More NTRU+Sign Signatures from Cyclotomic Trinomials[§]

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

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

April 4, 2025

Abstract

Recently, NTRU+Sign was proposed as a new compact signature scheme, following 'Fiat-Shamir with Aborts' (FSwA) framework. Its compactness is mainly based on their novel NTRU-based key structure that fits well with bimodal distributions in the FSwA framework. However, despite its compactness, NTRU+Sign fails to provide a diverse set of parameters that can meet some desired security levels. This limitation stems from its reliance on a ring $\mathbb{Z}_q[x]/\langle x^n+1\rangle$, where n is restricted to powers of two, limiting the flexibility in selecting appropriate security levels. To overcome this limitation, we propose a revised version of NTRU+Sign by adopting a ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2}+1\rangle$ from cyclotomic trinomials, where $n = 2^i3^j$ for some positive integers i and j. Our parameterization offers three distinct security levels: approximately 120, 190, and 260 bits, while preserving the compactness in $\mathbb{Z}_q[x]/\langle x^n+1\rangle$. We implement these re-parameterized NTRU+Sign schemes, showing that the performance of NTRU+Sign from cyclotomic trinomials is still comparable to previous lattice-based signature schemes such as Dilithium and HAETAE.

1 Introduction

Recently, Woo *et al.* [27] proposed a new NTRU-based signature scheme called NTRU+Sign, which achieves the most compact signature sizes at similar security levels. Compared to its predecessor BLISS [11], the compactness of NTRU+Sign comes from two distinct techniques. First, NTRU+Sign is based on their novel NTRU-based key structure that fits well with bimodal distributions when using 'Fiat-Shamir with Aborts (FSwA)' paradigm [20, 21]. Given two short polynomials $\boldsymbol{f}$ and $\boldsymbol{g}$ in a ring $\mathbb{Z}_q[x]/\langle x^n+1\rangle$, BLISS sets a signing key $\boldsymbol{s}$ as $\boldsymbol{s} = (\boldsymbol{g}, 2\boldsymbol{f}+1)$, whereas NTRU+Sign sets it as $\boldsymbol{s} = (\boldsymbol{g}, -\boldsymbol{f})$. This distinction significantly reduces the ℓ_2 norm of $\boldsymbol{sc}$ (that is, $||\boldsymbol{sc}||$), where $\boldsymbol{c}$ is a polynomial whose coefficients consist of 0 or 1 with a fixed Hamming weight. Second, [27] also uses the canonical embedding into $\mathbb{C}^n$ [8] to estimate an upper bound on $\|\boldsymbol{sc}\|$ in advance, providing a tighter bound than the one obtained by the Gram matrix method used in BLISS. These two techniques enable NTRU+Sign to achieve a more compact size of a signature than previous schemes such as Dilithium [12], HAETAE [8], and (re-parametrized) BLISS.

[§]This work was supported by Korea Research Institute for defense Technology planning and advancement(KRIT) grant funded by the Korea government(Defense Acquisition Program Administration) (KRIT-CT-24- 001, Defense Space Security Research Lab, 2025).

Table 1: Comparision to previous lattice-based signature schemes

Algorithm	Classical Security	|sig| (bytes)	|vk| (bytes)	|sig|+|vk| (bytes)	Sampling Distribution
Falcon-512[1]	120	897	666	1,563	Gaussian
Dilithium-2[2]	123	2,420	1,312	3,732	Hypercube
Dilithium-G-2[3]	118	1,921	800	2,721	Gaussian
HAETAE-120[4]	119	1,474	992	2,466	Hyperball
G+G-120[5]	121	1,677	1,472	3,149	Convolved Gaussian
Patronus-120[7]	120	2,070	832	2,902	Polytope
NTRU+Sign-648	118	1,009	1,053	2,062	Gaussian
Dilithium-3[2]	182	3,293	1,952	5,245	Hypercube
Dilithium-G-3[3]	183	2,462	1,184	3,646	Gaussian
HAETAE-180[4]	180	2,349	1,472	3,821	Hyperball
G+G-180[5]	178	2,143	1,952	4,095	Convolved Gaussian
NTRU-G+G-180[6]	178	1,769	2,080	3,849	Convolved Gaussian
Patronus-180[7]	182	2,575	1,152	3,727	Polytope
NTRU+Sign-1024[8]	211	1,511	1,664	3,215	Gaussian
NTRU+Sign-972	193	1,557	1,701	3,258	Gaussian
Falcon-1024[1]	273	1,793	1,280	3,073	Gaussian
Dilithium-5[2]	252	4,595	2,592	7,187	Hypercube
Dilithium-G-5[3]	277	3,553	1,760	5,313	Gaussian
HAETAE-260[4]	256	2,948	2,080	5,028	Hyperball
G+G-260[5]	260	2,804	2,336	5,140	Convolved Gaussian
Patronus-260[7]	262	3,721	1,632	5,353	Polytope
NTRU+Sign-1296	264	2,020	2,268	4,288	Gaussian

[1] Hash-and-Sign-based signature [16]
[2] FSwA-based signature with uniform hypercube distribution [12]
[3] Gaussian version of Dilithium [9]
[4] FSwA-based signature with bimodal hyperball distribution [8]
[5] Signature without rejection sampling based on M-LWE [10]
[6] G+G based on NTRU [10]
[7] FSwA-based signature with uniform polytope distribution [3]
[8] FSwA-based signature with bimodal Gaussian distribution [27]

However, despite its compactness, NTRU+Sign supports only a limited set of parameters due to its underlying ring $\mathbb{Z}_q[x]/\langle x^n+1\rangle$. Since n is selected as a power-of-two, the choice of n that meets some required security levels is very limited. In fact, setting $n = 1024$ is their only practical case, where the security parameter λ of NTRU+Sign-1024 is estimated to be $\lambda = 211$ under the classical Core-SVP (Shortest Vector Problem) methodology [1]. To address this limitation, we propose transitioning to the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2}+1\rangle$, where $n = 2^i 3^j$ for some positive integers i and j. As already shown in the case of key encapsulation mechanism (KEM) [24, 19], the rings from cyclotomic trinomials offer greater flexibility than $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ in achieving a desired level of security. Also, the Number Theoretic Transform (NTT) operation required for

polynomial multiplications is essentially as fast as in $\mathbb{Z}_q[x]/\langle x^n+1\rangle$, resulting in negligible performance degradation. Using these facts, we suggest three new parameter sets for compact NTRU+Sign signatures, denoted as NTRU+Sign-$\{648, 972, 1296\}$, targeting security levels of approximately $\lambda = \{120, 190, 260\}$, respectively.

To parameterize each of NTRU+Sign-$\{648, 972, 1296\}$, the first thing that must be done is to set the upper bound on $||\boldsymbol{sc}||$ as tightly as possible, which is critical to achieving compact signature sizes. To achieve this, we borrow the ideas from BLISS and HAETAE; BLISS uses the Gram matrix method, while HAETAE utilizes the canonical embedding into $\mathbb{C}^n$ based on the Fast Fourier Transform (FFT). Since both methods were originally designed for $\mathbb{Z}_q[x]/\langle x^n+1\rangle$, it is necessary to revisit these approaches to operate on $\mathbb{Z}_q[x]/\langle x^n - x^{n/2}+1\rangle$. We analyze both methods over $\mathbb{Z}_q[x]/\langle x^n - x^{n/2}+1\rangle$ and compare the resulting upper bounds on $\|\boldsymbol{sc}\|$, when parameterizing NTRU+Sign-$\{648, 972, 1296\}$ respectively. Importantly, our analysis shows that the Gram matrix method provides a tighter (i.e., smaller) upper bound on $\|\boldsymbol{sc}\|$ than the FFT method (see Table 2). Accordingly, we choose the Gram matrix method to upper bound $\|\boldsymbol{sc}\|$, which results in slower key generation for NTRU+Sign-$\{648, 972, 1296\}$ compared to Dilithium and HAETAE. Nevertheless, NTRU+Sign-$\{648, 972, 1296\}$ give more compact size of signatures.

Table 1 presents a comparison between previous lattice-based signature schemes [12, 9, 8, 10, 3, 16] and NTRU+Sign-$\{648, 972, 1296\}$. In terms of the combined size of a signature and verification key, Falcon [16] (based on 'Hash-and-Sign' framework [17]) achieves the shortest combined size among all compared lattice-based signature schemes. However, Falcon requires relatively complex and complicated implementations [26] and also has limitations on parameter diversity due to its choice of a ring $\mathbb{Z}_q[x]/\langle x^n+1\rangle$. Compared to other FSwA-based signature schemes [12, 9, 8, 10, 3], NTRU+Sign-$\{648, 972, 1296\}$ achieves the most compact size at the same security levels. For example, at the 120-bit security level, the combined size of NTRU+Sign-648 is about 40% smaller than Dilithium-2, and about 20% smaller than HAETAE-120. With appropriate parameterization, we implement NTRU+Sign-$\{648, 972, 1296\}$ and compare their performance against Dilithium and HAETAE (see Table 5). Our implementation shows that NTRU+Sign-$\{648, 972, 1296\}$, which require no floating-point operations, provide competitive signing performance and faster verification.

2 Preliminaries

2.1 Notation

Throughout this paper, we let $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denote the ring of integers modulo q. Define $\mathcal{R} = \mathbb{Z}[x]/\langle x^n - x^{n/2}+1\rangle$ and $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2}+1\rangle$, where q is a prime and n is of the form $2^i \cdot 3^j$ for positive integers i and j. Let $\mathcal{R}_{n,\tau} \subset \mathcal{R}_q$ denote the set of polynomials in which exactly τ coefficients are 1 and the remaining coefficients are 0. An element of the ring is written in bold lowercase letters (e.g., $\mathbf{a}$).
For positive integer a and q, $a \bmod q$ is equal to the unique integer a' in range $[-q/2, q/2)$ satisfying $a' \equiv a \bmod q$. Let $[\boldsymbol{a}]_d$ denote the bit-truncation of the polynomial $\boldsymbol{a} = \sum_{i=0}^{n-1} a_i x^i$, where each coefficient is truncated by removing its lowest d-bit. Formally, $[\boldsymbol{a}]_d = \sum_{i=0}^{n-1} [a_i]_d x^i$ where $[a_i]_d = (a_i - [a_i \bmod 2^d])/2^d$. Let $\{0,1\}^*$ denote the set of all binary strings of arbitrary length. The Central Binomial Distribution(CBD) with k-bit, denoted ψ_k, is defined as $\psi_k = \sum_{i=1}^{k}(b_i - b_i')$ where $b_i, b_i' \in \{0,1\}$ are sampled uniformly at random. For distribution χ, let $x \leftarrow \chi$ denote that the x is chosen according to the distribution χ. If S is a set, $x \leftarrow S$ denotes that x is chosen uniformly at random from S. For a ring element $\boldsymbol{a}$, let $\|\boldsymbol{a}\|_1$ denote ℓ_1 norm that is the sum of the absolute value of all coefficients. The ℓ_2 norm and ℓ_∞ norm are defined as $\|\boldsymbol{a}\| = \sqrt{\sum_i |a_i|^2}$ and $\|\boldsymbol{a}\|_\infty = \max_i\{|a_i|\}$.

