_{\mathbb{Z}\geq 0,\sigma} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma}) \\ &= R_{2\lambda}(\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)} \,|\, \mathcal{B}_{\exp(x/2\sigma^2)}) \cdot R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}\geq 0,\sigma_1}). \end{aligned}$$

Combining this with Lemma A.9 and Lemma A.3, we have the result:

$$R_{2\lambda}(\widetilde{D}_{\mathbb{Z}\geq 0,\sigma} \,|\, D_{\mathbb{Z}\geq 0,\sigma}) \leq \left(1 + \frac{1}{8Q_{\text{gauss}}}\right) \cdot c_{\text{g}}^2 \left(1 + \frac{1}{16Q_{\text{gauss}}}\right)^2 \leq 1 + \frac{1}{4Q_{\text{gauss}}}.$$

□

Lemma A.2 ([4, Lemma 2.9])**.** Let $a \in [1,\infty)$. For three distributions $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ with $\mathsf{Supp}(\mathcal{D}_i) \subset \mathsf{Supp}(\mathcal{D}_{i+1})$, we have

$$R_a(\mathcal{D}_1 \,|\, \mathcal{D}_3) \leq \begin{cases} R_a(\mathcal{D}_1 \,|\, \mathcal{D}_2) \cdot R_\infty(\mathcal{D}_2 \,|\, \mathcal{D}_3) \\ R_\infty(\mathcal{D}_1 \,|\, \mathcal{D}_2)^{\frac{a}{a-1}} \cdot R_a(\mathcal{D}_2 \,|\, \mathcal{D}_3) \quad \text{if } a \in (1,\infty) \end{cases}.$$

Lemma A.3. Let $\widetilde{D}_{[0,w_1],\sigma_1}$ be the output distribution of $\mathsf{ApproxBaseG}^+_{\sigma_1}$ in Algorithm 3. If τ_1 and θ_1 satisfy

$$\tfrac{1}{1-2C(\tau_1)} \leq c_{\mathrm{g}} \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right) \quad \text{and} \quad 2^{\theta_1} \geq w_1 \cdot \rho_{\sigma_1}([0,w_1]) \cdot c_{\mathrm{g}}^{-1} \cdot \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right)^{-1},$$

then

$$R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma_1}) \;\leq\; c_{\mathrm{g}}^2 \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right)^2.$$

Proof. The result is directly obtained from the following calculations:

$$\begin{aligned} R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma_1}) &= \max_{x \in [0,w_1]} \frac{\widetilde{D}_{[0,w_1],\sigma_1}(x)}{D_{\mathbb{Z}_{\geq 0},\sigma_1}(x)} \\ &\leq \max_{x \in [0,w_1]} \frac{\widetilde{D}_{[0,w_1],\sigma_1}(x)}{D_{[0,w_1],\sigma_1}(x)} \cdot \max_{x \in [0,w_1]} \frac{D_{[0,w_1],\sigma_1}(x)}{D_{\mathbb{Z}_{\geq 0},\sigma_1}(x)} \\ &= R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{[0,w_1],\sigma_1}) \cdot R_\infty(D_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma_1}) \\ &\leq c_{\mathrm{g}}^2 \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right)^2 \end{aligned}$$

where the last inequality comes from Lemma A.4 and Lemma A.6.

□

Lemma A.4. If $2^{\theta_1} \geq w_1 \cdot \rho_{\sigma_1}([0,w_1]) \cdot c_{\mathrm{g}}^{-1} \cdot \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right)^{-1}$, then

$$R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{[0,w_1],\sigma_1}) \;\leq\; c_{\mathrm{g}} \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right).$$

Proof. Recall that $R_\infty(\widetilde{D}_{[0,w_1],\sigma_1} \,|\, D_{[0,w_1],\sigma_1}) = \max\limits_{x \in [0,w_1]} \frac{\widetilde{D}_{[0,w_1],\sigma_1}(x)}{D_{[0,w_1],\sigma_1}(x)}$. Then, by the definition of $\widetilde{D}_{[0,w_1],\sigma_1}$,

it is trivial that $\frac{\widetilde{D}_{[0,w_1],\sigma_1}(x)}{D_{[0,w_1],\sigma_1}(x)} < 1$ for any $x \neq 0$. In addition, one can easily see that

$$\begin{aligned}
\frac{\widetilde{D}_{[0,w_1],\sigma_1}(0)}{D_{[0,w_1],\sigma_1}(0)} &= \left(1 - \sum_{i=1}^{w_1} \lfloor D_{[0,w_1],\sigma_1}(i) \cdot 2^{\theta_1} \rfloor \cdot 2^{-\theta_1}\right) \cdot \frac{\rho_{\sigma_1}([0,w_1])}{\rho_{\sigma_1}(0)} \\
&< \left(1 - \sum_{i=1}^{w_1} \left(D_{[0,w_1],\sigma_1}(i) \cdot 2^{\theta_1} - 1\right) \cdot 2^{-\theta_1}\right) \cdot \rho_{\sigma_1}([0,w_1]) \\
&= \left(1 - \sum_{i=1}^{w_1} \left(D_{[0,w_1],\sigma_1}(i) - 2^{-\theta_1}\right)\right) \cdot \rho_{\sigma_1}([0,w_1]) \\
&= \rho_{\sigma_1}([0,w_1]) - \sum_{i=1}^{w_1} \rho_{\sigma_1}(i) + w_1 \cdot 2^{-\theta_1} \cdot \rho_{\sigma_1}([0,w_1]) \\
&= 1 + w_1 \cdot 2^{-\theta_1} \cdot \rho_{\sigma_1}([0,w_1]) \\
&\leq c_{\mathrm{g}} \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right).
\end{aligned}$$

□

Lemma A.5 ([34, Section 2.3])**.** For random $n \in \mathbb{Z}_{>0}$, $\sigma \in \mathbb{R}_{>0}$, $\tau > 1 \in \mathbb{R}$,

$$\rho_\sigma(\mathbb{Z}^n \backslash \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq \tau\sigma\sqrt{n}\}) < 2C(\tau)^n \cdot \rho_\sigma(\mathbb{Z})^n$$

with $C(\tau) = \tau \exp\left(\frac{1-\tau^2}{2}\right) < 1$.

Lemma A.6. If τ_1 satisfies $\frac{1}{1-2C(\tau_1)} \leq c_{\mathrm{g}} \left(1 + \frac{1}{16Q_{\mathrm{gauss}}}\right)$, then

$$R_\infty(D_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma_1}) \leq c_{\mathrm{g}} \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right).$$

Proof. Using the inequality that $\frac{a}{b} < \frac{a-1/2}{b-1/2}$ for $a > b > 1$, we have

$$\begin{aligned}
R_\infty(D_{[0,w_1],\sigma_1} \,|\, D_{\mathbb{Z}_{\geq 0},\sigma_1}) &= \max_{x \in [0,w_1]} \frac{D_{[0,w_1],\sigma_1}(x)}{D_{\mathbb{Z}_{\geq 0},\sigma_1}(x)} = \frac{\rho_{\sigma_1}(\mathbb{Z}_{\geq 0})}{\rho_{\sigma_1}([0,w_1])} \\
&\leq \frac{\rho_{\sigma_1}(\mathbb{Z}_{\geq 0}) - 1/2}{\rho_{\sigma_1}([0,w_1]) - 1/2} = \frac{\rho_{\sigma_1}(\mathbb{Z})}{\rho_{\sigma_1}([-w_1,w_1])} \\
&\leq \frac{1}{1 - 2C(\tau_1)} \\
&\leq c_{\mathrm{g}} \left(1 + \tfrac{1}{16Q_{\mathrm{gauss}}}\right),
\end{aligned}$$

where the second inequality comes from the last holds by Lemma A.5.

□

Lemma A.7. Let $\widetilde{P}(x)$ denote the output of Algorithm 5 with parameters P and ϑ. If $\tilde{p}_1 \in [\frac{1}{2}, 2^{s_1 - \log_2 \lfloor 2\sigma^2 \rfloor})$, $\log_2 \lfloor 2\sigma^2 \rfloor - s_1 \leq 1$, and $\lceil \log_2 \lfloor 2\sigma^2 \rfloor \rceil - s_i \leq -1$, $\tilde{p}_i \in [\frac{1}{2}, 1)_\vartheta$ for $i \in \{2, \ldots, d\}$, then

$$\left| \widetilde{P}(x) - P(x) \right| \leq 2^{-(\vartheta+1)} \quad \text{and} \quad \tilde{P}(x) \in [0,1]_\vartheta$$

for all $x \in I \subseteq [-\lfloor 2\sigma^2 \rfloor, 0]$.

