NCC-Sign: A New Lattice-based Signature Scheme using Non-Cyclotomic Polynomials*

Kyung-Ah Shim[1], Jeongsu Kim[1], and Youngjoo An[1]

National Institute for Mathematical Sciences
kashim, jsk2357, hellojoo@nims.re.kr

Abstract. Majority of efficient lattice-based schemes are based on the structured lattices which use power-of-2 cyclotomics by default. Despite advantages for choosing cyclotomic polynomials, there has been some concerns on potential threats. In this document, we propose the first lattice-based signature scheme using non-cyclotomic polynomials to remove the structures available to the attackers that provides stronger security guarantee than cyclotomic counterparts. Our scheme follows the Fiat-Shamir paradigm and combines the Bai-Galbraith scheme with several improvements from previous lattice-based schemes including Dilithium. We also propose a new SampleInBall algorithm using the ring structure to choose the challenge polynomial in signing algorithm using two separate polynomials. Furthermore, we construct its cyclotomic trinomial counterpart expected to be as fast as Dilithium. We provide flexible parameter sets at NIST three security levels.

Keywords: Cyclotomic field · Digital signature · Non-cyclotomic polynomial · RLWE · RSIS · Inert Modulus · Trinomial.

1 Introduction

Majority of efficient lattice-based schemes including NIST Post-Quantum Cryptography (PQC) Standardization Round 4 algorithms [40] are based on the structured lattices using power-of-2 cyclotomics by default. Explicitly, Kyber, Saber, Dilithium, and Falcon use the $2n$-th cyclotomic polynomial $\phi(X) = X^n + 1$ for some n a power of 2, and NTRU KEM use a polynomial $\phi(X) = X^p - 1$, which is related to the p-th cyclotomic polynomial for some p a prime number [9, 15, 19, 42, 23, 28, 29]. They achieve high speeds on several architectures as well as reasonably small signatures and key sizes.

There are advantages for choosing cyclotomic polynomials, but there has been potential threads on about on attacks exploited unnecessary algebraic structures [7, 11]. The attacks exploited some additional structures use the fact that the field $\mathbb{Q}[X]/\phi(X)$ has many subfields for certain $\phi(X)$ [6, 2], some attacks use the fact that a number field $\mathbb{Q}[X]/\phi(X)$ has small Galois group [12], and some

* This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr).

attacks using ring homomorphisms from $\mathbb{Z}_q[X]/\phi(X)$ to some smaller nonzero rings [20, 21, 13]. There is sub-exponential time attack against NTRU assumptions ($\phi(X) = X^p - 1$ for some prime p) with large moduli, which invalidated security guarantees of some FHE schemes [2, 34, 10]. There are polynomial-time quantum attacks broke Soliloquy, the cyclotomic case of Gentry's original fully homomorphic encryption (FHE) at STOC 2009 and the cyclotomic case of the Grag-Gentry-Halevi scheme under plausible assumptions [8].

Although no attacks are known that perform significantly better on the schemes using the structured lattices of cyclotomics, it is still possible that further cryptanalysis will be able to exploit the additional structures. Thus, we need to think of countermeasure of the potential threats. As an opponent of these cyclotomics, there is a lattice-based KEM, NTRU Prime KEM, selected as one of the alternative candidates of NIST PQC Round 3 [1], but there is no such a digital signature counterpart. NTRU Prime KEM uses NTRU Prime field [7] that aimed remove unnecessary structures that have been exploited in the attacks. Suggestions for the NTRU Prime field as follows:

1. Choose $\phi(X)$ as a monic irreducible polynomial with degree p for some prime p whose Galois group is isomorphic to S_p (the largest Galois group possible).
2. Choose a prime q so that $\phi(X)$ is still an irreducible polynomial in $\mathbb{Z}_q[X]$, i.e. $\mathbb{Z}_q[X]/\phi(X)$ becomes a field.

NTRU Prime field uses an irreducible polynomial $\phi(X) = X^p - X - 1$ to satisfy the first condition, and the second condition was satisfied with probability $1/p$ for a random prime modulus q.

The schemes based on unstructured lattices guarantee stronger security than those based on the structured lattices, but they suffer from much larger key sizes. Our goal is to construct a lattice-based signature scheme that achieves stronger security guarantee than cyclotomic counterparts and better efficiency than unstructured lattice-based schemes.

1.1 Design Rationale, Advantages and Limitations

NCC-Sign means that it supports two types of signature schemes based on the non-cyclotomic polynomial $\phi(X) = X^p - X - 1$ for stronger security and its cylotomic trinomial counterpart $\phi(X) = X^n - X^{n/2} + 1$ for efficiency. Design Rationale, advantages and limitations of the two types of signature schemes as follows.