2.2 Definition

Definition 2.1 (Digital Signature)**.** A digital signature scheme $\mathbf{SIG}$ consists of three algorithms ($\mathbf{KeyGen}$, $\mathbf{Sign}$, $\mathbf{Verify}$) defined as follows:

- $\mathbf{KeyGen}(1^\lambda) \to (vk, sk)$: Given a security parameter 1^λ, the key generation algorithm outputs a verification key vk and a signing key sk.
- $\mathbf{Sign}(sk, \mu) \to \sigma$: Given a signing key sk and a message μ, the signature generation algorithm outputs a signature σ.
- $\mathbf{Verify}(vk, \sigma, \mu) \to b \in \{0, 1\}$: Given a verification key vk, a signature σ and a message μ, the verification algorithm, which is a deterministic algorithm, outputs a bit $b \in \{0, 1\}$ where $b = 1$ indicates acceptance and $b = 0$ indicates rejection.

The signature scheme is said to be $(1 - \gamma)$-correct for some function $\gamma(\lambda) > 0$, if for all messages μ, and all key pairs $(vk, sk) \leftarrow \mathbf{KeyGen}(1^\lambda)$, it holds that:

$$\mathsf{Pr}[\mathbf{Verify}(vk, \mathbf{Sign}(sk, \mu), \mu) = 1] \leq 1 - \gamma(\lambda).$$

We define the security model of digital signature, existential unforgeability against chosen message attacks (UF-CMA) and existential unforgeability against no message attacks (UF-NMA) as below.

Definition 2.2 (UF-CMA)**.** Let $\delta \geq 0$. A signature scheme $\mathbf{SIG} = (\mathbf{KeyGen}, \mathbf{Sign}, \mathbf{Verify})$ is said to be $\mathsf{UF\text{-}CMA}$ secure in the Random Oracle Model if, for any polynomial-time adversary $\mathcal{A}$ with access to both a signing oracle and a random oracle H, the following holds:

$$\Pr_{(vk,sk)\leftarrow\mathbf{KeyGen}(1^\lambda)}\left[\mathbf{Verify}(vk, \sigma^*, \mu^*) = 1 | (\mu^*, \sigma^*) \leftarrow \mathcal{A}^{H,\mathbf{Sign}}(vk)\right] \leq \delta,$$

where $(vk, sk) \leftarrow \mathbf{KeyGen}(1^\lambda)$, and μ^* was not previously queried to the signing oracle by $\mathcal{A}$. The probability of forging a signature δ is called the advantage of $\mathcal{A}$ and denoted by $\mathsf{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathbf{SIG}}(\mathcal{A})$.

Additionally, we can define $\mathsf{UF\text{-}NMA}$ the same as $\mathsf{UF\text{-}CMA}$, except the adversary $\mathcal{A}$ cannot access to the signing oracle.

Definition 2.3 (Continuous Gaussian Distribution)**.** The continuous Gaussian distribution over $\mathbb{R}^m$ centered at $\boldsymbol{v} \in \mathbb{R}^m$ with standard deviation σ is defined by:

$$\rho^m_{\boldsymbol{v},\sigma}(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{v}\|_2^2}{2\sigma^2}\right).$$

We denote by $\rho^m_\sigma(\boldsymbol{x}) = \rho^m_{\boldsymbol{v},\sigma}(\boldsymbol{x})$ when $\boldsymbol{v} = \boldsymbol{0}$.

Definition 2.4 (Discrete Gaussian Distribution)**.** The discrete Gaussian distribution over $\mathbb{Z}^m$ centered at $\boldsymbol{v} \in \mathbb{Z}^m$ with standard deviation σ is defined by:

$$D^m_{\boldsymbol{v},\sigma}(\boldsymbol{x}) = \frac{\rho^m_{\boldsymbol{v},\sigma}(\boldsymbol{x})}{\rho^m_\sigma(\mathbb{Z}^m)},$$

where $\rho^m_\sigma(\mathbb{Z}^m) = \sum_{\boldsymbol{z}\in\mathbb{Z}^m} \rho^m_\sigma(\boldsymbol{z})$. Additionally, we denote $D^m_\sigma(\boldsymbol{x}) = D^m_{\boldsymbol{v},\sigma}(\boldsymbol{x})$ when $\boldsymbol{v} = \boldsymbol{0}$.

2.3 Lattice Hardness Assumptions

Definition 2.5 (Decisional-$\mathsf{NTRU}_{n,q,\chi}$ [27])**.** Let q be an odd prime modulus, n the dimension of the ring $\mathcal{R}_q$, and χ a distribution over $\mathcal{R}_q$. The advantage of an adversary $\mathcal{A}$ solving the Decisional-$\mathsf{NTRU}_{n,q,\chi}$ problem is

$$\mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\chi}(\mathcal{A}) = \Big|\mathsf{Pr}\Big[b = 1|\ \boldsymbol{u} \leftarrow R_q; b \leftarrow \mathcal{A}(\boldsymbol{u})\Big] - \mathsf{Pr}\Big[b = 1|\ \boldsymbol{f}, \boldsymbol{g} \leftarrow \chi; b \leftarrow \mathcal{A}((\boldsymbol{f} + \hat{q})/\boldsymbol{g})\Big]\Big|,$$

where $\hat{q}$ is a multiplicative inverse of 2 in $\mathbb{Z}_q$.

Definition 2.6 (Search-$\mathsf{RSIS}_{n,q,\beta}$ [27])**.** Let q be an odd prime modulus, n the dimension of the ring $\mathcal{R}_q$, and β a positive real number. The advantage of an adversary $\mathcal{A}$ solving the Search-$\mathsf{RSIS}_{n,q,\beta}$ problem is

$$\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,\beta}(\mathcal{A}) = \mathsf{Pr}\Big[\ 0 < \|\boldsymbol{y}\|_2 \leq \beta\ \wedge\ [\boldsymbol{a}|\mathbf{I}_n] \cdot \boldsymbol{y} = 0\ \bmod q \Big| \boldsymbol{a} \leftarrow \mathcal{R}_q;\ \boldsymbol{y} \leftarrow \mathcal{A}(\boldsymbol{a})\ \Big],$$

where $\boldsymbol{y} \in \mathcal{R}_q^2$, and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix.

Definition 2.7 (BimodalSelfTargetRSIS$_{n,q,\beta,H}$ [27])**.** Let $\mathrm{H} : \{0,1\}^* \times \mathcal{M} \rightarrow \mathcal{R}_{n,\tau}$ be a cryptographic hash function, where $\mathcal{M} \subseteq \{0,1\}^*$ is a message space and $\hat{q}$ be a multiplicative inverse of 2 in $\mathcal{R}_q$. Let q be an odd prime modulus, n the dimension of the ring $\mathcal{R}_q$, and β a positive real number. The advantage of an adversary $\mathcal{A}$ solving the BimodalSelfTargetRSIS$_{n,q,\beta,H}$ problem is defined as:

$$\mathsf{Adv}^{\mathsf{BimodalSelfTargetRSIS}}_{n,q,\beta,H}(\mathcal{A}) =$$
$$\mathsf{Pr}\Big[\ \begin{array}{c} 0 \leq \|\mathbf{Y}\|_2 \leq \beta\ \wedge \|\boldsymbol{Y}\|_\infty \leq (q-2)/4\ \wedge \\ H([\boldsymbol{a}|\boldsymbol{I}] \cdot \boldsymbol{Y} + \hat{q}\boldsymbol{c}, \mu) = \boldsymbol{c} \end{array} \Big|\ \boldsymbol{a} \leftarrow \mathcal{R}_q; \left(\boldsymbol{Y} := \begin{bmatrix}\boldsymbol{y}_1 \\ \boldsymbol{y}_2\end{bmatrix}, \boldsymbol{c}, \mu\right) \leftarrow \mathcal{A}^{|H\rangle}(\boldsymbol{a})\ \Big]$$

where $\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{c} \in R_q$.

Theorem 2.8 (Reduction from RSIS to BimodalSelfTargetRSIS [27])**.** For positive odd modulus q, $\mathrm{H} : \{0,1\}^* \times \mathcal{M} \rightarrow \mathcal{R}_{n,\tau}$ be a cryptographic hash function modeled as random oracle, and there exist an adversary $\mathcal{B}$ solving $\mathrm{RSIS}_{n,q,4\beta+2\sqrt{\tau}}$ with negligible advantage. Then the advantage of an adversary $\mathcal{A}$ solving $\mathrm{BimodalSelfTargetRSIS}_{n,q,\beta,H}$ is:

$$\mathsf{Adv}^{\mathsf{BiomodalSelfTargetRSIS}}_{H,n,q,\beta}(\mathcal{A}) \approx \sqrt{\mathsf{Adv}^{\mathsf{RSIS}}_{n,q,4\beta+2\sqrt{\tau}}(\mathcal{B})/Q_h},$$

where Q_h is the number of classical queries to H.

2.4 Rejection Sampling

Lemma 2.9 (Rejection Sampling [21])**.** Let V be an arbitrary set, and let $h : V \rightarrow \mathbb{R}$ and $f : \mathbb{Z}^m \rightarrow \mathbb{R}$ be probability distributions. If $g_v : \mathbb{Z}^m \rightarrow \mathbb{R}$ 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

$$\forall v \in V, \Pr[M \cdot g_v(\boldsymbol{z}) \geq f(\boldsymbol{z}) | \boldsymbol{z} \leftarrow f] \geq 1 - \epsilon,$$

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

1. $v \leftarrow h, \boldsymbol{z} \leftarrow g_v$, output $(\boldsymbol{z}, v)$ with probability $\mathsf{min}\big(\frac{f(z)}{M \cdot g_v(\boldsymbol{z})}, 1\big)$.
2. $v \leftarrow h, \boldsymbol{z} \leftarrow f$, output $(\boldsymbol{z}, v)$ with probability $1/M$.

And the probability that those two algorithms output $\boldsymbol{z}$ is identically $f(\boldsymbol{z}) \cdot (1 - \epsilon)/M$.

3 NTRU+Sign Signature Schemes

3.1 Algorithms

We present three algorithms of NTRU+Sign consisting of $\mathbf{KeyGen}$, $\mathbf{Sign}$, and $\mathbf{Verify}$. The algorithm description is the same as [27], except with two underlined notations: one is a function $\mathcal{N}_\tau(\boldsymbol{s})$ computing upper bound on $\|\boldsymbol{sc}\|$ in the $\mathbf{KeyGen}$ algorithm, and the other is a more generalized form of the modulus $p = (q + q_0)/2^d$ for $(-q_0) \equiv q \bmod 2^d$, instead of $p = (q-1)/2^d$ in [27].