Proof. Let $\eta = \lceil \log_2 \lfloor 2\sigma^2 \rfloor \rceil$. For $x \in I_1$, we have

$$\begin{aligned} |y_d - \tilde{y}_d| &= \left| (\tilde{p}_d \cdot x) \cdot 2^{-s_d} - \lfloor (\tilde{p}_d \cdot x) \cdot 2^{\vartheta - s_d} \rceil \cdot 2^{-\vartheta} \right| \\ &= 2^{-\vartheta} \cdot \left| (\tilde{p}_d \cdot x) \cdot 2^{\vartheta - s_d} - \lfloor (\tilde{p}_d \cdot x) \cdot 2^{\vartheta - s_d} \rceil \right| \\ &\leq 2^{-(\vartheta+1)}. \end{aligned}$$

Moreover, since $\tilde{p}_d \in [\frac{1}{2}, 1)_\vartheta$ and $x \in (-2^\eta, 0]$, we have $\lfloor (\tilde{p}_d \cdot x) \cdot 2^{\vartheta - s_d} \rceil \in [-2^{\eta + \vartheta - s_d}, 0]$, which in turn implies $\tilde{y}_d = \lfloor (\tilde{p}_d \cdot x) \cdot 2^{\vartheta - s_d} \rceil \cdot 2^{-\vartheta} \in [-2^{\eta - s_d}, 0] \subseteq [-\frac{1}{2}, 0]_\vartheta$ by $\eta - s_d \leq -1$.

Now, let us consider the case that $i = d - 1$. Given that $\tilde{p}_{d-1} \in [\frac{1}{2}, 1)_\vartheta$ and $\tilde{y}_d \in [-\frac{1}{2}, 0]_\vartheta$, we have $\tilde{y}_d + \tilde{p}_{d-1} \in [0, 1)_\vartheta$. As before, letting $\tilde{z}_{d-1} = \tilde{y}_d + \tilde{p}_{d-1}$, we then have

$$\begin{aligned} |y_{d-1} - \tilde{y}_{d-1}| &= \left| (\tilde{z}_{d-1} \cdot x) \cdot 2^{-s_{d-1}} - \lfloor (\tilde{z}_{d-1} \cdot x) \cdot 2^{\vartheta - s_{d-1}} \rceil \cdot 2^{-\vartheta} \right| \\ &= 2^{-\vartheta} \cdot \left| (\tilde{z}_{d-1} \cdot x) \cdot 2^{\vartheta - s_{d-1}} - \lfloor (\tilde{z}_{d-1} \cdot x) \cdot 2^{\vartheta - s_{d-1}} \rceil \right| \\ &\leq 2^{-(\vartheta+1)} \end{aligned}$$

and $\tilde{y}_{d-1} \in [-\frac{1}{2}, 0]_\vartheta$ by $\eta - s_{d-1} \leq -1$.

Since $\tilde{y}_i$ and y_i are computed in the same manner under the identical conditions on $\tilde{p}_i$ and $\tilde{y}_{i+1}$ from $i = d - 1$ down to 1, we have $|y_1 - \tilde{y}_1| \leq 2^{-\vartheta+1}$, and thus

$$|\widetilde{P}(x) - P(x)| = |y_0 - \tilde{y}_0| = |(\tilde{p}_0 + y_1) - (\tilde{p}_0 + \tilde{y}_1)| = |y_1 - \tilde{y}_1| \leq 2^{-(\vartheta+1)}.$$

Since $\tilde{y}_2 \in [-\frac{1}{2}, 0]_\vartheta$ and $\tilde{p}_1 \in [\frac{1}{2}, 2^{s_1 - \log_2 \lfloor 2\sigma^2 \rfloor})$, we have $((\tilde{y}_2 + \tilde{p}_1) \cdot x) \cdot 2^{\vartheta - s_1} \in (-2^\vartheta, 0]$, which implies $\tilde{y}_1 = \lfloor ((\tilde{y}_2 + \tilde{p}_1) \cdot x) \cdot 2^{\vartheta - s_1} \rceil \cdot 2^{-\vartheta} \in [-1, 0]_\vartheta$. Thus, $\tilde{P}(x) = \tilde{y}_0 = 1 + \tilde{y}_1 \in [0, 1]_\vartheta$.

□

Lemma A.8 ([36, Lemma 3])**.** Assume that two distributions, $\mathcal{D}_1$ and $\mathcal{D}_2$, satisfy the following conditions:

- $\mathrm{supp}(\mathcal{D}_1) = \mathrm{supp}(\mathcal{D}_2)$
- $\exists\, \delta > 0$ such that $\left| \frac{\mathcal{D}_1}{\mathcal{D}_2} - 1 \right| \leq \delta$ over $\mathrm{supp}(\mathcal{D}_1)$

Given that $a \in (1, \infty)$,

$$R_a(\mathcal{D}_1 \,|\, \mathcal{D}_2) \leq \left(1 + \frac{a(a-1)\delta^2}{2(1-\delta)^{a+1}} \right)^{\frac{1}{a-1}} \underset{\delta \to 0}{\sim} 1 + \frac{a\delta^2}{2}.$$

Lemma A.9. Let $P_{\text{exp}}^{I_1}(x)$ be a approximate polynomial for $\exp(x/2\sigma^2)$ over $x \in I_1 = [-\lfloor 2\sigma^2 \rfloor, 0]$ satisfying $P_{\text{exp}}^{I_1}(0) = 1$. Assume that $1 - \exp\left(-\frac{1}{2\sigma^2}\right) \leq \frac{1}{e}$. If $P_{\text{exp}}^{I_1}(x)$ satisfies

$$\left|\frac{P_{\text{exp}}^{I_1}(x)}{\exp(x/2\sigma^2)} - 1\right| \leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} \quad \text{and} \quad \left|\frac{\exp(x/2\sigma^2) - P_{\text{exp}}^{I_1}(x)}{1-\exp(x/2\sigma^2)}\right| \leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}}$$

for all $x \in I_1 \cap \mathbb{Z}$, then

$$R_{2\lambda}(\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)} \,|\, \mathcal{B}_{\exp(x/2\sigma^2)}) \;\leq\; 1 + \frac{1}{8Q_{\text{gauss}}}$$

where $\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}$ denote the output distribution of Algorithm 4.

Proof. Note that $1 + \frac{2\lambda\delta^2}{2} \leq 1 + \frac{1}{8Q_{\text{gauss}}} \Leftrightarrow \delta \leq \frac{1}{2\sqrt{2\lambda Q_{\text{gauss}}}}$. From Lemma A.8, it is enough to show that the relative error

$$\delta := \left|\frac{\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}}{\mathcal{B}_{\exp(x/2\sigma^2)}} - 1\right| = \max\left(\left|\frac{\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(0)}{\mathcal{B}_{\exp(x/2\sigma^2)}(0)} - 1\right|, \left|\frac{\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)} - 1\right|\right)$$

is bounded by $\frac{1}{2\sqrt{2\lambda Q_{\text{gauss}}}}$.