[**Non-cyclotomic Case**]
To the best of our knowledge, our scheme is the first lattice-based signature scheme using a prime-degree large Galois group inert modulus with $\phi(X) = X^p - X - 1$, which allows us to remove the structures that were the causes of the previous attacks in the cyclotomic cases. We follow the design paradigm of Dilithium based on Bai and Galbraith scheme with public key compression. However, some critical distinctions exist between our scheme and Dilithium: our scheme is based on the RLWE and RSIS problems using the non-cyclotomic

polynomial instead of the MLWE and MSIS problems using the power-of-2 cyclotomic polynomial. Our choice of the non-cyclotomic polynomial leads to different selection of parameters and different implementation techniques. We also propose a new optimized SampleInBall to choose the challenge polynomial using two separate polynomials.

Intermediate Security Guarantee. In terms of the potential attacks, the schemes based on non-cyclotomic polynomials are more confidence than their ring and module counterparts. NTRU Prime KEM [7, 11] presented evidences that non-cyclotomic scheme had lower risks against the related classical and quantum attacks than the cyclotomic counterparts. Our construction provides intermediate security guarantees between unstructured lattices and cyclotomic structured lattices against the potential threats.

A New Optimized SampleInBall. We propose a new optimized SampleInBall to choose the challenge polynomial in signing and verification algorithms using two different polynomials. This algorithm offers speed-up ranging from 9% to 24% in the rejection sampling phase, depending on the parameter sets.

Security Proofs in ROM and QROM. Existential unforgeability of our scheme is proved in (Q)ROM under the RLWE, RSIS and SelfTargetRSIS assumptions in a similar way to Dilithium [19, 42].

Flexible Choice of Parameters. In the RLWE and MLWE-based schemes over the power-of-2 cyclotomic ring, the degree of polynomials must jump in increasingly by doubling or 256, respectively. Our scheme provides the flexibility for the parameter selections without the jumps that appear in the schemes. We provide flexible three parameter sets: in our second and third parameter sets, the classical Core-SVP estimates exceed the required bit-security at NIST three security levels. Particularly, the expected number of repetitions in the rejection samplings are 1.58, 1.74 and 1.98 at the three security levels in the third parameter set, respectively.

Protection against Side-Channel Attacks. The Fiat-Shamir with Aborts type signatures opt to sample their error vectors from a Gaussian distribution and use rejection sampling to hide the information about the secret-key in the signature. Most of the side channel analysis targeted the data dependent side-channel leakage from these Gaussian sampling, the rejection sampling components and the computation of NTT. Our scheme uses uniform distribution and does not use the NTT for polynomial multiplications which eliminate the causes of the related side-channel attacks.

Our scheme using non-cyclotomic rings ensures stronger security guarantee against the potential threats, but is inefficient since we cannot use optimized implementation techniques for the cyclotomic power-of-2 rings. To overcome this problem, we propose its cyclotomic trinomial counterpart as fast as Dilithium.

[Cyclotomic Trinomial Counterpart]
Our cyclotomic trinomial counterpart uses a $3n$-th cyclotomic trinomial $\phi(X) = X^n - X^{n/2} + 1$, where $n = 2^a \cdot 3^b$.

RLWE-based Signature Scheme. There exist attacks on module lattices: BKZ algorithm for modules [38] which does not outperform the BKZ algorithm, but the algorithm uses only modules and algorithm for the SVP in rank-2 modules [33] which needs an oracle solving the CVP in a fixed lattice of dimension n^2. These algorithms does not outperform the known algorithm on ideal lattices yet, but could be a potential threat on the module lattices. This is why we don't choose the module lattices.

Flexible Choice of Parameters. In the RLWE-based signature scheme using the cyclotomic power-of-2 rings, the degree of polynomials must jump in increasingly by doubling. For the construction of a RLWE-based signature scheme, we choose a trinomial $\phi(X) = X^n - X^{n/2} + 1$ ($n = 2^a \cdot 3^b$) to avoid such parameter jump. We provide two types of parameter sets: one uses a modulus q as a power-of-two, the other is chosen to the use of NTT.

Efficient implementation. In the parameters with a power-of-2 modulus that do not require explicit modular reduction, we use polynomial multiplication algorithms such as Toom-Cook and Karatsuba. Our scheme using incomplete NTT over the NTT-friendly ring $\mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ is expected to be as fast as or faster than Dilithium.