Algorithm 1 KeyGen

Input: 1^λ
Output: verification key vk, signing key sk

$\boldsymbol{f}, \boldsymbol{g} \leftarrow \psi_1^n$ ▷ CBD sampling
if $\boldsymbol{g}$ is not invertible in R_q, **then** restart
$\boldsymbol{s} := (\boldsymbol{g}, -\boldsymbol{f})$
if $(B_{\boldsymbol{sc}})^2 < \mathcal{N}_\tau(\boldsymbol{s})$, then restart
$\boldsymbol{a} := (\boldsymbol{f} + \hat{q})/\boldsymbol{g} \bmod q$ ▷ $\hat{q} = 2^{-1} \bmod q$
$vk := \boldsymbol{a}$
$sk := \boldsymbol{s}$
return (vk, sk)

Algorithm 2 Sign

Input: signing key $sk = \boldsymbol{s}$, message μ
Output: signature $\sigma = (\boldsymbol{z}_1, \boldsymbol{h}, \boldsymbol{c})$ on message μ

$\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2) \leftarrow \mathcal{D}_\sigma^n \times \mathcal{D}_\sigma^n$ ▷ Discrete Gaussian sampling
$\boldsymbol{u} := \boldsymbol{a}\boldsymbol{y}_1 + \boldsymbol{y}_2 \bmod q$
$\boldsymbol{c} := H([\boldsymbol{u}]_d \bmod p, \mu) \in R_{n,\tau}$ ▷ $p = (q + q_0)/2^d$
Choose a random bit $b \leftarrow \{0, 1\}$
$\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2) \leftarrow \boldsymbol{y} + (-1)^b \boldsymbol{sc}$ ▷ $\boldsymbol{z}_i = \boldsymbol{y}_i + (-1)^b \boldsymbol{s}_i \boldsymbol{c}$
Continue with probability $1/(M\exp(-\frac{\|\boldsymbol{sc}\|^2}{2\sigma^2})\cosh(\frac{\langle \boldsymbol{z}, \boldsymbol{sc}\rangle}{\sigma^2}))$ ▷ Rejection sampling
  otherwise restart
if $[\boldsymbol{u}]_d \neq [\boldsymbol{u} + (-1)^b \boldsymbol{c}]_d$, **then** restart ▷ Equality check
if $\|(\boldsymbol{z}_1, 2^d \boldsymbol{h})\| > B_2$, **then** restart
if $\|(\boldsymbol{z}_1, 2^d \boldsymbol{h})\|_\infty > B_\infty$, **then** restart
return $(\boldsymbol{z}_1, \boldsymbol{h}, \boldsymbol{c})$

Algorithm 3 Verification

Input: verification key vk, signature $(\boldsymbol{z}_1, \boldsymbol{h}, \boldsymbol{c})$, message μ
Output: Accept or Reject the signature

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

Rejection in the KeyGen and Sign algorithms. There are two rejection conditions in the **KeyGen** algorithm. The first one relates to the invertibility of $\boldsymbol{g}$, which is an unavoidable step due to the reliance on the NTRU problem. The second one is to select $\boldsymbol{s}$ such that $\mathcal{N}_\tau(\boldsymbol{s})$ does not exceed $(B_{sc})^2$. Since we set B_{sc} quite high to ensure about a 90% acceptance rate, the second rejection rarely occurs. We will discuss the function $\mathcal{N}_\tau(\cdot)$ and the selection of B_{sc} in Section 4 and 5.1.

In the **Sign** algorithm, a rejection sampling is done at line 6, with the probability of passing this step being $\frac{1}{M}$ according to Lemma 2.9. Also, an equality check is done at line 8, which can pass with probability $\frac{1}{M_{eq}} = \left((2^d-1)/2^d\right)^\tau$. Lastly, the **Sign** algorithm checks whether the Euclidean norm and infinity norm of $(\boldsymbol{z}_1, 2^d\boldsymbol{h})$ exceed the thresholds B_2 and B_∞ at line 9 and 10, respectively. These threshold values are set sufficiently high so that the rejection at these steps occurs with negligible probability. Therefore, the total expected number of repetitions in the **Sign** algorithm, denoted by M_{total}, is given by $M_{total} = M \times M_{eq}$. We discuss the selection of M and M_{eq} in Section 5.2.

3.2 Security

Let $\mathsf{FS}(\mathsf{Ident}, H)$ be the Fiat-Shamir transform [15] that converts an identification scheme Ident into a signature scheme using a hash function H. As usual, NTRU+Sign is derived from $\mathsf{FS}(\mathsf{NTRU+Ident}, H)$, where NTRU+Ident is the underlying identification scheme corresponding to NTRU+Sign. We refer to [27] for the details on NTRU+Ident and [4] for its relevant paHVZK (Perfect Accepting Honest-verifier Zero-knowledge) property. Also, the commitment min-entropy α is given by $\log_2\left((2^d/(\sqrt{2\pi}\sigma-1))^n\right)$ [27], which holds under the assumption that the d low bits of a commitment $\boldsymbol{u}$ are uniformly distributed.

Theorem 3.1 (Reduction from UF-NMA to UF-CMA [27, 4])**.** Let $\epsilon_{zk}, \alpha \geq 0$ and H is a hash function modeled as a random oracle. Assume that NTRU+Ident is a paHVZK public-coin identification protocol with aborting probability β and the commitment w has min-entropy α. If there exist an adversary $\mathcal{A}$ against UF-CMA security of NTRU+Sign $= \mathsf{FS}(\mathsf{NTRU+Ident}, H)$ with at most Q_H queries to the random oracle H and Q_S classical queries to the signing oracle, there exists an adversary $\mathcal{B}$ against UF-NMA security of NTRU+Sign such that

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

Theorem 3.2 (Reduction from NTRU and BimodalSelfTargetRSIS to UF-NMA [27])**.** Let H and H' are hash function modeled as a random oracle such that $H(\boldsymbol{x} \bmod p, \mu) = H'((\boldsymbol{x} \bmod p) \cdot 2^d \bmod q, \mu)$. Assume that there exists an adversary $\mathcal{A}$ against UF-NMA security of NTRU+Sign with at most Q_H queries to the random oracle H. Then there exist two adversaries $\mathcal{B}$ against NTRU problem and $\mathcal{C}$ against BimodalSelfTargetRSIS such that

$$\mathsf{Adv}^{\mathsf{UF\text{-}NMA}}_{\mathsf{NTRU+Sign}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\psi_1}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{BimodalSelfTargetRSIS}}_{n,q,B_2+(2^d+q_0)\sqrt{n},H'}(\mathcal{C}).$$

Notice that the only difference with $p = (q+q_0)/2^d$ is that the BimodalSelfTargetRSIS bound changes from $B_2 + (2^d+1)\sqrt{n}$ to $B_2 + (2^d+q_0)\sqrt{n}$. Other than, the remaining proof of Theorem 3.2 is exactly the same as [27].

4 Bound on $\|\boldsymbol{sc}\|$ in $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$

For $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2) \in R_q^2$ and $\boldsymbol{c} \in R_{n,\tau}$, we provide two approaches to bounding $\|\boldsymbol{sc}\|$ in the ring $R_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. The patterns in polynomial multiplication vary depending on the underlying ring, affecting the bound on $\|\boldsymbol{sc}\|$. In the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$, unlike in $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$, the coefficient of the term $c_n x^n$ (if it exists) is added to the coefficients of both $x^{n/2}$ and the constant term. This follows from the relation $x^n = x^{n/2} - 1$ in the ring, which introduces additional complexity in multiplication patterns. Consequently, deriving a tight bound on $\|\boldsymbol{sc}\|$ becomes challenging. To resolve this problem, we adapt the techniques from prior methods based on FFT [8] and Gram matrix [11], each of which was originally suggested for $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$. We extend their methods to the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$ to obtain a suitable bound on $\|\boldsymbol{sc}\|$.

4.1 Bounding $\|\boldsymbol{sc}\|$ with Fast Fourier Transform

There are three common ways to represent ring elements. For $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle \Phi_m(x)\rangle$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial of degree $n = \phi(m)$, the most common representation of a ring element $\boldsymbol{a} \in \mathcal{R}_q$ is $\boldsymbol{a} = \sum_{i=0}^{n-1} a_i x^i$ as a polynomial. For an element $\boldsymbol{a} \in \mathcal{R}_q$, the coefficient embedding is written as $\overrightarrow{\boldsymbol{a}} = \{a_0, a_1, \ldots, a_{n-1}\} \in \mathbb{Z}^n$ as a vector. The canonical embedding of $\boldsymbol{a} \in \mathcal{R}_q$ is defined as $\mathcal{C}(\boldsymbol{a}) = \mathbf{U_m} \cdot \overrightarrow{\boldsymbol{a}}^T = (\boldsymbol{a}(\omega_1), \boldsymbol{a}(\omega_2), \cdots, \boldsymbol{a}(\omega_n)) \in \mathbb{C}^n$, where $\{\omega_i\}_{i=1}^n$ are the distinct primitive m-th roots of unity and $\mathbf{U_m}$ denotes the following $n \times n$ transformation matrix for the canonical embedding, also known as the Vandermonde matrix of all the roots.

$$\mathbf{U_m} = \begin{bmatrix} 1 & \omega_1 & \omega_1^2 & \cdots & \omega_1^{\phi(m)-1} \\ 1 & \omega_2 & \omega_2^2 & \cdots & \omega_2^{\phi(m)-1} \\ & & & \vdots & \\ 1 & \omega_{\phi(m)} & \omega_{\phi(m)}^2 & \cdots & \omega_{\phi(m)}^{\phi(m)-1} \end{bmatrix} \in \mathbb{C}^{\phi(m)\times\phi(m)}. \tag{1}$$

To bound $\|\boldsymbol{sc}\|$ in the ring $\mathbb{Z}_q[x]/\langle x^n+1\rangle$, HAETAE leveraged the property that $\|\overrightarrow{\boldsymbol{a}}\|$ is equal to $\|\mathcal{C}(\boldsymbol{a})\|/\sqrt{n}$. However, this equality does not hold in the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. Thus, we first generalize the relationship between $\|\overrightarrow{\boldsymbol{a}}\|$ and $\|\mathcal{C}(\boldsymbol{a})\|$ in the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle \Phi_m(x)\rangle$. Using the relationship, along with the properties of the cyclotomic trinomials, we derive a bound on $\|\boldsymbol{sc}\|$ in the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$.

Let $\|\mathbf{U_m}\|$ denote the spectral norm of $\mathbf{U_m}$, i.e., $\|\mathbf{U_m}\| = \max_{\|\mathbf{x}\|\neq 0} \frac{\|\mathbf{U_m x}\|}{\|\mathbf{x}\|}$ for $\mathbf{x} \in \mathbb{C}^n$, and let $s_1(\mathbf{U_m})$ be the largest singular value of $\mathbf{U_m}$. It is well known that $\|\mathbf{U_m}\| = s_1(\mathbf{U_m})$ and $\|\mathbf{U_m x}\| \leq \|\mathbf{U_m}\| \cdot \|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{C}^n$. Before deriving the generalized relationship between $\|\overrightarrow{\boldsymbol{a}}\|$ and $\|\mathcal{C}(\boldsymbol{a})\|$, we introduce the following lemma about the singular value of $\mathbf{U_m}$, based on the previous works [22, 23, 2].

Lemma 4.1 (Upper bound of $s_1(\mathbf{U_m})$ [23, 2])**.** Let $s_1(\mathbf{U_m})$ is the largest singular value of $\mathbf{U_m}$, where $\mathbf{U_m}$ is the Vandermonde matrix in (1). For any positive integer m that corresponds to a cyclotomic polynomial $\Phi_m(x)$,

$$s_1(\mathbf{U_m}) \leq \sqrt{\gamma(m)}$$

where $\gamma(m) = \begin{cases} m & \textit{if } m \textit{ is odd} \\ m/2 & \textit{if } m \textit{ is even.} \end{cases}$ The equality is satisfied when m is power of prime.