Let $\widetilde{P}_{\text{exp}}^{I_1}(x)$ denote the output of Algorithm 5 with parameters $P_{\text{exp}}^{I_1}, \vartheta_1$. Note that $\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)} = \mathcal{B}_{\exp(x/2\sigma^2)}$ for $x = 0$ since $P_{\text{exp}}^{I_1}(0) = 1$. Now, we only consider the case that $x \in [-\lfloor 2\sigma^2 \rceil, -1]$. Since $|\widetilde{P}_{\text{exp}}^{I_1}(x) - P_{\text{exp}}^{I_1}(x)| < \frac{1-\exp(-1/2\sigma^2)}{2^2\sqrt{2\lambda Q_{\text{gauss}}}}$ by Lemma A.7 and (8) of ϑ_1, we have

$$\begin{aligned}
\left|\frac{\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(0)}{\mathcal{B}_{\exp(x/2\sigma^2)}(0)} - 1\right| &= \left|\frac{1-\widetilde{P}_{\text{exp}}^{I_1}(x)}{1-\exp(x/2\sigma^2)} - 1\right| = \left|\frac{\widetilde{P}_{\text{exp}}^{I_1}(x) - \exp\left(x/2\sigma^2\right)}{1-\exp(x/2\sigma^2)}\right| \\
&\leq \left|\frac{\widetilde{P}_{\text{exp}}^{I_1}(x) - P_{\text{exp}}^{I_1}(x)}{1-\exp(x/2\sigma^2)}\right| + \left|\frac{P_{\text{exp}}^{I_1}(x) - \exp\left(x/2\sigma^2\right)}{1-\exp(x/2\sigma^2)}\right| \\
&\leq \left|\frac{\widetilde{P}_{\text{exp}}^{I_1}(x) - P_{\text{exp}}^{I_1}(x)}{1-\exp(-1/2\sigma^2)}\right| + \left|\frac{P_{\text{exp}}^{I_1}(x) - \exp\left(x/2\sigma^2\right)}{1-\exp(x/2\sigma^2)}\right| \\
&\leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} + \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}}
\end{aligned}$$

and

$$\begin{aligned}
\left|\frac{\widetilde{\mathcal{B}}_{\exp(x/(2\sigma^2))}(1)}{\mathcal{B}_{\exp(x/(2\sigma^2))}(1)} - 1\right| &= \left|\frac{\widetilde{P}_{\text{exp}}^{I_1}(x)}{\exp(x/(2\sigma^2))} - 1\right| \\
&\leq \left|\frac{\widetilde{P}_{\text{exp}}^{I_1}(x)}{\exp(x/(2\sigma^2))} - \frac{P_{\text{exp}}^{I_1}(x)}{\exp(x/(2\sigma^2))}\right| + \left|\frac{P_{\text{exp}}^{I_1}(x)}{\exp(x/(2\sigma^2))} - 1\right| \\
&\leq e\left|\widetilde{P}_{\text{exp}}^{I_1}(x) - P_{\text{exp}}^{I_1}(x)\right| + \left|\frac{P_{\text{exp}}^{I_1}(x)}{\exp(x/(2\sigma^2))} - 1\right| \\
&\leq \frac{e(1-\exp\left(-1/2\sigma^2\right))}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} + \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} \\
&\leq \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}} + \frac{1}{2^2\sqrt{2\lambda Q_{\text{gauss}}}}.
\end{aligned}$$

Then, these two inequalities show that the relative error is bounded by $\frac{1}{2\sqrt{2\lambda Q_{\text{gauss}}}}$.

□

A.1.4 Concrete parameters

The parameters for the constant-time discrete Gaussian sampler are set as follows:

parameter set	1	2
λ	93	211
n	512	1024
B_{sc}	169	341
σ	110	200
$\hat{\alpha}\,(=\sigma/B_{sc})$	0.61	0.58
$M\,(=\exp(1/2\hat{\alpha}^2))$	3.25	4.28
$k\,(=2^{\lfloor \log_2 \sigma \rfloor})$	2^6	2^7
$\sigma_1\,(=\sigma/k)$	1.71875	1.5625
τ_1	9	9
$w_1\,(=\lfloor \tau_1\sigma_1 \rfloor)$	15	14
θ_1	86	86
ϑ_1	58	60
$P_{\text{exp}}^{I_1}$	P_{g512}	P_{g1024}

Table 5: Parameters for constant-time discrete Gaussian sampler

i	cdt$[i]$
0	29,151,226,717,600,037,870,661,691
1	53,763,390,821,665,321,263,532,167
2	68,575,902,413,396,088,331,456,359
3	74,930,580,255,359,372,543,360,029
4	76,873,900,198,393,993,982,160,770
5	77,297,524,641,422,720,945,905,655
6	77,363,351,578,546,129,243,855,570
7	77,370,643,000,298,910,318,530,342
8	77,371,218,714,211,971,522,174,381
9	77,371,251,117,356,994,444,585,424
10	77,371,252,417,387,484,525,835,338
11	77,371,252,454,567,181,491,505,901
12	77,371,252,455,325,137,652,441,583
13	77,371,252,455,336,152,240,596,897
14	77,371,252,455,336,266,338,686,520
15	77,371,252,455,336,267,181,195,263

Figure 4: CDT for parameter set 1

i	cdt$[i]$
0	31,473,435,849,184,983,252,505,562
1	57,118,314,364,810,086,106,862,155
2	70,991,305,743,784,588,362,155,679
3	75,973,866,085,059,201,192,707,076
4	77,161,952,411,522,559,008,139,834
5	77,350,038,384,741,789,294,624,734
6	77,369,807,070,584,075,157,898,075
7	77,371,186,540,301,596,754,123,086
8	77,371,250,448,933,808,020,018,060
9	77,371,252,414,645,652,688,308,441
10	77,371,252,454,787,112,897,757,494
11	77,371,252,455,331,339,114,273,267
12	77,371,252,455,336,237,790,445,107
13	77,371,252,455,336,267,065,046,065
14	77,371,252,455,336,267,181,195,263

Figure 5: CDT for parameter set 2

$$
\begin{aligned}
P_{\text{g512}}(x) = (((((((((&\tfrac{8004071744321549}{2^{58}} \cdot x \cdot 2^{-19} \\
&+ \tfrac{6144316755720785}{2^{58}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{3070710202578901}{2^{58}}) \cdot x \cdot 2^{-17} \\
&+ \tfrac{5162723504379443}{2^{58}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{3817631981177055}{2^{58}}) \cdot x \cdot 2^{-17} \\
&+ \tfrac{4934484841355621}{2^{58}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{2733198821433297}{2^{58}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{630792911273043}{2^{58}}) \cdot x \cdot 2^{-13} \\
&+ \tfrac{7453705375596337}{2^{58}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{2064281503328677}{2^{58}}) \cdot x \cdot 2^{-14} \\
&+ \tfrac{6098097214449223}{2^{58}}) \cdot x \cdot 2^{-9} \\
&+ 1
\end{aligned}
$$

$$
\begin{aligned}
P_{\text{g1024}}(x) = ((((((((((&\tfrac{1018039829034577}{2^{60}} \cdot x \cdot 2^{-21} \\
&+ \tfrac{645870593390651}{2^{60}}) \cdot x \cdot 2^{-20} \\
&+ \tfrac{533527473870741}{2^{60}}) \cdot x \cdot 2^{-16} \\
&+ \tfrac{5930642997942607}{2^{60}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{7248725463683367}{2^{60}}) \cdot x \cdot 2^{-24} \\
&+ \tfrac{241976988139137}{2^{60}}) \cdot x \cdot 2^{-17} \\
&+ \tfrac{886151729371391}{2^{60}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{676080313016947}{2^{60}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{825293359306145}{2^{60}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{377789318627941}{2^{60}}) \cdot x \cdot 2^{-16} \\
&+ \tfrac{922337203685445}{2^{60}}) \cdot x \cdot 2^{-6} \\
&+ 1
\end{aligned}
$$

Figure 6: Polynomial approximation for $\exp\big(x/2\sigma^2\big)$ over I_1

A.2 Rejection Sampling

A.2.1 Algorithm

Essentially, Algorithm 6 describes a way of sampling bits from the Bernoulli distribution with parameter $\Psi(\mathbf{z}, \mathbf{v}) = 1/(M \exp(-\|\mathbf{v}\|^2/(2\sigma^2)) \cosh(\langle \mathbf{z}, \mathbf{v}\rangle/\sigma^2))$ where $M = \exp(1/(2\hat{\alpha}^2))$. To do this, Algorithm 6 combines two Bernoulli distributions by observing that the Bernoulli distribution $\mathcal{B}_\Psi$ can be decomposed into the product of $\mathcal{B}_{\Psi_1}$ and $\mathcal{B}_{\Psi_2}$ where $\Psi_1(\mathbf{v}) = 1/(M \exp(-\|\mathbf{v}\|^2/(2\sigma^2)))$, $\Psi_2(\mathbf{z}, \mathbf{v}) = 1/\cosh(\langle \mathbf{z}, \mathbf{v}\rangle/\sigma^2)$.