1.2 Related Works

The earlier lattice-based signatures, the GGH scheme [25] and NTRUSign [27], were completely broken by Nguyen and Regev [39] from the leakage of some secret information in lattice trapdoors. To prevent such leakage, Gentry, Peikert, and Vaikuntanathan [24] proposed a hash-and-sign type scheme secure under the hardness of worst-case lattice problems. At Eurocrypt 2012, Lyubashevsky [35] constructed a Fiat-Shamir aborts type signatures based on the LWE and SIS problems with a security reduction to the worst-case problems in general lattices. Subsequently, Güneysu *et al.* [26] proposed a compression technique without requiring Gaussian sampling based on the DCK and RSIS problem and Bai and Galbraith (BG) [4] introduced an improved compression technique for signature schemes based on the LWE problem.

Many lattice-based schemes base on the BG scheme have been proposed qTESLA [3] based on RLWE and RSIS problems, Dilithium [19, 42] based on MLWE and MSIS problems, MLWRSign [32] based on MLWR problem as particular instantiations of the BG framework. The Hash-and-Sign type schemes are Falcon [23] based on NTRU problem, its variant Mitaka [22] and ModFalcon [14] based on Module-NTRU problem. Recently, NIST recommended Dilithium and FALCON as digital signatures of NIST PQC Standardization [40].

2 Signature Scheme: NCC-Sign

2.1 Basic Operations

Throughout this document, we let $R := \mathbb{Z}[X]/(X^p - X - 1)$ and $R_q := \mathbb{Z}_q[X]/(X^p - X - 1)$ for some prime numbers p and q such that R_q is a field. Boldface lower-case letters represent elements in R or R_q, and non-boldface lower-case letters represent elements in $\mathbb{Z}$ and $\mathbb{Z}_q$.

Modular Reductions. For an integer α, we let $r' = r \mod^{\pm} \alpha$ to be the unique integer $r' \in (-\alpha/2, \alpha/2]$ such that $r' \equiv r \mod \alpha$. Similarly, we let $r' = r \mod^{+} \alpha$ to be the unique integer $r' \in [0, \alpha)$. For an element $\mathbf{r} = r_0 + r_1 X + \ldots r_{p-1} X^{p-1} \in R$, we let $\mathbf{r'} = \mathbf{r} \mod^{\pm} \alpha$ (resp. $\mathbf{r'} = \mathbf{r} \mod^{+} \alpha$) to be the unique element in R such that $\mathbf{r'} = r'_0 + r'_1 X + \ldots r'_{p-1} X^{p-1}$ and $r'_i = r_i \mod^{\pm} \alpha$ (resp. $r'_i = r_i \mod^{+} \alpha$) for all i. When we do not require exact representation, we write $r \mod \alpha$ or $\mathbf{r} \mod \alpha$.

Sizes of elements. For $w \in \mathbb{Z}_q$, let $\|w\|_\infty := \left|w \mod^{\pm} q\right|$. We also define l_∞ and l_2 norm of $\mathbf{w} = w_0 + w_1 X + \cdots + w_{p-1} X^{p-1} \in R$ as

$$\|\mathbf{w}\|_\infty := \max_i \|w_i\|_\infty, \ \|\mathbf{w}\|_2 := \sqrt{\|w_0\|_\infty^2 + \cdots + \|w_{p-1}\|_\infty^2},$$

respectively. We write S_η to denote the set of elements $\mathbf{w} \in R$ that satisfy $\|\mathbf{w}\|_\infty \leq \eta$. We let $\tilde{S}_\eta := \{\mathbf{w} \mod^{\pm} 2\eta \ : \ \mathbf{w} \in R\}$. One can see that $\tilde{S}_\eta \subset S_\eta$, but $\tilde{S}_\eta$ does not include the elements with at least one $-\eta$ coefficient.

A New SampleInBall Algorithm. We use multiple hashing algorithms that map strings in $\{0,1\}^*$ to random elements in desired domains such as S_η and R_q. SampleInBall algorithm maps a random seed $\rho \in \{0,1\}^{256}$ to an element of B_τ, the subset of S_1 consists of elements that have total τ nonzero coefficients in $\{-1, 0, 1\}$. We propose a new SampleInBall algorithm using our ring structure as follows: the challenge polynomial can be chosen in the following two ways

- choose a single polynomial $\mathbf{c} \in R$ having τ non-zero coefficients,
- choose two polynomials $\mathbf{c}_i \in R$ having τ_i non-zero coefficients for $i = 1, 2$ and combine them such that $\mathbf{c} = \mathbf{c}_2 + X^{p_2} \mathbf{c}_1$. Note that $\mathbf{c}_i$ is a degree-$(p_i - 1)$ polynomial.