Lemma 4.2 (Lower bound of $s_n(\mathbf{U_m})$ [22])**.** Let $s_n(\mathbf{U_m})$ is the smallest singular value of $\mathbf{U_m}$ where $\mathbf{U_m}$ is the Vandermonde matrix in (1). For any positive integer m that corresponds to a cyclotomic polynomial $\Phi_m(x)$, the smallest singular value of $\mathbf{U_m}$ is:

$$s_n(\mathbf{U_m}) = \sqrt{m/rad(m)},$$

where $rad(m)$ is a product of all primes dividing m.

Now, we present the following two useful inequalities between $\|\overrightarrow{\boldsymbol{a}}\|$ and $\|\mathcal{C}(\boldsymbol{a})\|$.

Lemma 4.3 (Upper bound of canonical embedding)**.** Let $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle\Phi_m(x)\rangle$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial, and $\gamma(m) = \begin{cases} m & \textit{if } m \textit{ is odd} \\ m/2 & \textit{if } m \textit{ is even.} \end{cases}$ For $\boldsymbol{a} \in \mathcal{R}_q$, the upper bound of $\|\mathcal{C}(\boldsymbol{a})\|$ can be expressed as follows:

$$\|\mathcal{C}(\boldsymbol{a})\| \leq \sqrt{\gamma(m)} \cdot \|\overrightarrow{\boldsymbol{a}}\|.$$

Proof. Note that we can represent $\mathcal{C}(\boldsymbol{a}) = \mathbf{U_m} \cdot \overrightarrow{\boldsymbol{a}}$. Using the fact that $\|\mathbf{U_m}\| = s_1(\mathbf{U_m})$ and $\|\mathbf{U_m x}\| \leq \|\mathbf{U_m}\| \cdot \|\mathbf{x}\|$ for $\mathbf{x} \in \mathbb{C}^n$, and also $s_1(\mathbf{U_m}) \leq \sqrt{\gamma(m)}$ from Lemma 4.1, we can bound $\|\mathcal{C}(\boldsymbol{a})\|$ as: $\|\mathcal{C}(\boldsymbol{a})\| = \|\mathbf{U_m} \cdot \overrightarrow{\boldsymbol{a}}\| \leq \|\mathbf{U_m}\| \cdot \|\overrightarrow{\boldsymbol{a}}\| = s_1(\mathbf{U_m}) \cdot \|\overrightarrow{\boldsymbol{a}}\| \leq \sqrt{\gamma(m)}\|\overrightarrow{\boldsymbol{a}}\|$, as required. □

It is trivial that $s_n(\mathbf{U_m})$ is the inverse of the largest singular value $s_1(\mathbf{U}_m^{-1})$, i.e., $s_n(\mathbf{U_m}) = 1/s_1(\mathbf{U}_m^{-1})$. Based on this, we derive the following result.

Lemma 4.4 (Upper bound of coefficient embedding)**.** For all $\boldsymbol{a} \in \mathcal{R}_q$, the upper bound of $\|\overrightarrow{\boldsymbol{a}}\|$ is

$$\|\overrightarrow{\boldsymbol{a}}\| \leq \sqrt{\frac{rad(m)}{m}} \cdot \|\mathcal{C}(\boldsymbol{a})\|,$$

where $rad(m)$ is a product of all primes dividing m.

Proof. Note that we can represent $\overrightarrow{\boldsymbol{a}} = {\mathbf{U_m}}^{-1}\mathcal{C}(\boldsymbol{a})$. By Lemma 4.2, we can bound $\|\overrightarrow{\boldsymbol{a}}\|$ as: $\|\overrightarrow{\boldsymbol{a}}\| = \|{\mathbf{U_m}}^{-1}\psi(\boldsymbol{a})\| \leq \|{\mathbf{U_m}}^{-1}\| \cdot \|\mathcal{C}(\boldsymbol{a})\| = s_1({\mathbf{U_m}}^{-1}) \cdot \|\mathcal{C}(\boldsymbol{a})\| = \frac{1}{s_n(\mathbf{U_m})} \cdot \|\mathcal{C}(\boldsymbol{a})\| = \sqrt{\frac{rad(m)}{m}} \cdot \|\mathcal{C}(\boldsymbol{a})\|$, as required. □

Now we propose a new bound on $\|\boldsymbol{sc}\|$ in the ring $\mathbb{Z}_q/\langle x^n - x^{n/2} + 1\rangle$, based on the above Lemma 4.3 and 4.4. Below is our new result that is obtained by following HAETAE. In the lemma below, we define the $\bmod$ operation differently: for a given integer a and modulus q, $a \bmod q$ is equal to the unique integer a' in range $[0, q)$.

Lemma 4.5 (Upper bound of $\|\boldsymbol{sc}\|$ in $\mathbb{Z}_q/\langle x^n - x^{n/2} + 1\rangle$)**.** Set the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$ for $n = 2^a 3^b$ where a and b are positive integers. For any challenge $\boldsymbol{c} \in \mathcal{R}_{n,\tau}$ with Hamming weight τ (such that $\tau \geq 2$) and a secret $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2) \in \mathcal{R}_q^2$, the value $\|\boldsymbol{sc}\|^2$ is upper-bounded by

$$\mathcal{N}_\tau(\boldsymbol{s}) := \frac{2}{n}\Big\{\tau^2 \sum_{i=1}^{k} \max_{j}^{i-\mathtt{th}} \|\mathbf{s}(\omega_j)\|^2 + r \cdot \max_{j}^{(k+1)-\mathtt{th}} \|\mathbf{s}(\omega_j)\|^2\Big\}, \tag{2}$$

where $k = \lfloor \frac{3n}{2\tau} \rfloor$, $r = \frac{3n}{2}\tau \bmod \tau^2$, and ω_j's are the primitive $3n$-th roots of unity.

Proof. Notice that $m = 3n = 2^a 3^{b+1}$ and $rad(m) = 2 \cdot 3$. By Lemma 4.4, we see that $\|\overrightarrow{\boldsymbol{s_i c}}\|^2 \leq \frac{rad(m)}{m} \cdot \|\mathcal{C}(\boldsymbol{s_i c})\|^2 = (2/n) \cdot \|\mathcal{C}(\boldsymbol{s_i c})\|^2$ for $i = 1, 2$. Also, by the homomorphic property of the canonical

embedding $\mathcal{C}$, we have $\|\mathcal{C}(\boldsymbol{s_i c})\| = \|\mathcal{C}(\boldsymbol{s_i}) \cdot \boldsymbol{\mathcal{C}(c)}\|$. Then, we rewrite $\|\boldsymbol{sc}\|^2$ as:

$$\begin{aligned}
\|\mathbf{sc}\|^2 &= \|\boldsymbol{s}_1\boldsymbol{c}\|^2 + \|\boldsymbol{s}_2\boldsymbol{c}\|^2 \\
&\leq \frac{2}{n}\|\mathcal{C}(\boldsymbol{s}_1\boldsymbol{c})\|^2 + \frac{2}{n}\|\mathcal{C}(\boldsymbol{s}_2\boldsymbol{c})\|^2 \\
&= \frac{2}{n}\big(\|\mathcal{C}(\boldsymbol{s}_1)\cdot\mathcal{C}(\boldsymbol{c})\|^2 + \|\mathcal{C}(\boldsymbol{s}_1)\cdot\mathcal{C}(\boldsymbol{c})\|^2\big) \\
&= \frac{2}{n}\bigg(\sum_{i=1}^{n}|\boldsymbol{s}_1(\omega_i)\boldsymbol{c}(\omega_i)|^2 + \sum_{i=1}^{n}|\boldsymbol{s}_2(\omega_i)\boldsymbol{c}(\omega_i)|^2\bigg) \\
&= \frac{2}{n}\sum_{i=1}^{n}(|\boldsymbol{s}_1(\omega_i)|^2 + |\boldsymbol{s}_2(\omega_i)|^2)|\boldsymbol{c}(\omega_i)|^2 \\
&= \frac{2}{n}\sum_{i=1}^{n}\|\boldsymbol{s}(\omega_i)\|^2\cdot|\boldsymbol{c}(\omega_i)|^2,
\end{aligned}$$

where $\boldsymbol{s}(\omega_i) = (\boldsymbol{s}_1(\omega_i), \boldsymbol{s}_2(\omega_i))$.

According to Lemma 4.3, we have the relation $\sum_{i=1}^{n}|\boldsymbol{c}(\omega_i)|^2 \leq \gamma(3n)\cdot\|\overrightarrow{\boldsymbol{c}}\|^2 = \frac{3n}{2}\tau$, where $\|\overrightarrow{\boldsymbol{c}}\|^2 = \tau$. Also, it is trivial that $|\mathbf{c}(\omega_i)|^2 = |\omega_{i,1} + \omega_{i,2} + ... + \omega_{i,\tau}|^2 \leq \tau^2$, where $\omega_{i,j}$'s are the rearranged primitive $3n$-th roots of unity. Let $k = \lfloor\frac{3n\tau}{2}\cdot\frac{1}{\tau^2}\rfloor = \lfloor\frac{3n}{2\tau}\rfloor$ and $r = \frac{3n}{2}\tau \bmod \tau^2$. This means that k is the maximum number of values $|\boldsymbol{c}(\omega_i)|^2$ that can be equal to τ^2, and r becomes $\frac{3n\tau}{2} - k\tau^2$, when $\sum_{i=1}^{n}|\boldsymbol{c}(\omega_i)|^2 = \frac{3n}{2}\tau$.