Algorithm 6 $\mathsf{RejectSample}_{\sigma,\hat{\alpha}}(\mathbf{z}, \mathbf{v})$

Input $\mathbf{z}, \mathbf{v} \in \mathcal{R}^2$ satisfying $\|\mathbf{v}\| \le \sigma/\hat{\alpha}$ and $|\langle \mathbf{z}, \mathbf{v}\rangle| \le \beta_2\sigma/\hat{\alpha}$
Output $b \in \{0, 1\}$

$x \leftarrow ||\mathbf{v}||^2 - \frac{\sigma^2}{\hat{\alpha}^2}$ // $x \in I_2 = [-\frac{\sigma^2}{\hat{\alpha}^2}, 0]$
$y \leftarrow 2\langle \mathbf{z}, \mathbf{v}\rangle$ // $y \in [-2\beta_2\sigma/\hat{\alpha}, 2\beta_2\sigma/\hat{\alpha}]$
$b_1 \leftarrow \mathsf{SampleBernExp}_\sigma^{I_2}(x)$ // $b_1 \dot{\sim} \mathcal{B}_{\exp(x/2\sigma^2))}$
$b_2 \leftarrow \mathsf{SampleBernCosh}_\sigma(y)$ // $b_2 \dot{\sim} \mathcal{B}_{1/\cosh(y/2\sigma^2)}$
return $b_1 b_2$

$\mathsf{SampleBernExp}_\sigma^{I_2}(x)$ implements an approximated version of the Bernoulli distribution $\mathcal{B}_{\exp(x/(2\sigma^2))}$ for $x \in I_2$, based on the following approximations: for $x = x_1\hat{c} + x_0, x_0 \in I_2' = (-\hat{c}, 0]$,

$$
\begin{aligned}
\exp\big(\tfrac{x}{2\sigma^2}\big) &= \exp\Big(\tfrac{x_1 c + x_1(\hat{c}-c) + x_0}{2\sigma^2}\Big) = 2^{x_1} \exp\big(\tfrac{\hat{c}-c}{2\sigma^2}\big)^{x_1} \exp\big(\tfrac{x_0}{2\sigma^2}\big) \\
&\approx 2^{x_1}(1 + \omega x_1) \cdot P_{\text{exp}}^{I_2'}(x_0)
\end{aligned}
$$

where

$$\hat{c} \approx c = 2\sigma^2 \ln(2), \quad \exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{x_1} \approx (1+\omega x_1), \quad \exp\left(\tfrac{x_0}{2\sigma^2}\right) \approx P_{\exp}^{I_2'}(x_0).$$

In particular, the output of $\mathsf{SampleBernExp}_{\sigma}^{I_2}(x)$ is distributed according to the Bernoulli distribution with parameter similar to $2^{x_1}(1+\omega x_1)\cdot P_{\exp}^{I_2'}(x_0)$. Note that $\mathsf{EvaluatePoly}[P_{\exp}^{I_2'}, \vartheta_2](x_0)$ is described in the Algorithm 5 with parameter $P_{\exp}^{I_2'}$ and ϑ_2.

Algorithm 7 $\mathsf{SampleBernExp}_{\sigma}^{I_2}(x)$

Input $x \in I_2 = [-\frac{\sigma^2}{\hat{\alpha}^2}, 0]$
Output $b \in \{0,1\}$

$x_1 \leftarrow \lceil x/\hat{c} \rceil$ // $\hat{c} \approx 2\sigma^2 \ln(2)$
$x_0 \leftarrow x - x_1\hat{c}$ // $x_0 \in I_2' = (-\hat{c}, 0]$
$\hat{p}_0 \leftarrow \mathsf{EvaluatePoly}[P_{\exp}^{I_2'}, \vartheta_2](x_0)$
$\hat{p}_1 \leftarrow 2^{x_1}(1+\omega x_1)$
$\hat{p} \leftarrow \mathsf{Mul}[\vartheta_2](\hat{p}_0, \hat{p}_1)$
$u \leftarrow \mathcal{U}([0,1)_{\vartheta_2})$
if $u \le \hat{p}$ **then return** 1 **else return** 0

$\mathsf{SampleBernCosh}_{\sigma}(y)$ implements an approximated version of the Bernoulli distribution $\mathcal{B}_{1/\cosh(y/2\sigma^2)}$, based on the following approximations:

$$\begin{aligned} 2\cosh\left(\tfrac{y}{2\sigma^2}\right) &= \exp\left(\tfrac{y}{2\sigma^2}\right) + \exp\left(\tfrac{-y}{2\sigma^2}\right) \\ &= 2^{y_1}\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{y_1}\exp\left(\tfrac{y_0}{2\sigma^2}\right) + 2^{-y_1+1}\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{-y_1+1}\exp\left(\tfrac{-y_0-\hat{c}}{2\sigma^2}\right) \\ &\approx 2^{y_1}(1+y_1\omega)P_{\exp}^{I_2'}(y_0) + 2^{-y_1+1}(1+(1-y_1)\omega)P_{\exp}^{I_2'}(-y_0-\hat{c}). \end{aligned}$$

where $y = y_1\hat{c} + y_0$, $y_0 \in I_2'$. In particular, the output of $\mathsf{SampleBernCosh}_{\sigma}(y)$ is distributed according to the Bernoulli distribution with parameter similar to

$$\frac{2}{2^{y_1}(1+y_1\omega)P_{\exp}^{I_2'}(y_0) + 2^{-y_1+1}(1+(1-y_1)\omega)P_{\exp}^{I_2'}(-y_0-\hat{c})}.$$

Note that $u\hat{p}$ is multiplied with exactly and then compared with 1.

Algorithm 8 $\mathsf{SampleBernCosh}_{\sigma}(y)$

Input $y \in [-2\beta_2\sigma/\hat{\alpha}, 2\beta_2\sigma/\hat{\alpha}]$
Output $b \in \{0,1\}$

$y_1 \leftarrow \lceil y/\hat{c} \rceil$
$y_0 \leftarrow y - y_1\hat{c}$
$\hat{p}_0 \leftarrow \mathsf{EvaluatePoly}[P_{\exp}^{I_2'}, \vartheta_2](-y_0-\hat{c})$
$\hat{p}_1 \leftarrow 2^{-y_1+1}(1+\omega(1-y_1))$
$\hat{q}_0 \leftarrow \mathsf{EvaluatePoly}[P_{\exp}^{I_2'}, \vartheta_2](y_0)$
$\hat{q}_1 \leftarrow 2^{y_1}(1+\omega y_1)$
$\hat{p} \leftarrow 2^{-1}(\mathsf{Mul}[\vartheta_2](\hat{p}_0, \hat{p}_1) + \mathsf{Mul}[\vartheta_2](\hat{q}_0, \hat{q}_1))$
$u \leftarrow \mathcal{U}([0,1)_{\vartheta_2})$
if $u\hat{p} \le 1$ **then return** 1 **else return** 0

$\mathsf{Mul}[\vartheta](a,b)$ returns an approximate value of $a \cdot b$ based on the following approximations:

$$a \cdot b \approx \lfloor a' \cdot b' \cdot 2^{\vartheta} \rceil \cdot 2^{-\vartheta + \lfloor \log_2 |a| \rfloor + \lfloor \log_2 |b| \rfloor}$$

where $a' = a \cdot 2^{-\lfloor \log_2 a \rfloor} \in (\frac{1}{2}, 1]$ and $b' = b \cdot 2^{-\lfloor \log_2 b \rfloor} \in (\frac{1}{2}, 1]$. For $a, b \in (0,1)$, the difference between $\mathsf{Mul}[\vartheta](a,b)$ and $a \cdot b$ is bounded by the following inequality:

$$\begin{aligned} |\mathsf{Mul}[\vartheta](a,b) - a \cdot b| &= \left| \lfloor a' \cdot b' \cdot 2^{\vartheta} \rceil \cdot 2^{-\vartheta + \lfloor \log_2 a \rfloor + \lfloor \log_2 b \rfloor} - a \cdot b \right| \\ &= 2^{-\vartheta + \lfloor \log_2 a \rfloor + \lfloor \log_2 b \rfloor} \cdot \left| \lfloor a' \cdot b' \cdot 2^{\vartheta} \rceil - a' \cdot b' \cdot 2^{\vartheta} \right| \\ &\leq 2^{-(\vartheta+1) + \lfloor \log_2 a \rfloor + \lfloor \log_2 b \rfloor} \leq 2^{-(\vartheta+1)}. \end{aligned}$$