It is enough to specify the method of choosing polynomial having fixed number of non-zero coefficients. Basically, we follow [19, 42]. High-level description is described in Algorithm 1. More specifically, Step 3 and 4 in Algorithm 1 can be done in the following way from the 256-bit hash seed ρ. We use SHAKE-256 to obtain a stream of random bytes of variable length from the seed ρ. The first τ bits in the first 8 bytes of this random stream are τ random sign bits $s_i \in \{0, 1\}$, $i = 0, \ldots, \tau - 1$, required in Step 4. The remaining $64 - \tau$ bits are discarded. For the random j required in Step 3, we use next 10 or 11 bits from the next two bytes in the stream and interpret it as a single number less than 2^{10} or 2^{11} depending on p. When this number is less than or equal to i, we use it as j.

If not, we use next two bytes in the stream to choose j. Lastly, for the case of two polynomials, we use another SHAKE-256 to obtain 512-bits from the seed ρ. Then the first 256-bits are used as a seed for $\mathbf{c}_1$ while the second 256-bits are used as a seed for $\mathbf{c}_2$. From the seeds, the needed randomness can be extracted as is described in Algorithm 1. We will analyze the probability of the rejection in Sign algorithm using the proposed SampleInBall in §3.3.

Algorithm 1: $\mathsf{SampleInBall}_{p,\tau}(\rho)$.
Create a random p-element array with τ ± 1's and $p-\tau$ 0's.
Use the input seed ρ (and an XOF) to generate the randomness needed in Step 3 and 4.

1 Initialize $\mathbf{c} = c_0 c_1 \dots c_{p-1} = 00\dots0$
2 **for** $i := p-\tau$ *to* $p-1$ **do**
3 $\quad j \leftarrow \{0, 1, \dots, i\}$
4 $\quad s \leftarrow \{0, 1\}$
5 $\quad c_i := c_j$
6 $\quad c_j := (-1)^s$
7 **return c**

Algorithm 2: $\mathsf{Decompose}_q(r, \alpha)$

1 $r := r \mod^+ q$
2 $r_0 := r \mod^\pm \alpha$
3 **if** $r - r_0 = q - 1$ **then**
4 $\quad r_1 := 0$
5 $\quad r_0 := r_0 - 1$
6 **else**
7 $\quad r_1 := (r - r_0)/\alpha$
8 **return** (r_1, r_0)

Algorithm 3: $\mathsf{UseHint}_q(h, r, \alpha)$

1 $m := (q-1)/\alpha$
2 $(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$
3 **if** $h = 1$ *and* $r_0 > 0$ **then**
4 $\quad$ **return** $(r_1 + 1) \mod^+ m$
5 **if** $h = 1$ *and* $r_0 \le 0$ **then**
6 $\quad$ **return** $(r_1 - 1) \mod^+ m$
7 **return** r_1

Algorithm 4: $\mathsf{Power2Round}_q(r, d)$

1 $r := r \mod^+ q$
2 $r_0 := r \mod^\pm 2^d$
3 **return** $\big((r - r_0)/2^d, r_0\big)$

Algorithm 5: $\mathsf{HighBits}_q(r, \alpha)$

1 $(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$
2 **return** r_1

Algorithm 6: $\mathsf{LowBits}_q(r, \alpha)$

1 $(r_1, r_0) := \mathsf{Decompose}_q(r, \alpha)$
2 **return** r_0

Algorithm 7: $\mathsf{MakeHint}_q(z, r, \alpha)$

1 $r_1 := \mathsf{HighBits}_q(r, \alpha)$
2 $v_1 := \mathsf{HighBits}_q(r + z, \alpha)$
3 **return** $[\![r_1 \neq v_1]\!]$

High/Low Order Bits and Hints. We use several algorithms, Algorithm 2-7, that extract higher/lower bits of an input, and the other algorithms that help to correctly produce higher bits of a summation $r + z \in \mathbb{Z}_q$ when $r \in \mathbb{Z}_q$ and $z \in \mathbb{Z}_q$ is small. The algorithms can be extended to use inputs in R_q (except for d and α) by applying the algorithm to each coefficient.

Other Functions. ExpandA, ExpandS and ExpandMask maps random seeds to $\mathbf{a} \in R_q$, $(\mathbf{s}_1, \mathbf{s}_2) \in S_\eta \times S_\eta$ and $\mathbf{y} \in \tilde{S}_\eta$, respectively. We instantiate function H as the extendable-output function (XOF) SHAKE-256.