We now bound $\sum_{i=1}^{n}\big\{\|\boldsymbol{s}(\omega_i)\|^2\cdot|\boldsymbol{c}(\omega_i)|^2\big\}$ by rearranging the values of $\|\boldsymbol{s}(\omega_i)\|$ in a decreasing order,

$$\|\boldsymbol{s}(\omega_{\sigma(1)})\| \geq \|\boldsymbol{s}(\omega_{\sigma(2)})\| \geq \cdots \geq \|\boldsymbol{s}(\omega_{\sigma(n)})\|,$$

where σ is a permutation for the indices. Then, we have

$$\sum_{i=1}^{n}\|\boldsymbol{s}(\omega_i)\|^2\cdot|\boldsymbol{c}(\omega_i)|^2 \leq \sum_{i=1}^{k}\|\boldsymbol{s}(\omega_{\sigma(i)})\|^2\cdot|\boldsymbol{c}(\omega_{\sigma(i)})|^2 + \sum_{i=k+1}^{n}\|\boldsymbol{s}(\omega_{\sigma(i)})\|^2\cdot|\boldsymbol{c}(\omega_{\sigma(i)})|^2.$$

Then, it reaches the maximum when the k largest $\|\boldsymbol{s}(\omega_i)\|^2$'s are multiplied with τ^2, i.e.,

$$\begin{aligned}
\sum_{i=1}^{n}\|\boldsymbol{s}(\omega_i)\|^2\cdot|\boldsymbol{c}(\omega_i)|^2 &\leq \sum_{i=1}^{k}\tau^2\|\boldsymbol{s}(\omega_{\sigma(i)})\|^2 + \big(\sum_{i=1}^{n}|\boldsymbol{c}(\omega_i)|^2 - k\tau^2\big)\|\boldsymbol{s}(\omega_{\sigma(k+1)})\|^2 \\
&= \tau^2\sum_{i=1}^{k}\|\boldsymbol{s}(\omega_{\sigma(i)})\|^2 + r\cdot\|\boldsymbol{s}(\omega_{\sigma(k+1)})\|^2,
\end{aligned}$$

which concludes the proof. □

4.2 Bounding $\|\boldsymbol{sc}\|$ with Gram matrix

The main idea of bounding $\|\boldsymbol{sc}\|$ in BLISS is based on the fact that $\|\boldsymbol{sc}\|^2 = \boldsymbol{c}^T\mathbf{S}^T\mathbf{S}\boldsymbol{c}$, where $\mathbf{S}$ is a matrix derived from a polynomial $\boldsymbol{s}$ in a ring R_q. Intuitively, considering that $\|\boldsymbol{c}\|_1 = \tau$, the upper bound on $\|\boldsymbol{sc}\|^2$ is obtained by summing the τ largest values in each row of $\mathbf{S}^T\mathbf{S}$, sorting the resulting vector, and summing its τ largest components of the vector. Obviously, given $\boldsymbol{s} \in R_q$, the upper bound on $\|\boldsymbol{sc}\|^2$ holds for any $\boldsymbol{c} \in R_{n,\tau}$ [11]. This observation also holds in the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$, and the only thing we need

to consider is constructing an appropriate matrix $\mathbf{S}$ (and then $\mathbf{S}^T\mathbf{S}$), once $\boldsymbol{s}$ is chosen. More precisely, the signing key $\boldsymbol{s}$ in NTRU+Sign is composed of $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2) \in \mathcal{R}_q^2$. Using this fact, we establish the following lemma:

Lemma 4.6 (Upper bound of $\|\boldsymbol{sc}\|$ with Gram matrix)**.** Let $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$, where $n = 2^a 3^b$ for some positive integer a and b, and let $\boldsymbol{c} \in \mathcal{R}_{n,\tau}$. For $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2) \in \mathcal{R}_q^2$, let $\mathbf{S}_1$ and $\mathbf{S}_2$ be the matrices corresponding to the polynomials $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ in R_q, respectively. Then, the upper bound $\mathcal{N}_\tau(\boldsymbol{s})$ of $\|\boldsymbol{sc}\|^2$ is given by:

$$\mathcal{N}_\tau(\boldsymbol{s}) := \max_{\mathbf{I}\subset\{1,\dots,n\},\#\mathbf{I}=\tau} \sum_{i\in\mathbf{I}} \left(\max_{\mathbf{J}\subset\{1,\dots,n\},\#\mathbf{J}=\tau} \sum_{j\in\mathbf{J}} \left((\mathbf{T}_1)_{i,j} + (\mathbf{T}_2)_{i,j}\right)\right), \tag{3}$$

where $\mathbf{T}_i = \mathbf{S}_i^T\mathbf{S}_i \in \mathbb{R}^{n\times n}$ for $i = 1, 2$.

Proof. The proof is straightforward if we set $\mathbf{S} = \begin{bmatrix}\mathbf{S}_1 \\ \mathbf{S}_2\end{bmatrix}$. □

Before describing how the matrices $\{\mathbf{T}_i\}_{i=1,2}$ are obtained, we define some notations. For a vector $\overrightarrow{\boldsymbol{d}} = (a_0, a_1, \cdots, a_{n-1}) \in \mathbb{Z}_q^n$, which represents the coefficient embedding of $\boldsymbol{a} \in \mathcal{R}_q$, $x^i \cdot \overrightarrow{\boldsymbol{d}}$ denotes the coefficient embedding corresponding to the multiplication $x^i\boldsymbol{a}$ (for some integer i) in $\mathcal{R}_q$. In general, given $\boldsymbol{s}_i \in R_q$, the corresponding matrix $\mathbf{T}_i$ is represented, regardless of the ring choice, as follows:

$$\mathbf{T}_i = \begin{bmatrix} \langle \overrightarrow{\boldsymbol{s}_i}, \overrightarrow{\boldsymbol{s}_i}\rangle & \langle \overrightarrow{\boldsymbol{s}_i}, x\cdot\overrightarrow{\boldsymbol{s}_i}\rangle & \cdots & \langle \overrightarrow{\boldsymbol{s}_i}, x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}\rangle \\ \langle x\cdot\overrightarrow{\boldsymbol{s}_i}, \overrightarrow{\boldsymbol{s}_i}\rangle & \langle x\cdot\overrightarrow{\boldsymbol{s}_i}, x\cdot\overrightarrow{\boldsymbol{s}_i}\rangle & \cdots & \langle x\cdot\overrightarrow{\boldsymbol{s}_i}, x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}\rangle \\ \vdots & \vdots & & \vdots \\ \langle x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}, \overrightarrow{\boldsymbol{s}_i}\rangle & \langle x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}, x\cdot\overrightarrow{\boldsymbol{s}_i}\rangle & \cdots & \langle x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}, x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}\rangle \end{bmatrix}, \tag{4}$$

where $\mathbf{S}_i = \left[\overrightarrow{\boldsymbol{s}_i},\ \ x\cdot\overrightarrow{\boldsymbol{s}_i},\ \ \cdots,\ \ x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_i}\right]$. As shown in BLISS, a ring $\mathbb{Z}[x]/\langle x^n + 1\rangle$ gives a simple representation of the matrix $\mathbf{T} := \mathbf{T}_1 + \mathbf{T}_2 = [\boldsymbol{t}, x\cdot\boldsymbol{t}, \dots, x^{n-1}\cdot\boldsymbol{t}]$, where

$$\boldsymbol{t} = \left(\langle \overrightarrow{\boldsymbol{s}_1}, \overrightarrow{\boldsymbol{s}_1}\rangle + \langle \overrightarrow{\boldsymbol{s}_2}, \overrightarrow{\boldsymbol{s}_2}\rangle,\ \langle \overrightarrow{\boldsymbol{s}_1}, x\cdot\overrightarrow{\boldsymbol{s}_1}\rangle + \langle \overrightarrow{\boldsymbol{s}_2}, x\cdot\overrightarrow{\boldsymbol{s}_2}\rangle, \cdots,\ \langle \overrightarrow{\boldsymbol{s}_1}, x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_1}\rangle + \langle \overrightarrow{\boldsymbol{s}_2}, x^{n-1}\cdot\overrightarrow{\boldsymbol{s}_2}\rangle\right).$$

This simplicity is because of the following equality:

$$\langle x^l\cdot\overrightarrow{\boldsymbol{s}_i}, x^j\cdot\overrightarrow{\boldsymbol{s}_i}\rangle = \begin{cases} \langle x^{l+1}\cdot\overrightarrow{\boldsymbol{s}_i}, x^{j+1}\cdot\overrightarrow{\boldsymbol{s}_i}\rangle & \text{for } 0 \le l \le n-2,\ 0 \le j \le n-2, \\ -\langle x^{l+1}\cdot\overrightarrow{\boldsymbol{s}_i}, \overrightarrow{\boldsymbol{s}_i}\rangle & \text{for } 0 \le l \le n-2,\ j = n-1. \end{cases} \tag{5}$$

However, the Equation (5) does not hold in a ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. We can check this difference in a simple example, where n is 6. For $\boldsymbol{s} = s_0 + s_1x + \cdots + s_5x^5 \in \mathbb{Z}_q[x]/\langle x^6 - x^3 + 1\rangle$, the corresponding matrix $\mathbf{S}$ is:

$$\begin{aligned}\mathbf{S} &= \left[\overrightarrow{\boldsymbol{s}},\ x\cdot\overrightarrow{\boldsymbol{s}},\ x^2\cdot\overrightarrow{\boldsymbol{s}},\ x^3\cdot\overrightarrow{\boldsymbol{s}},\ x^4\cdot\overrightarrow{\boldsymbol{s}},\ x^5\cdot\overrightarrow{\boldsymbol{s}}\right] \\ &= \begin{bmatrix} s_0 & -s_5 & -s_4 & -s_3 & -s_2-s_5 & -s_1-s_4 \\ s_1 & s_0 & -s_5 & -s_4 & -s_3 & -s_2-s_5 \\ s_2 & s_1 & s_0 & -s_5 & -s_4 & -s_3 \\ s_3 & s_2+s_5 & s_1+s_4 & s_0+s_3 & s_2 & s_1 \\ s_4 & s_3 & s_2+s_5 & s_1+s_4 & s_0+s_3 & s_2 \\ s_5 & s_4 & s_3 & s_2+s_5 & s_1+s_4 & s_0+s_3 \end{bmatrix}.\end{aligned}$$

To obtain some components associated with (4), we compute $\langle \overrightarrow{\boldsymbol{s}}, x \cdot \overrightarrow{\boldsymbol{s}} \rangle$ and $\langle x \cdot \overrightarrow{\boldsymbol{s}}, x^2 \cdot \overrightarrow{\boldsymbol{s}} \rangle$, and the difference between them is shown as the following (blue) colored values:

$$\begin{aligned}\langle \overrightarrow{\boldsymbol{s}}, x \cdot \overrightarrow{\boldsymbol{s}} \rangle &= s_0(-s_5) + s_1 s_0 + s_2 s_1 + s_3(s_2 + s_5) + s_4 s_3 + s_5 s_4, \\ \langle x \cdot \overrightarrow{\boldsymbol{s}}, x^2 \cdot \overrightarrow{\boldsymbol{s}} \rangle &= (-s_5)(-s_4) + s_0(-s_5) + s_1 s_0 + (s_2 + s_5)(s_1 + s_4) + s_3(s_2 + s_5) + s_4 s_3.\end{aligned}$$

Then, we can get a relation as:

$$\langle \overrightarrow{\boldsymbol{s}}, x \cdot \overrightarrow{\boldsymbol{s}} \rangle = \langle x \cdot \overrightarrow{\boldsymbol{s}}, x^2 \cdot \overrightarrow{\boldsymbol{s}} \rangle - (s_2 + s_5)(s_1 + s_4) + s_2 s_1.$$

Based on this observation, we can extend this relation to represent the components of $\mathbf{T}_i$ in (4), which is specific to the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1 \rangle$. More precisely, for all integers l and k such that $0 \le l \le n-2$ and $0 \le k \le n-1$, we have

$$\langle x^l \cdot \overrightarrow{\boldsymbol{s}_i}, x^k \cdot \overrightarrow{\boldsymbol{s}_i} \rangle = \begin{cases} \langle x^{l+1} \cdot \overrightarrow{\boldsymbol{s}_i}, x^{k+1} \cdot \overrightarrow{\boldsymbol{s}_i} \rangle - (x^{l+1} \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}]} (x^{k+1} \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}]} + (x^l \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}-1]} (x^k \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}-1]} \\ \qquad \text{for } 0 \le l \le n-2,\ 0 \le k \le n-2 \\ \langle x^{l+1} \cdot \overrightarrow{\boldsymbol{s}_i}, x^{\frac{n}{2}} \cdot \overrightarrow{\boldsymbol{s}_i} - \overrightarrow{\boldsymbol{s}_i} \rangle - (x^{l+1} \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}]} (x^{\frac{n}{2}} \cdot \overrightarrow{\boldsymbol{s}_i} - \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}]} + (x^l \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}-1]} (x^{n-1} \cdot \overrightarrow{\boldsymbol{s}_i})_{[\frac{n}{2}-1]} \\ \qquad \text{for } 0 \le l \le n-2,\ k = n-1 \end{cases}, \tag{6}$$

where $(\overrightarrow{\boldsymbol{a}})_{[j]}$ denotes the j-th component of $\overrightarrow{\boldsymbol{a}} = (a_0, a_1, \ldots, a_j, \ldots, a_{n-1})$. Using this equation, we can compute all components of $\mathbf{T}_i$ in (4), when a polynomial $\boldsymbol{s}_i$ in $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1 \rangle$ is given. Once we get two matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ associated with a signing key $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$, we can compute the upper bound $\sqrt{\mathcal{N}_\tau(\boldsymbol{s})}$ of $\|\boldsymbol{sc}\|$ from the Equation 4.6.