Algorithm 9 $\mathsf{Mul}[\vartheta](a,b)$

Input $a, b > 0$
Output c

1: $a' \leftarrow a \cdot 2^{-\lfloor \log_2 a \rfloor}$
2: $b' \leftarrow b \cdot 2^{-\lfloor \log_2 b \rfloor}$
3: $c \leftarrow \lfloor a' \cdot b' \cdot 2^{\vartheta} \rceil \cdot 2^{-\vartheta + \lfloor \log_2 a \rfloor + \lfloor \log_2 b \rfloor}$
4: **return** c

A.2.2 Constraints for parameters

$\delta_{\mathbf{reject}}, c_{\mathbf{r},1}, c_{\mathbf{r},2}$ $\quad \delta_{\text{reject}} = \frac{1}{2^6 \sigma^2 \sqrt{\lambda Q_{\text{reject}}}} < 1$, and

$$c_{\text{r},1} = \frac{1}{2\delta_{\text{reject}}} \left(1 - \sqrt{\frac{1}{4\delta_{\text{reject}}(c_{\text{r},2} - 5/8) + 1}} \right) < 1, \tag{9}$$

$$c_{\text{r},2} = \frac{1}{8\delta_{\text{reject}}} \left(\sqrt{1 + 16\delta_{\text{reject}}} - 1 \right) < 1. \tag{10}$$

are small positive constants used for proper scaling to adjust some inequalities. Note that $c_{\text{r},1}$, $c_{\text{r},2}$ satisfy some inequalities such as $c_{\text{r},1} < c_{\text{r},2} - \frac{5}{8}$,

$$2(1 + 2^{\lfloor \log_2 (1+\omega\zeta) \rfloor}) \left(1 + \frac{2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}} \right)^2 < 5, \tag{11}$$

$$\frac{c_{\text{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}} + \left(\frac{c_{\text{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}} \right)^2 \leq (c_{\text{r},2} - \tfrac{5}{8}) \cdot \delta_{\text{reject}} \tag{12}$$

where $\zeta = \lceil \frac{\beta_2}{\sigma \hat{\alpha} \ln 2} \rceil$.

$\hat{c}$ $\quad \hat{c}$ is an approximate value of $c = 2\sigma^2 \ln 2$, and it is set to satisfy

$$(\hat{c} - c)^2 \leq \frac{8\sigma^4}{e\zeta^2} \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}. \tag{13}$$

ω $\quad \omega$ is a approximate value of $\frac{\hat{c} - c}{2\sigma^2}$ and satisfies

$$\frac{\hat{c} - c}{2\sigma^2} - \omega \leq \frac{1}{e(\zeta + 1)} \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}. \tag{14}$$

$P_{\mathbf{exp}}^{I_2'}$ $P_{\text{exp}}^{I_2'}$ is a polynomial approximation for $\exp\left(-\frac{x}{2\sigma^2}\right)$ over $I_2' = (-\hat{c}, 0]$, which satisfies

$$\left| \frac{P_{\text{exp}}^{I_2'}(x)}{\exp\left(x/2\sigma^2\right)} - 1 \right| \leq c_{\text{r},1} \cdot \delta_{\text{reject}} \tag{15}$$

for all $x \in I_2'$.

ϑ_2 ϑ_2 represents the precision required for evaluating the polynomial $P_{\text{exp}}^{I_2'}$ at $x \in I_2'$. It needs to satisfy

$$2^{\vartheta_2} \geq (2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}})^{-1}. \tag{16}$$

A.2.3 Security

Rényi divergence between the Bernoulli distribution $\mathcal{B}_{1/(M\exp(-||\mathbf{v}||^2/2\sigma^2)\cosh(\langle \mathbf{z},\mathbf{v}\rangle/\sigma^2))}$ and output distribution of $\mathsf{RejectSample}_{\sigma,\hat{\alpha}}(\mathbf{z}, \mathbf{v})$ is demonstrated by showing how closely the approximation polynomial represents the exponential and hyperbolic cosine functions. In Theorem A.10, the Rényi divergence is bounded by demonstrating the bounds of each part.

Theorem A.10. Let $\tilde{\mathcal{B}}_{\Psi(\mathbf{z},\mathbf{v})}$ be a probability distribution of $\mathsf{RejectSample}_{\sigma,\hat{\alpha}}(\mathbf{z}, \mathbf{v})$ in Algorithm 6. If $P_{\text{exp}}^{I_2'}(x)$, $\hat{c}$ and ω satisfy (13), (14), (15), then

$$R_{2\lambda}(\tilde{\mathcal{B}}_{\Psi(\mathbf{z},\mathbf{v})} \,|\, \mathcal{B}_{\Psi(\mathbf{z},\mathbf{v})}) \;\leq\; 1 + \frac{1}{4Q_{\text{reject}}}$$

for all $\mathbf{z}$, $\mathbf{v}$ satisfying $\|\mathbf{v}\| \leq \frac{\sigma}{\hat{\alpha}}$, $|\langle \mathbf{z}, \mathbf{v}\rangle| \leq \frac{\beta_2 \sigma}{\hat{\alpha}}$ where $\Psi(\mathbf{z}, \mathbf{v}) = 1/(M\exp(-\|\mathbf{v}\|^2/(2\sigma^2))\cosh(\langle \mathbf{z}, \mathbf{v}\rangle/\sigma^2))$, $M = \exp(\frac{1}{2\hat{\alpha}^2})$.

Proof. Let us denote the output distributions of $\mathsf{SampleBernExp}_\sigma(x)$ in Algorithm 7 and $\mathsf{SampleBernCosh}_\sigma(y)$ in Algorithm 8 as $\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}$ and $\widetilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}$, respectively. Then, by Lemma A.8, the Rényi divergence is bounded by its relative error.

Letting $A = \frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}$ and $B = \frac{\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)}$, we have, for $x \in [-\sigma^2/\hat{\alpha}^2, 0]$,

$$\begin{aligned}
\left| \frac{\tilde{\mathcal{B}}_{\Psi(\mathbf{z},\mathbf{v})}(1)}{\mathcal{B}_{\Psi(\mathbf{z},\mathbf{v})}(1)} - 1 \right| &= \left| \frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1) \cdot \tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1) \cdot \mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1 \right| = |AB - 1| \\
&\leq |A - 1| + |B - 1| + |(A - 1)(B - 1)| \\
&\leq 2^4 \cdot c_{\text{r},2} \cdot \delta_{\text{reject}} + 2^6 \cdot c_{\text{r},2}^2 \cdot \delta_{\text{reject}}^2 \\
&\leq 2^4 \cdot \delta_{\text{reject}}
\end{aligned}$$

where the second inequality from the last comes from Lemma A.12, Lemma A.14, and the last equality

holds by (10). Similarly, we have, for $x \in [-\sigma^2/\hat{\alpha}^2, 0]$,

$$\begin{aligned}
&\left|\frac{\tilde{\mathcal{B}}_{\Psi(\mathbf{z},\mathbf{v})}(0)}{\mathcal{B}_{\Psi(\mathbf{z},\mathbf{v})}(0)} - 1\right| = \left|\frac{1-\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)\cdot\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{1-\mathcal{B}_{\exp(x/2\sigma^2)}(1)\cdot\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1\right| \\
&= \left|\frac{\mathcal{B}_{\exp(x/2\sigma^2)}(1)\cdot\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)}{1-\mathcal{B}_{\exp(x/2\sigma^2)}(1)\cdot\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)}\right| \cdot \left|\frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)\cdot\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)\cdot\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1\right| \\
&\le \left|\frac{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}{1-\mathcal{B}_{\exp(x/2\sigma^2)}(1)}\right| \cdot |AB - 1| \\
&\le \left|\frac{\exp(-1/2\sigma^2)}{1-\exp(-1/2\sigma^2)}\right| \cdot |AB - 1| \\
&\le \left|\frac{\exp(-1/2\sigma^2)}{1-\exp(-1/2\sigma^2)}\right| \cdot 2^4 \cdot \delta_{\text{reject}} \\
&\le 2^5 \cdot \sigma^2 \cdot \delta_{\text{reject}}
\end{aligned}$$

where the last inequality holds since $\exp(\frac{1}{2\sigma^2}) > 1 + \frac{1}{2\sigma^2}$ and $\left|\frac{\exp(-1/2\sigma^2)}{1-\exp(-1/2\sigma^2)}\right| < 2\sigma^2$.

Thus, the relative error between the two distributions are bounded by $2^5 \cdot \sigma^2 \cdot \delta_{\text{reject}}$, which proves the result by Lemma A.8.

□

Lemma A.11. Let $\widetilde{P}^{I_2'}_{\text{exp}}(x)$ be the output value of Algorithm 5 with parameters $P^{I_2'}_{\text{exp}}$ and ϑ_2. If $P^{I_2'}_{\text{exp}}(x)$, ϑ_2 satisfies (15), (16), then

$$\left|\frac{\widetilde{P}^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)} - 1\right| \le 2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}}$$

for $x \in I_2'$.

Proof. The result is derived from following inequality:

$$\begin{aligned}
\left|\frac{\widetilde{P}^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)} - 1\right| &\le \left|\frac{\widetilde{P}^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)} - \frac{P^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)}\right| + \left|\frac{P^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)} - 1\right| \\
&\le \left|\widetilde{P}^{I_2'}_{\text{exp}}(x) - P^{I_2'}_{\text{exp}}(x)\right| + \left|\frac{P^{I_2'}_{\text{exp}}(x)}{\exp\left(x/(2\sigma^2)\right)} - 1\right| \\
&\le 2 \cdot c_{\text{r},1} \cdot \delta_{\text{reject}},
\end{aligned}$$

where the last inequality comes from the (15), (16) and Lemma A.7.

□

Lemma A.12. Let $\widetilde{\mathcal{B}}_{\exp(x/2\sigma^2)}$ be the output distribution of $\mathsf{SampleBernExp}^{I_2}_{\sigma}(x)$ in Algorithm 7. If $P^{I_2'}_{\text{exp}}(x)$, $\hat{c}$, ϑ_2 and ω satisfy (13), (14), (15), (16), then

$$\left|\frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)} - 1\right| \le 2^3 \cdot c_{\text{r},2} \cdot \delta_{\text{reject}}$$

for all $x \in I_2 = \left[-\frac{\sigma^2}{\hat{\alpha}^2}, 0\right]$.

Proof. Note that

$$\left|\frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}-1\right| \le \underbrace{\left|\frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}-\frac{\mathcal{B}_{\widetilde{\exp}(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}\right|}_{(*)} + \underbrace{\left|\frac{\mathcal{B}_{\widetilde{\exp}(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}-1\right|}_{(\star)},$$

where $\widetilde{\exp}(x/2\sigma^2) = 2^{x_1}(1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0)$. We will prove the result by finding an upper bound of $(\star)$ and an upper bound of $(*)$ respectively, and then by combining them.

Let us define $A = \frac{1+\omega x_1}{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{x_1}}$ and $B = \frac{\widetilde{P}^{I_2'}_{\exp}(x_0)}{\exp\left(\frac{x_0}{2\sigma^2}\right)}$. Then, for $x \in (-\hat{c}, 0]$, we have $x_1 = 0$ and so

$$(\star) = \left|\frac{2^{x_1}(1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0)}{\exp(x/2\sigma^2)}-1\right| = \left|\frac{\widetilde{P}^{I_2'}_{\exp}(x_0)}{\exp(x_0/2\sigma^2)}-1\right| \le 2\cdot c_{\mathrm{r},1}\cdot\delta_{\text{reject}} < 2^3\cdot(c_{\mathrm{r},2}-\tfrac{5}{8})\cdot\delta_{\text{reject}},$$

where the inequality holds by Lemma A.11 and $c_{\mathrm{r},1} < c_{\mathrm{r},2} - \frac{5}{8} < 1$. Moreover, for $x = x_1\hat{c} + x_0$, $x_0 \in (-\hat{c}, 0]$, we have

$$\begin{aligned}
(\star) &\le \left|\frac{2^{x_1}(1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0)}{2^{x_1}\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{x_1}\exp\left(\frac{x_0}{2\sigma^2}\right)}-1\right| \\
&= |AB-1| \\
&\le |A-1| + |B-1| + \|A-1||B-1| \\
&\le 4\cdot c_{\mathrm{r},1}\cdot\delta_{\text{reject}} + 4\cdot c_{\mathrm{r},1}^2\cdot\delta_{\text{reject}}^2 \\
&< 2^3\cdot(c_{\mathrm{r},2}-\tfrac{5}{8})\cdot\delta_{\text{reject}}
\end{aligned}$$

where two inequalities from the last come from Lemma A.11, Lemma A.13, and $c_{\mathrm{r},1} < c_{\mathrm{r},2} - \frac{5}{8} < 1$.

Since $\mathsf{SampleBernExp}^{I_2}_{\sigma}(x)$ computes its value with $\mathsf{Mul}[\vartheta_2](\,)$, we have

$$\begin{aligned}
(*) &= \left|\frac{2^{x_1}\mathsf{Mul}[\vartheta_2](1+\omega x_1,\ \widetilde{P}^{I_2'}_{\exp}(x_0)) - 2^{x_1}(1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0)}{2^{x_1}\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{x_1}\exp\left(\frac{x_0}{2\sigma^2}\right)}\right| \\
&\le e\cdot\left|\mathsf{Mul}[\vartheta_2](1+\omega x_1,\ \widetilde{P}^{I_2'}_{\exp}(x_0)) - (1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0)\right| \\
&\le e\cdot 2^{-(\vartheta_2+1)}
\end{aligned}$$

where the last inequality comes from $(1+\omega x_1)\widetilde{P}^{I_2'}_{\exp}(x_0) < 1$.

Finally, combining these two upper bounds, we have

$$\begin{aligned}
\left|\frac{\tilde{\mathcal{B}}_{\exp(x/2\sigma^2)}(1)}{\mathcal{B}_{\exp(x/2\sigma^2)}(1)}-1\right| &\le e\cdot 2^{-(\vartheta_2+1)} + 2^3\cdot(c_{\mathrm{r},2}-\tfrac{5}{8})\cdot\delta_{\text{reject}} \\
&\le e\cdot c_{\mathrm{r},1}\cdot\delta_{\text{reject}} + 2^3\cdot(c_{\mathrm{r},2}-\tfrac{5}{8})\cdot\delta_{\text{reject}} < 2^3\cdot c_{\mathrm{r},2}\cdot\delta_{\text{reject}}
\end{aligned}$$

where the last inequality comes from $c_{\mathrm{r},1} < 1$.

□

Lemma A.13. If $\hat{c}, \omega$ satisfy (13) and (14), then

$$\left|\frac{1+\omega x}{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^x} - 1\right| \;\leq\; 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}, \tag{17}$$

$$\left|\frac{1+\omega(1-x)}{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{1-x}} - 1\right| \;\leq\; 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}} \tag{18}$$

for all $x \in (-\zeta, 0]$ where $\zeta = \lceil \frac{\beta_2}{\sigma\hat{\alpha}\ln 2} \rceil$.

Proof. Note that $\left|e^t - (1+t)\right| \leq \frac{t^2}{2}$ for $t \in (-1, 0]$ and $\left|e^t - (1+t)\right| \leq \frac{t^2}{2}e$ for $t \in (0, 1)$. Since $x \in (-\zeta, 0]$ and $0 < \frac{\zeta(\hat{c}-c)}{2\sigma^2} < 1$, we have

$$\begin{aligned}\left|\frac{1+\omega x}{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^x} - 1\right| &= \left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^x - (1+\omega x)\right| \cdot \exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{-x} \\ &\leq \underbrace{\left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^x - (1+\omega x)\right| \cdot e}_{(\star)}.\end{aligned}$$

Then

$$\begin{aligned}(\star) &\leq \left(\left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}x\right) - (1+\tfrac{\hat{c}-c}{2\sigma^2}x)\right| + \left|(1+\tfrac{\hat{c}-c}{2\sigma^2}x) - (1+\omega x)\right|\right) \cdot e \\ &\leq \left(\tfrac{1}{2}\left(\tfrac{\zeta(\hat{c}-c)}{2\sigma^2}\right)^2 + \left|(1+\tfrac{\hat{c}-c}{2\sigma^2}x) - (1+\omega x)\right|\right) \cdot e \\ &\leq \tfrac{e}{2}\left(\tfrac{\zeta(\hat{c}-c)}{2\sigma^2}\right)^2 + e\zeta\left|\tfrac{\hat{c}-c}{2\sigma^2} - \omega\right| \\ &\leq 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}\end{aligned}$$

where the last inequality comes from the (13), (14).