5 Parameter Selection for NTRU+Sign

5.1 Selection of $\|\boldsymbol{sc}\|$-Bounding Method

We begin by choosing one of the two $\|\boldsymbol{sc}\|$-bounding method described in Section 4. Our goal is to minimize the upper-bound of $\|\boldsymbol{sc}\|$. To define the ring $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1 \rangle$ where $n = 2^i 3^j$ for some positive integers i and j, we select n as $648 = 2^3 3^4$, $972 = 2^2 3^5$, and $1296 = 2^4 3^4$, based on their target security levels.[1] For each fixed n, we compute the value τ such that $\binom{n}{\tau} \ge 2^\lambda$ for the target security level λ, which determines the min-entropy of the challenge space. Using a pair of (n, τ) and a signing key $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$, we can calculate $\mathcal{N}_\tau(\boldsymbol{s})$, based on the Equations 2 and 3. Note that each coefficient of $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ is sampled according to the distribution ψ_1. We repeat this process $1,000$ times by generating $1,000$ distinct signing keys $\{\boldsymbol{s}\}$. After sorting the resulting $1,000$ vaules of $\mathcal{N}_\tau(\boldsymbol{s})$ values in increasing order, we select the 900-th value as the upper bound of $\|\boldsymbol{sc}\|$, ensuring a approximately 90% acceptance rate of a signing key $\boldsymbol{s}$ in the **KeyGen** algorithm. We refer to $B_{\boldsymbol{sc}}$ as the chosen upper bound of $\|\boldsymbol{sc}\|$ (with respect to the pair (n, τ)). Table 2 presents the result of our experiments, showing that the Gram matrix method described in Section 4.2 gives smaller $B_{\boldsymbol{sc}}$ values than the FFT method in Section 4.1. Based on our experiments, we choose the Gram matrix method to set $B_{\boldsymbol{sc}}$, and indeed $B_{\boldsymbol{sc}}$ values (shown in Table 2) will be used to set other parameters of NTRU+Sign.

[1] The exact security levels will be estimated later with other related parameters.

Table 2: Upper bound of $\|\boldsymbol{sc}\|$ with 90% acceptance rate

(n, τ)	NTRU+Sign-648 $(648, 35)$	NTRU+Sign-972 $(972, 38)$	NTRU+Sign-1296 $(1296, 41)$
FFT method	475	605	730
Gram matrix method	372	466	552

The above procedure means that, for a (candidate) signing key $\boldsymbol{s}$, the $\mathbf{KeyGen}$ algorithm needs to calculate $\mathcal{N}_\tau(\boldsymbol{s})$, using the Gram matrix method in order to check if $(B_{\boldsymbol{sc}})^2 < \mathcal{N}_\tau(\boldsymbol{s})$. Obviously, computing $\mathcal{N}_\tau(\boldsymbol{s})$ requires a significant amount of computation, compared to the FFT method. To mitigate this computational burden in the $\mathbf{KeyGen}$ algorithm, we set the acceptance rate (denoted as ϵ_{sc}) high around 90% to reduce the number of repetitions. Also, it is worth noting that there is a trade-off between ϵ_{sc} and $B_{\boldsymbol{sc}}$; if we lower ϵ_{sc} by allowing more computation of the $\mathbf{KeyGen}$ algorithm, we can set $B_{\boldsymbol{sc}}$ to be smaller, which could further reduce the signature size.

5.2 Concrete Parameters

As we have chosen n, τ, and $B_{\boldsymbol{sc}}$, we can determine the remaining parameters step by step. The standard deviation σ of the discrete Gaussian distribution $\mathcal{D}_\sigma$ and the expected number M of repetitions in rejection sampling are given by $\sigma = \hat{\alpha} B_{\boldsymbol{sc}}$ and $M = \exp(1/(2\hat{\alpha}^2))$ by setting an appropriate $\hat{\alpha} \in [0, 1]$. If $\hat{\alpha}$ is close to 1, M approaches 1, but σ increases, leading to a larger signature size. Therefore, it is necessary to find a balance between M and σ by adjusting $\hat{\alpha}$. In our setting, we set $\hat{\alpha}$ to be 0.56 to keep the expected total number of repetitions M_{total}[2] in the $\mathbf{Sign}$ algorithm between 5 and 6. Next, we select d as the number of dropped bits in the commitment. As shown in [9], the choice of d must satisfy the requirement that the commitment's min-entropy α is much larger than the security parameter λ, satisfying $2^{-\alpha} \ll 2^{-2\lambda}$ as shown in Table 3. We set $d = 8$ for $\lambda = 118, 193$ and $d = 9$ for $\lambda = 264$.

Next, we choose the modulus q under the requirement that $q \equiv 1 \bmod 3n/\hat{b}$, where $\hat{b}$ is the degree of the lowest level of a polynomial in NTT decomposition. Once q is chosen, we can obtain the other modulus p such that $p = (q + q_0)/2^d$, where $(-q_0) \equiv q \bmod 2^d$, and compute the infinite norm bound B_∞ such that $B_\infty \leq (q - 2)/4 - 2^{d-1} - 1$. Notice that the inequality for B_∞ is derived from the security proof of [27]. When setting $B_\infty = (q - 2)/4 - 2^{d-1} - 1$, the inequality condition necessary for the security proof is satisfied for signatures that pass line 10 of the $\mathbf{Sign}$ algorithm.

We then need to set the Euclidean norm bound B_2 in an experimental manner. After generating $1,000$ signatures $\{(\boldsymbol{z}_1, \boldsymbol{h}, \boldsymbol{c})\}$ (without executing line 9 of the $\mathbf{Sign}$ algorithm), we compute $1,000$ values of $\|(\boldsymbol{z}_1, 2^d\boldsymbol{h})\|$ and set B_2 as the largest Euclidean norm, hoping that the rejection from line 9 of the $\mathbf{Sign}$ algorithm rarely occurs. Importantly, B_2 is linked to the SIS bound, $B_2 + (2^d + q_0)\sqrt{n}$, which is used to evaluate the hardness of a related SIS problem. There is also a trade-off between B_2 and the hardness of a SIS problem: if B_2 is smaller (which implies more repetitions in the $\mathbf{Sign}$ algorithm), then the SIS problem becomes harder, and vice versa. Once all parameters are set, we compute the size of an encoded signature, using range Asymmetric Numeral System (rANS) encoding [14, 18].

Based on the above analysis, Table 4 presents three parameter sets for NTRU+Sign-$\{648, 972, 1296\}$,

[2] $M_{total} = M \times M_{eq}$, where $1/M$ is the repetition rate related to rejection sampling and $1/M_{eq}$ is the repetition rate related to equality check.

Table 3: Requirements for Parameter Selection

Parameter Requirements		
1)	Dimension of a ring $\mathcal{R}_q$	$n = 2^i 3^j$ for $\{i, j\} \in \mathbb{N}^2$
2)	Modulus for NTT	$q \equiv 1 \bmod 3n/\hat{b}$
3)	Standard deviation of Gaussian distribution	$\sigma = B_{\boldsymbol{sc}} \cdot \hat{\alpha}$
4)	Expected # of repetitions in rejection sampling	$M = \exp(1/(2\hat{\alpha}^2))$
5)	Modulus for hint generation	$p = (q + q_0)/2^d$ where $(-q_0) \equiv q \bmod 2^d$
6)	Min-entropy of commitment	$2^{-\alpha} = (2^d/(\sqrt{2\pi}\sigma - 1))^n \ll 2^{-2\lambda}$
7)	Challenge space	$\binom{n}{\tau} \geq 2^\lambda$
8)	Infinite norm bound	$B_\infty \leq (q-2)/4 - 2^{d-1} - 1$

meeting the requirements in Table 3. Each scheme aims to achieve 118, 193, and 264 bits of security in the classical random oracle model, respectively. For security estimation, we use the estimators from [8, 7] to evaluate the hardness of related SIS and NTRU problems determined by these parameters. Compared to the previous schemes such as Dilithium and HAETAE, NTRU+Sign achieves smaller signature sizes at similar security levels, as shown in Table 5. For instance, at approximately the 120-bit security level, the signature sizes of Dilithium-2 and HAETAE-120 are about $2.4(\approx 2420/1009)$ times and $1.5(\approx 1474/1009)$ times longer than that of NTRU+Sign-648, respectively. Moreover, in case of the signature size plus the verification key size, Dilithium-2 and HAETAE-120 are about $1.8(\approx 3732/2062)$ times and $1.2(\approx 2466/2062)$ times longer than NTRU+Sign-648, respectively.

6 Performance Analysis

We provide a reference implementation for NTRU+Sign-$\{648, 972, 1296\}$. For constant-time implementation resistant to side-channel attacks [6, 13, 25], we follow the technique of GALACTICS[3] [5] as in previous work [27], where polynomial approximations of transcendental functions are used to ensure constant-time operations related to discrete Gaussian sampling and rejection sampling in the **Sign** algorithm. The approximated polynomials and related tables for NTRU+Sign-$\{648, 972, 1296\}$ are given in Appendix A. In addition, we use the rANS encoding [14] to optimize the signature size.

6.1 Implementation changes according to $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$

NTT in $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. All polynomial multiplications are performed in $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. To accelerate these multiplications, we adapt the NTT technique [24] for $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$, which is essentially as fast as the NTT in the ring $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ with a power-of-two n. The main difference is that the modulus q must satisfy $q \equiv 1 \bmod 3n/\hat{b}$, where $\hat{b}$ is the degree of the lowest layer in the NTT decomposition. Indeed, for each pair (n, q) of NTRU+Sign-$\{648, 972, 1296\}$, we set up b to be 3 or 4.