On the other hand, since $-1 < \frac{(x-1)(\hat{c}-c)}{2\sigma^2} \leq 0$, we have

$$\begin{aligned}\left|\frac{1+\omega(1-x)}{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{1-x}} - 1\right| &= \left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{1-x} - (1+\omega(1-x))\right| \cdot \exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{x-1} \\ &\leq \underbrace{\left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{1-x} - (1+\omega(1-x))\right|}_{(*)}.\end{aligned}$$

Then

$$\begin{aligned}(*) &\leq \left|\exp\left(\tfrac{\hat{c}-c}{2\sigma^2}(1-x)\right) - (1+\tfrac{\hat{c}-c}{2\sigma^2}(1-x))\right| + \left|(1+\tfrac{\hat{c}-c}{2\sigma^2}(1-x)) - (1+\omega(1-x))\right| \\ &\leq \tfrac{e}{2}\left(\tfrac{\zeta(\hat{c}-c)}{2\sigma^2}\right)^2 + \left|(1+\tfrac{\hat{c}-c}{2\sigma^2}(1-x)) - (1+\omega(1-x))\right| \\ &\leq \tfrac{e}{2}\left(\tfrac{\zeta(\hat{c}-c)}{2\sigma^2}\right)^2 + (\zeta+1)\left|\tfrac{\hat{c}-c}{2\sigma^2} - \omega\right| \\ &\leq 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}\end{aligned}$$

where the last inequality also comes from the (13), (14).

□

Lemma A.14. Let $\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}$ be the output distribution of $\mathsf{SampleBernCosh}_\sigma(y)$ in Algorithm 8. If $P^{I'_2}_{\exp}(x)$, $\hat{c}$, ω, ϑ_2 satisfy (13), (14), (15), (16), then

$$\left|\frac{\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1\right| \le 2^3 \cdot c_{\mathrm{r},2} \cdot \delta_{\mathrm{reject}}$$

for all $y \in \left[-\frac{2\beta_2\sigma}{\hat{\alpha}}, \frac{2\beta_2\sigma}{\hat{\alpha}}\right]$.

Proof. Note that

$$\left|\frac{\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1\right| \le \underbrace{\left|\frac{\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - \frac{\mathcal{B}_{1/\widetilde{\cosh}(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)}\right|}_{(*)} + \underbrace{\left|\frac{\mathcal{B}_{1/\widetilde{\cosh}(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1\right|}_{(\star)}.$$

We will prove the result by finding an upper bound of $(\star)$ and an upper bound of $(*)$ respectively, and then by combining them.

Let us define $\widetilde{\cosh}(y/2\sigma^2) = \frac{2^{y_1}(1+\omega y_1)\widetilde{P}^{I'_2}_{\exp}(y_0) + 2^{1-y_1}(1+\omega(1-y_1))\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})}{2}$ and

$$A = \frac{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{y_1}}{1+\omega y_1}, \quad B = \frac{\exp\left(\frac{y_0}{2\sigma^2}\right)}{\widetilde{P}^{I'_2}_{\exp}(y_0)}, \quad C = \frac{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{1-y_1}}{1+\omega(1-y_1)}, \quad D = \frac{\exp\left(-\frac{y_0+\hat{c}}{2\sigma^2}\right)}{\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})}.$$

Then, for $y = y_1\hat{c} + y_0$, $y_0 \in (-\hat{c}, 0]$, we have

$$\begin{aligned}
(\star) &= \left|\frac{\exp\left(\frac{y}{2\sigma^2}\right) + \exp\left(-\frac{y}{2\sigma^2}\right)}{2^{y_1}(1+\omega y_1)\widetilde{P}^{I'_2}_{\exp}(y_0) + 2^{1-y_1}(1+\omega(1-y_1))\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})} - 1\right| \\
&\le \left|\frac{\exp\left(\frac{y}{2\sigma^2}\right)}{2^{y_1}(1+\omega y_1)\widetilde{P}^{I'_2}_{\exp}(y_0)} - 1\right| + \left|\frac{\exp\left(-\frac{y}{2\sigma^2}\right)}{2^{1-y_1}(1+\omega(1-y_1))\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})} - 1\right| \\
&= \left|\frac{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{y_1}\exp\left(\frac{y_0}{2\sigma^2}\right)}{(1+\omega y_1)\widetilde{P}^{I'_2}_{\exp}(y_0)} - 1\right| + \left|\frac{\exp\left(\frac{\hat{c}-c}{2\sigma^2}\right)^{1-y_1}\exp\left(-\frac{y_0+\hat{c}}{2\sigma^2}\right)}{(1+\omega(1-y_1))\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})} - 1\right| \\
&\le |A-1| + |B-1| + |(A-1)(B-1)| + |C-1| + |D-1| + |(C-1)(D-1)|.
\end{aligned}$$

Since $\left|\frac{1}{A}-1\right|, \left|\frac{1}{B}-1\right|, \left|\frac{1}{C}-1\right|, \left|\frac{1}{D}-1\right|$ are less than or equal to $2c_{\mathrm{r},1}\delta_{\mathrm{reject}}$ by Lemma A.13, Lemma A.11, we have

$$(\star) \le 4\frac{2\cdot c_{\mathrm{r},1}\cdot\delta_{\mathrm{reject}}}{1-2\cdot c_{\mathrm{r},1}\cdot\delta_{\mathrm{reject}}} + 2\left(\frac{2\cdot c_{\mathrm{r},1}\cdot\delta_{\mathrm{reject}}}{1-2\cdot c_{\mathrm{r},1}\cdot\delta_{\mathrm{reject}}}\right)^2 \le 2^3\cdot(c_{\mathrm{r},2} - \tfrac{5}{8})\cdot\delta_{\mathrm{reject}}$$

where two inequalities come from Lemma A.15 and (12).

Let us define

$$\begin{aligned}
&E = \exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{y_1}\exp\left(\tfrac{y_0}{2\sigma^2}\right), \quad F = \exp\left(\tfrac{\hat{c}-c}{2\sigma^2}\right)^{1-y_1}\exp\left(\tfrac{-y_0-\hat{c}}{2\sigma^2}\right), \\
&G = \mathsf{Mul}[\vartheta_2](1+\omega y_1, \widetilde{P}^{I'_2}_{\exp}(y_0)), \quad H = \mathsf{Mul}[\vartheta_2](1+\omega(1-y_1), \widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c})), \\
&I = (1+\omega y_1)\widetilde{P}^{I'_2}_{\exp}(y_0), \qquad J = (1+\omega(1-y_1))\widetilde{P}^{I'_2}_{\exp}(-y_0-\hat{c}).
\end{aligned}$$