Computation of $\mathcal{N}_\tau(\boldsymbol{s})$. As mentioned in Section 5.1, the $\mathbf{KeyGen}$ algorithm needs to compute $\mathcal{N}_\tau(\boldsymbol{s})$ for a candidate secret key $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2)$, using the Gram matrix method. $\mathcal{N}_\tau(\boldsymbol{s})$ is calculated based on

[3]The open source is available at "https://github.com/espitau/GALACTICS"

Table 4: Parameter sets for NTRU+Sign-$\{648, 972, 1296\}$

Parameters		I	III	V
n	Dimension of a ring $\mathcal{R}_q$	648	972	1,296
q	Modulus	7,129	9,721	9,721
σ	Standard deviation of D_σ	208.32	260.96	309.12
$\hat{b}$	Degree of a polynomial of lowest level in NTT	3	3	4
τ	Hamming weight of $\boldsymbol{c}$	35	38	41
$\lfloor \log_2 \binom{n}{\tau} \rfloor$	Min-entropy of challenge space	192	227	258
d	# of dropped bits in commitment	8	8	9
M	Expected # of repetitions for rejection sampling	4.92	4.92	4.92
M_{eq}	Expected # of repetitions for equality check	1.14	1.16	1.08
M_{total}	$M_{total} = M \times M_{eq}$	5.65	5.7	5.33
$B_{\boldsymbol{sc}}$	Upper bound of $\|\boldsymbol{sc}\|$	372	466	552
$\epsilon_{\boldsymbol{sc}}$	Key acceptance rate	0.90	0.90	0.90
$\hat{\alpha}$	$\hat{\alpha} = \sigma / B_{\boldsymbol{sc}}$	0.56	0.56	0.56
α	Min-entropy of commitment	664	1,313	772
B_2	Euclidean norm bound	8,500	12,520	18,185
B_∞	Infinite norm bound	1,300	1,250	1,650
q_0	$(-q_0) \equiv q \bmod 2^d$	39	7	7
NTRU Hardness (Core-SVP)				
	BKZ block-size b	418	663	933
	Classical Core-SVP	121	193	272
	Quantum Core-SVP	107	170	239
SIS Hardness (Core-SVP)				
	BKZ block-size b	404	705	906
	Classical Core-SVP	118	206	264
	Quantum Core-SVP	103	181	232

the matrices $\{\mathbf{T}_i\}_{i=1,2}$ in Equation 4, which correspond to $\boldsymbol{s}_i$. In a naive implementation, it takes about $240,232$ K cycles to obtain $\mathcal{N}_\tau(\boldsymbol{s})$ for a given $\boldsymbol{s}$ for NTRU+Sign-648. To improve this, we employ two properties of the matrices $\{\mathbf{T}_i\}_{i=1,2}$. First, due to the symmetric property of $\{\mathbf{T}_i\}_{i=1,2}$, we can compute only the upper triangular components including diagonal entries. Second, once the first row of $\{\mathbf{T}_i\}_{i=1,2}$ is computed, the entries of all the other rows can be obtained by the sequential calculations from the first row to the last row, using the equation 6. Especially, the sequential calculations involve two multiplications with the $(\frac{n}{2}-1)$-th and the $\frac{n}{2}$-th components of the vectors $\overrightarrow{\boldsymbol{s}_i}, x \cdot \overrightarrow{\boldsymbol{s}_i}, \ldots, x^{n-1} \cdot \overrightarrow{\boldsymbol{s}_i}$, without performing full inner products. As a result, our new implementation reduces the computational cost of $\mathcal{N}_\tau(\boldsymbol{s})$ to $13,502$ K cycles for NTRU+Sign-648, achieving about 17.8 times speed-up.

Table 5: Performance Comparison between Dilithium, HAETAE, and NTRU+Sign

Algorithms	Classical Security	Size (bytes)			Performance (k cycles)		
		|sig|	|vk|	|sig|+|vk|	KeyGen	Sign	Verify
Dilithium-2	123	2,420	1,312	3,732	313	1,384	341
Dilithium-3	182	3,293	1,952	5,245	576	2,259	555
Dilithium-5	252	4,595	2,592	7,187	895	2,911	930
HAETAE-120	119	1,474	992	2,466	2,005	8,130	327
HAETAE-180	180	2,349	1,472	3,821	2,862	10,913	670
HAETAE-260	256	2,948	2,080	5,028	2,536	14,115	802
NTRU+Sign-648	118	1,009	1,053	2,062	12,991	4,107	156
NTRU+Sign-972	193	1,557	1,701	3,258	26,333	6,891	235
NTRU+Sign-1296	264	2,020	2,268	4,288	47,139	8,521	306

6.2 Comparison

Table 5 presents a performance comparison between Dilithium[4], HAETAE[5], and NTRU+Sign[6] at the three security levels: $\lambda = 120, 180, 260$. We run those schemes on an Intel(R) Core(TM) i7-8700K CPU @ 3.70 GHz with 16.0 GB of RAM, using their reference codes. We evaluate the performance of the $\mathbf{KeyGen}$, $\mathbf{Sign}$, and $\mathbf{Verify}$ algorithms with average cycles of $1,000$ trials. Table 5 shows that the $\mathbf{KeyGen}$ algorithm of NTRU+Sign is slower than the others. Obviously, this is due to our key rejection algorithm that requires computing the Gram matrix. Nevertheless, the $\mathbf{KeyGen}$ algorithm of NTRU+Sign is still practically usable in the sense that it takes about 11.9 ms(millisecond) on average in case of NTRU+Sign-1296. In terms of the signature generation speed, NTRU+Sign is between Dilithium and HAETAE. For instance, at the 120-bit security level, NTRU+Sign is approximately 3 times slower than Dilithium, but about 1.9 times faster than HAETAE. As already observed in [27], when relying on the FSwA framework, the speed of a signature generation depends more heavily on the sampling method used to generate a commitment $\boldsymbol{y}$, rather than on the computation of rejection sampling. Then, the efficiency of the $\mathbf{Sign}$ algorithm can be explained by the fact that, to generate $\boldsymbol{y}$, Dilithium uses a uniform sampling in hypercube, HAETAE uses a hyperball sampling, and NTRU+Sign uses a discrete Gaussian sampling. Finally, with respect to verification speed, NTRU+Sign is faster than the others at the same security level. As stated in [27], the speedup is because of a relatively small modulus q and absence of a need to reconstruct a part of a verification key from a seed during the $\mathbf{Verify}$ algorithm.

7 Discussion

In this paper, we suggest more NTRU+Sign signature schemes that work over $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. Because of the flexibility of choosing the dimension n, we can present three new signature schemes, called NTRU+Sign-$\{648, 972, 1296\}$, each of which aims to achieve $\{120, 190, 260\}$-bit security levels, respectively. To parametrize these schemes, the critical factor is minimizing the upper bound of $\|\boldsymbol{sc}\|$, since

[4]The reference code is available at "https://pq-crystals.org/index.shtml".

[5]The reference code is available at "https://kpqc.cryptolab.co.kr/haetae".

[6]The reference code is available at "https://github.com/GHH33/NTRU-T-plus-SIGN.git".

this upper bound determines the overall system parameters including the signature size. To compute the bound on $\|\boldsymbol{sc}\|$, we choose the Gram matrix method used in BLISS [11]. However, there is still a significant gap between the theoretical upper bound (from the Gram matrix method) and the largest $\|\boldsymbol{sc}\|$ value observed in practice. Indeed, for 10^6 signatures generated using 10^3 (distinct) signing keys $\{\boldsymbol{s}_i\}$ and 10^3 distinct messages $\{m_i\}$, the largest observed values of $\|\boldsymbol{sc}\|$ are $\{217, 278, 319\}$ for NTRU+Sign-$\{648, 972, 1296\}$, respectively. Those observed values are significantly smaller than the corresponding upper bounds $\{376, 470, 558\}$ shown in Table 2. Therefore, it would be interesting to develop a new and more precise method that sets a tighter upper bound on $\|\boldsymbol{sc}\|$ in a theoretical manner.

References

[1] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange: a new hope. In: Proceedings of the 25th USENIX Conference on Security Symposium. p. 327–343. SEC'16, USENIX Association, USA (2016)

[2] Attema, T., Cramer, R., Xing, C.: A note on short invertible ring elements and applications to cyclotomic and trinomials number fields. Mathematical Cryptology **1**(1), 45–70 (2021)

[3] 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

[4] 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

[5] 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

[6] 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

[7] Chen, C., Danba, O., Hoffstein, J., Hulsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P., Whyte, W., Zhang, Z., Saito, T., Yamakawa, T., Xagawa, K.: NTRU. Tech. rep., National Institute of Standards and Technology (2020), available at `https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions`

[8] 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

[9] 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

[10] Devevey, J., Passelègue, A., Stehlé, D.: G+G: A fiat-shamir lattice signature based on convolved gaussians. IACR Cryptol. ePrint Arch. **2023**, 1477 (2023), https://api.semanticscholar.org/CorpusID:263269468

[11] 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

[12] 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

[13] 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

[14] Espitau, T., Tibouchi, M., Wallet, A., Yu, Y.: Shorter hash-and-sign lattice-based signatures. In: Annual International Cryptology Conference. pp. 245–275. Springer (2022)

[15] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology — CRYPTO' 86. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987)

[16] Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., et al.: Falcon: Fast-fourier lattice-based compact signatures over ntru. Submission to the NIST's post-quantum cryptography standardization process **36**(5), 1–75 (2018)

[17] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. p. 197–206. STOC '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1374376.1374407, https://doi.org/10.1145/1374376.1374407

[18] Giesen, F.: Interleaved entropy coders (2014), https://arxiv.org/abs/1402.3392

[19] Kim, J., Park, J.H.: Ntru+: Compact construction of ntru using simple encoding method. IEEE Transactions on Information Forensics and Security **18**, 4760–4774 (2023). https://doi.org/10.1109/TIFS.2023.3299172

[20] 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

[21] 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

[22] Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. pp. 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

[23] Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018. pp. 204–224. Springer International Publishing, Cham (2018)

[24] Lyubashevsky, V., Seiler, G.: Nttru: Truly fast ntru using ntt. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(3), 180–201 (May 2019). https://doi.org/10.13154/tches.v2019.i3.180-201, https://tches.iacr.org/index.php/TCHES/article/view/8293

[25] 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

[26] Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[27] Woo, J., Kim, J., Hong, G.H., Lee, S., Kim, M., Lee, H., Park, J.H.: NTRU+Sign: Compact NTRU-Based Signatures Using Bimodal Distributions. Cryptology ePrint Archive, Paper 2025/106 (2025), https://eprint.iacr.org/2025/106

A Constant-time Implementation of Discrete Gaussian and Rejection Sampling

We refer to NTRU+Sign [27] for detailed explanations about parameters and polynomial approximations that follow in subsequent subsections.