Since $\mathsf{SampleBernCosh}_\sigma(y)$ computes its value with $\mathsf{Mul}[\vartheta_2](\,)$, we have the following inequalities. Without loss of generality, we assume $y \geq 0$. Then, for $y \in [0, \hat{c})$, we have $y_1 = 0$ and so

$$\begin{aligned}
(*) &= \left| \frac{(\exp\left(\frac{y_0}{2\sigma^2}\right) + 2F)(\widetilde{P}^{I_2'}_{\exp}(y_0) + 2H - \widetilde{P}^{I_2'}_{\exp}(y_0) - 2J)}{(\widetilde{P}^{I_2'}_{\exp}(y_0)) + 2H) \cdot (\widetilde{P}^{I_2'}_{\exp}(y_0) + 2J)} \right| \\
&\leq \left| \frac{\exp\left(\frac{y_0}{2\sigma^2}\right) + 2F}{\widetilde{P}^{I_2'}_{\exp}(y_0) + 2J} \right| \cdot 2\,|H - J| \\
&\leq \left(\left| \frac{\exp\left(\frac{y_0}{2\sigma^2}\right)}{\widetilde{P}^{I_2'}_{\exp}(y_0)} \right| + \left| \frac{F}{J} \right| \right) \cdot 2\,|H - J| = (|B| + |C \cdot D|) \cdot 2\,|H - J| \\
&\leq 2\big((1 + \tfrac{c_{\mathrm{r},1} \cdot \delta_{\text{reject}}}{1 - c_{\mathrm{r},1} \cdot \delta_{\text{reject}}}) + (1 + \tfrac{2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}})^2\big) 2^{-(\vartheta_2+1)} \\
&\leq 5 \cdot 2^{-(\vartheta_2+1)}
\end{aligned}$$

where last two inequalities come from $J < 1$, Lemma A.11, Lemma A.13 and (11). Moreover, for $y \in [\hat{c}, 2\beta_2\sigma/\hat{\alpha}]$, we have

$$\begin{aligned}
(*) &= \left| \frac{(2^{y_1}E + 2^{1-y_1}F)(2^{y_1}G + 2^{1-y_1}H - 2^{y_1}I - 2^{1-y_1}J)}{(2^{y_1}G + 2^{1-y_1}H) \cdot (2^{y_1}I + 2^{1-y_1}J)} \right| \\
&= \left| \frac{(E + 2^{1-2y_1}F)(G + 2^{1-2y_1}H - I - 2^{1-2y_1}J)}{(G + 2^{1-2y_1}H) \cdot (I + 2^{1-2y_1}J)} \right| \\
&\leq \left| \frac{(E + 2^{1-2y_1}F)(G + 2^{1-2y_1}H - I - 2^{1-2y_1}J)}{I + 2^{1-2y_1}J} \right| \\
&\leq \left(\left| \frac{E}{I + 2^{1-2y_1}J} \right| + \left| \frac{2^{1-2y_1}F}{I + 2^{1-2y_1}J} \right| \right) \left(|G - I| + 2^{1-2y_1}\,|H - J| \right) \\
&\leq \left(\left|\tfrac{E}{I}\right| + \left|\tfrac{F}{J}\right| \right) \left(|G - I| + 2^{1-2y_1}\,|H - J| \right) = (|A \cdot B| + |C \cdot D|) \left(|G - I| + 2^{1-2y_1}\,|H - J| \right) \\
&\leq 2 \left(1 + \tfrac{2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}} \right)^2 \left(2^{\lfloor \log_2(1+\omega y_1) \rfloor} + 2^{1-2y_1} \right) 2^{-(\vartheta_2+1)} \\
&\leq 2 \left(1 + \tfrac{2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}}{1 - 2 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}}} \right)^2 \left(1 + 2^{\lfloor \log_2(1+\omega\zeta) \rfloor} \right) 2^{-(\vartheta_2+1)} \\
&\leq 5 \cdot 2^{-(\vartheta_2+1)}
\end{aligned}$$

where $\zeta = \lceil \frac{\beta_2}{\sigma\hat{\alpha}\ln 2} \rceil$ and last four inequalities come from $J < 1$, Lemma A.11, Lemma A.13, and (11).

Finally, combining these two upper bounds, we have

$$\begin{aligned}
\left| \frac{\tilde{\mathcal{B}}_{1/\cosh(y/2\sigma^2)}(1)}{\mathcal{B}_{1/\cosh(y/2\sigma^2)}(1)} - 1 \right| &\leq 5 \cdot 2^{-(\vartheta_2+1)} + 2^3 \cdot (c_{\mathrm{r},2} - \tfrac{5}{8}) \cdot \delta_{\text{reject}} \\
&\leq 5 \cdot c_{\mathrm{r},1} \cdot \delta_{\text{reject}} + 2^3 \cdot (c_{\mathrm{r},2} - \tfrac{5}{8}) \cdot \delta_{\text{reject}} < 2^3 \cdot c_{\mathrm{r},2} \cdot \delta_{\text{reject}}
\end{aligned}$$

where the last inequality comes from $c_{\mathrm{r},1} < 1$.

□

Lemma A.15. If $|a - 1| \leq \kappa$, then $\left|\frac{1}{a} - 1\right| \leq \frac{\kappa}{1-\kappa}$.

Proof. From the hypothesis, $\left|\frac{1}{a}\right| \leq \frac{1}{1-\kappa}$. Thus, we have $\left|\frac{1}{a} - 1\right| = |a - 1| \cdot \left|\frac{1}{a}\right| \leq \frac{\kappa}{1-\kappa}$.

□

A.2.4 Concrete parameters

The parameters for the constant time rejection sampling are set as follows:
($Q_{\text{sign}} = 2^{64}, Q_{\text{reject}} = M \cdot Q_{\text{sign}}$)

parameter set	1	2
n	512	1024
σ	110	200
$\hat{c}$	$4397245863/2^{18}$	$58145399841/2^{20}$
ω	$58857/2^{52}$	$41553/2^{52}$
ϑ_2	56	59
$P_{\text{exp}}^{I_2'}$	P_{r512}	P_{r1024}

Table 6: Parameters for the constant-time Bernoulli sampler

$$
\begin{aligned}
P_{r512}(x) = (((((((((\tfrac{65907013597553883}{2^{56}} \cdot x \cdot 2^{-37} \\
&+ \tfrac{50325055043457447}{2^{56}}) \cdot x \cdot 2^{-38} \\
&+ \tfrac{13404400374123511}{2^{56}}) \cdot x \cdot 2^{-34} \\
&+ \tfrac{49757646361832495}{2^{56}}) \cdot x \cdot 2^{-41} \\
&+ \tfrac{1292424452228331}{2^{56}}) \cdot x \cdot 2^{-34} \\
&+ \tfrac{3818042287379289}{2^{56}}) \cdot x \cdot 2^{-34} \\
&+ \tfrac{9869033955450437}{2^{56}}) \cdot x \cdot 2^{-35} \\
&+ \tfrac{109327985819144 61}{2^{56}}) \cdot x \cdot 2^{-39} \\
&+ \tfrac{630792917896649}{2^{56}}) \cdot x \cdot 2^{-33} \\
&+ \tfrac{1863426344390227}{2^{56}}) \cdot x \cdot 2^{-32} \\
&+ \tfrac{8257126013350391}{2^{56}}) \cdot x \cdot 2^{-33} \\
&+ \tfrac{12196194428898877}{2^{56}}) \cdot x \cdot 2^{-30} \\
&+ 1
\end{aligned}
$$

$$
\begin{aligned}
P_{r1024}(x) = ((((((((((\tfrac{166491802315343951}{2^{59}} \cdot x \cdot 2^{-40} \\
&+ \tfrac{206051247573429639}{2^{59}}) \cdot x \cdot 2^{-43} \\
&+ \tfrac{22563387418474339}{2^{59}}) \cdot x \cdot 2^{-35} \\
&+ \tfrac{553337835353119253}{2^{59}}) \cdot x \cdot 2^{-40} \\
&+ \tfrac{380075599106701163}{2^{59}}) \cdot x \cdot 2^{-39} \\
&+ \tfrac{463968372140871685}{2^{59}}) \cdot x \cdot 2^{-41} \\
&+ \tfrac{123893025271994865}{2^{59}}) \cdot x \cdot 2^{-42} \\
&+ \tfrac{14178431944258309}{2^{59}}) \cdot x \cdot 2^{-35} \\
&+ \tfrac{173076561948552247}{2^{59}}) \cdot x \cdot 2^{-37} \\
&+ \tfrac{422550200075985057}{2^{59}}) \cdot x \cdot 2^{-39} \\
&+ \tfrac{193428131138340275}{2^{59}}) \cdot x \cdot 2^{-38} \\
&+ \tfrac{118059162071741129}{2^{59}}) \cdot x \cdot 2^{-34} \\
&+ \tfrac{576460752303423487}{2^{59}}
\end{aligned}
$$

Figure 7: Polynomial approximation for $\exp(x/2\sigma^2)$ over $I_2' = (-\hat{c}, 0]$