A.1 Discrete Gaussian

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

Table 6: Parameters for constant-time discrete Gaussian sampler

parameter set	I	III	V
λ	118	193	264
n	648	972	1296
B_{sc}	372	466	552
σ	208.32	260.96	309.12
$\hat{\alpha}\,(=\sigma/B_{sc})$	0.56	0.56	0.56
$M\,(=\exp(1/(2\hat{\alpha}^2)))$	4.92	4.92	4.92
$k\,(=2^{\lfloor\log_2\sigma\rfloor})$	2^7	2^8	2^8
$\sigma_1\,(=\sigma/k)$	1.6275	1.01	1.207
τ_1	9	9	9
$w_1\,(=\lfloor\tau_1\sigma_1\rfloor)$	14	9	10
θ_1	86	86	86
ϑ_1	60	61	62
$P_{\text{exp}}^{I_1}$	P_{g648}	P_{g972}	P_{g1296}

Table 7: CDT for NTRU+Sign-648

i	cdt$[i]$
0	30,463,896,446,073,161,550,872,468
1	55,687,354,025,600,429,753,252,937
2	70,004,686,913,464,605,127,428,774
3	75,576,005,108,806,338,193,817,339
4	77,062,256,039,480,156,901,128,835
5	77,334,065,552,342,848,575,357,286
6	77,368,143,662,387,384,311,989,851
7	77,371,072,698,228,315,934,032,801
8	77,371,245,286,868,506,760,026,713
9	77,371,252,258,558,450,231,571,557
10	77,371,252,451,622,875,961,586,271
11	77,371,252,455,288,135,959,366,420
12	77,371,252,455,335,838,955,015,692
13	77,371,252,455,336,264,577,786,320
14	77,371,252,455,336,267,181,195,263

Table 8: CDT for NTRU+Sign-972

i	cdt$[i]$
0	43,525,753,036,693,317,709,751,015
1	70,427,168,452,549,206,935,188,574
2	76,778,454,067,146,715,743,456,116
3	77,351,257,787,814,540,776,010,011
4	77,370,991,450,127,548,357,347,746
5	77,371,251,147,304,660,124,260,041
6	77,371,252,452,827,398,800,863,750
7	77,371,252,455,334,427,622,851,656
8	77,371,252,455,336,266,665,869,181
9	77,371,252,455,336,267,181,195,263

Table 9: CDT for NTRU+Sign-1296

i	$\mathrm{cdt}[i]$
0	38,428,599,979,725,204,174,068,606
1	65,701,121,986,719,449,682,357,581
2	75,449,615,690,336,689,573,577,967
3	77,204,672,612,493,178,206,663,406
4	77,363,815,185,019,285,173,139,720
5	77,371,083,320,565,963,821,849,574
6	77,371,250,507,039,939,108,699,962
7	77,371,252,444,000,309,379,775,658
8	77,371,252,455,302,999,176,011,080
9	77,371,252,455,336,218,008,005,127
10	77,371,252,455,336,267,181,195,263

$$
\begin{aligned}
P_{\text{g648}}(x) = ((((((((((&\tfrac{26578561361878513}{2^{60}} \cdot x \cdot 2^{-22} \\
&+ \tfrac{9147031620286697}{2^{60}}) \cdot x \cdot 2^{-13} \\
&+ \tfrac{1049304851576759935}{2^{60}}) \cdot x \cdot 2^{-20} \\
&+ \tfrac{790911809217535903}{2^{60}}) \cdot x \cdot 2^{-26} \\
&+ \tfrac{8193691819476293}{2^{60}}) \cdot x \cdot 2^{-26} \\
&+ \tfrac{74188016838301}{2^{60}}) \cdot x \cdot 2^{-11} \\
&+ \tfrac{18864681459464431}{2^{60}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{31229957314226703}{2^{60}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{20680125693366481}{2^{60}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{20541218970628285}{2^{60}}) \cdot x \cdot 2^{-17} \\
&+ \tfrac{27204327023802679}{2^{60}}) \cdot x \cdot 2^{-11} \\
&+ 1
\end{aligned}
$$

Figure 1: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over I_1 for NTRU+Sign-648

$$
\begin{aligned}
P_{\text{g972}}(x) = (((((((((&(\tfrac{1532522478042276863}{2^{61}} \cdot x \cdot 2^{-21} \\
&+ \tfrac{1655215554052763647}{2^{61}}) \cdot x \cdot 2^{-21} \\
&+ \tfrac{1163898039310057983}{2^{61}}) \cdot x \cdot 2^{-20} \\
&+ \tfrac{1376654952773268479}{2^{61}}) \cdot x \cdot 2^{-29} \\
&+ \tfrac{2797511150711145}{2^{61}}) \cdot x \cdot 2^{-11} \\
&+ \tfrac{1302450636498694655}{2^{61}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{2030123327421736447}{2^{61}}) \cdot x \cdot 2^{-20} \\
&+ \tfrac{1318468464673532927}{2^{61}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{1370051626021183487}{2^{61}}) \cdot x \cdot 2^{-18} \\
&+ \tfrac{2135479334019111935}{2^{61}}) \cdot x \cdot 2^{-27} \\
&+ \tfrac{4334036786774485}{2^{61}}) \cdot x \cdot 2^{-8} \\
&+ 1
\end{aligned}
$$

Figure 2: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over I_1 for NTRU+Sign-972

$$
\begin{aligned}
P_{\text{g1296}}(x) = (((((((((((&\tfrac{576819958923685}{2^{62}} \cdot x \cdot 2^{-20} \\
&+ \tfrac{1748379721332813}{2^{62}}) \cdot x \cdot 2^{-21} \\
&+ \tfrac{1725070838155373}{2^{62}}) \cdot x \cdot 2^{-27} \\
&+ \tfrac{22367349482431}{2^{62}}) \cdot x \cdot 2^{-15} \\
&+ \tfrac{1044931119250887}{2^{62}}) \cdot x \cdot 2^{-30} \\
&+ \tfrac{1302007587143}{2^{62}}) \cdot x \cdot 2^{-15} \\
&+ \tfrac{45561808385529}{2^{62}}) \cdot x \cdot 2^{-15} \\
&+ \tfrac{1328631552034153}{2^{62}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{1937216022562437}{2^{62}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{2118423581129849}{2^{62}}) \cdot x \cdot 2^{-19} \\
&+ \tfrac{1544387552610237}{2^{62}}) \cdot x \cdot 2^{-6} \\
&+ 1
\end{aligned}
$$

Figure 3: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over I_1 for NTRU+Sign-1296

A.2 Rejection Sampling

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

Table 10: Parameters for the constant-time Bernoulli sampler

parameter set	I	III	V
n	648	972	1296
σ	208.32	260.96	309.12
$\hat{c}$	$126167442421/2^{21}$	$197985439981/2^{21}$	$277804691115/2^{21}$
ϵ	$1329119/2^{58}$	$2017081/2^{59}$	$2724241/2^{61}$
ϑ_2	59	60	61
$P_{\text{exp}}^{I_2'}$	P_{r648}	P_{r972}	P_{r1296}

$$
\begin{aligned}
P_{r648}(x) = ((((((((((& \tfrac{465402488218683831}{2^{59}} \cdot x \cdot 2^{-39} \\
& + \tfrac{338388767152840201}{2^{59}}) \cdot x \cdot 2^{-39} \\
& + \tfrac{350878231149770981}{2^{59}}) \cdot x \cdot 2^{-38} \\
& + \tfrac{80232318842108307}{2^{59}}) \cdot x \cdot 2^{-40} \\
& + \tfrac{66435196238638723}{2^{59}}) \cdot x \cdot 2^{-34} \\
& + \tfrac{197974032325117345}{2^{59}}) \cdot x \cdot 2^{-38} \\
& + \tfrac{32774603476967113}{2^{59}}) \cdot x \cdot 2^{-37} \\
& + \tfrac{303846553242195703}{2^{59}}) \cdot x \cdot 2^{-42} \\
& + \tfrac{37726316616175407}{2^{59}}) \cdot x \cdot 2^{-33} \\
& + \tfrac{249823462795688873}{2^{59}}) \cdot x \cdot 2^{-36} \\
& + \tfrac{330865921510395043}{2^{59}}) \cdot x \cdot 2^{-35} \\
& + \tfrac{328648849046703943}{2^{59}}) \cdot x \cdot 2^{-36} \\
& + \tfrac{435262177044040003}{2^{59}}) \cdot x \cdot 2^{-34} \\
& + \tfrac{576460752303423487}{2^{59}})
\end{aligned}
$$

Figure 4: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over $I_2' = (-\hat{c}, 0]$ for NTRU+Sign-648

$$
\begin{aligned}
P_{r972}(x) = ((((((((((&\frac{681177064847054353}{2^{60}} \cdot x \cdot 2^{-42} \\
&+ \frac{777192538885365835}{2^{60}}) \cdot x \cdot 2^{-42} \\
&+ \frac{632294353443556367}{2^{60}}) \cdot x \cdot 2^{-43} \\
&+ \frac{226877812469521901}{2^{60}}) \cdot x \cdot 2^{-40} \\
&+ \frac{589590714562061567}{2^{60}}) \cdot x \cdot 2^{-41} \\
&+ \frac{689255719562105755}{2^{60}}) \cdot x \cdot 2^{-43} \\
&+ \frac{179056183464486601}{2^{60}}) \cdot x \cdot 2^{-43} \\
&+ \frac{40701081089678979}{2^{60}}) \cdot x \cdot 2^{-37} \\
&+ \frac{507523010541386817}{2^{60}}) \cdot x \cdot 2^{-40} \\
&+ \frac{659225952735263505}{2^{60}}) \cdot x \cdot 2^{-44} \\
&+ \frac{42813709696405141}{2^{60}}) \cdot x \cdot 2^{-35} \\
&+ \frac{1067732956293621641}{2^{60}}) \cdot x \cdot 2^{-39} \\
&+ \frac{1109509930775006215}{2^{60}}) \cdot x \cdot 2^{-38} \\
&+ \frac{1152921504606846975}{2^{60}})
\end{aligned}
$$

Figure 5: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over $I_2' = (-\hat{c}, 0]$ for NTRU+Sign-972

$$
\begin{aligned}
P_{r1296}(x) = ((((((((((&\frac{1066845239910447733}{2^{62}} \cdot x \cdot 2^{-42} \\
&+ \frac{1707951595931589967}{2^{62}}) \cdot x \cdot 2^{-43} \\
&+ \frac{974858467919598631}{2^{62}}) \cdot x \cdot 2^{-43} \\
&+ \frac{490816946408019057}{2^{62}}) \cdot x \cdot 2^{-41} \\
&+ \frac{894856524535287207}{2^{62}}) \cdot x \cdot 2^{-41} \\
&+ \frac{1467872713586135197}{2^{62}}) \cdot x \cdot 2^{-42} \\
&+ \frac{1070120251064287363}{2^{62}}) \cdot x \cdot 2^{-43} \\
&+ \frac{341314272545377341}{2^{62}}) \cdot x \cdot 2^{-44} \\
&+ \frac{46655142161771401}{2^{62}}) \cdot x \cdot 2^{-36} \\
&+ \frac{1360514904223865691}{2^{62}}) \cdot x \cdot 2^{-42} \\
&+ \frac{495926257753446185}{2^{62}}) \cdot x \cdot 2^{-39} \\
&+ \frac{1084631351111408987}{2^{62}}) \cdot x \cdot 2^{-39} \\
&+ \frac{1581451743979008053}{2^{62}}) \cdot x \cdot 2^{-38} \\
&+ \frac{2305843009213693951}{2^{62}})
\end{aligned}
$$

Figure 6: Polynomial approximation of $\exp\left(\frac{x}{2\sigma^2}\right)$ over $I_2' = (-\hat{c}, 0]$ for NTRU+Sign-1296