2.2 Specification of NCC-Sign

We give KeyGen, Sign and Verify, of NCC-Sign in Algorithm 8, 9, and 10, respectively.

Algorithm 8: KeyGen

$(\zeta, \zeta') \leftarrow \{0,1\}^{256} \times \{0,1\}^{256}$
$(\xi_1, \xi_2, K) \in \{0,1\}^{256} \times \{0,1\}^{256} \times \{0,1\}^{256} := \mathrm{H}(\zeta')$
$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$
$(\mathbf{s}_1, \mathbf{s}_2) \in S_\eta \times S_\eta := \mathsf{ExpandS}(\xi_1, \xi_2)$
$\mathbf{t} := \mathbf{a}\mathbf{s}_1 + \mathbf{s}_2$
$(\mathbf{t}_1, \mathbf{t}_0) := \mathsf{Power2Round}_q(\mathbf{t}, d)$
$ph \in \{0,1\}^{256} := \mathrm{H}(\zeta \parallel \mathbf{t}_1)$
return $(pk = (\zeta, \mathbf{t}_1), sk = (\zeta, ph, dK, \mathbf{s}_1, \mathbf{s}_2, \mathbf{t}_0))$

Algorithm 9: $\mathsf{Sign}(sk, M)$

$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$
$\mu \in \{0,1\}^{512} := \mathrm{H}(ph \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho \in \{0,1\}^{512} := \mathrm{H}(dK \parallel \mu)$ (or $\rho \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
    $\mathbf{y} \in \tilde{S}_{\gamma_1} := \mathsf{ExpandMask}(\rho, \kappa)$
    $\mathbf{w} := \mathbf{a}\mathbf{y}$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathrm{H}(\mu \parallel \mathbf{w}_1)$
    $\mathbf{c} \in B_\tau := \mathsf{SampleInBall}_{p,\tau}(\tilde{c})$
    $\mathbf{z} := \mathbf{y} + \mathbf{c}\mathbf{s}_1$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)$
    if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ **or** $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ **then**
        $(\mathbf{z}, \mathbf{h}) := \perp$
    else
        $\mathbf{h} := \mathsf{MakeHint}_q(-\mathbf{c}\mathbf{t}_0, \mathbf{w} - \mathbf{c}\mathbf{s}_2 + \mathbf{c}\mathbf{t}_0, 2\gamma_2)$
        if $\|\mathbf{c}\mathbf{t}_0\|_\infty \geq \gamma_2$ **or** the # of 1's in $\mathbf{h}$ is greater than ω **then**
            $(\mathbf{z}, \mathbf{h}) := \perp$
        $\kappa := \kappa + 1$
return $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$

Algorithm 10: $\mathsf{Verify}(pk, M, \sigma) = (\tilde{c}, \mathbf{z}, \mathbf{h})$

$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$
$\mu \in \{0,1\}^{512} := \mathrm{H}(\mathrm{H}(\zeta \parallel \mathbf{t}_1) \parallel M)$
$\mathbf{c} := \mathsf{SampleInBall}(\tilde{c})$
$\mathbf{w}_1' := \mathsf{UseHint}_q(\mathbf{h}, \mathbf{az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2)$
return $[\![\|\mathbf{z}\|_\infty < \gamma_1 - \beta]\!]$ **and** $[\![\tilde{c} = \mathrm{H}(\mu \parallel \mathbf{w}_1')]\!]$ **and**
$[\![\#$ of 1's in $\mathbf{h}$ is $\leq \omega]\!]$

We offer both deterministic and randomized versions of the algorithm Sign. For randomized version, the procedure for generating ρ is replaced by random sampling from $\{0,1\}^{512}$, whereas deterministic version uses collision-resistant hash function to digest a message M into μ using the hash value of the public key ph, then uses a secret key dK and μ as an input of H to safely generate ρ. We use two separate seeds, ζ and ζ', to generate a public key $\mathbf{a}$ and a secret key $(\mathbf{s}_1, \mathbf{s}_2, K)$, respectively, not to exclude the case of sharing the public key $\mathbf{a}$.

2.3 Correctness

For the correctness and security analysis of our scheme, we need the following lemmas. Since the Lemmas are similar to Lemma 1, 2 and 3 in [42, 19], we omit their proofs.

Lemma 1 ([19, 42]). *Suppose that q and α are positive integers satisfying $q > 2\alpha$, $q \equiv 1 \pmod{\alpha}$ and α even. Let $\mathbf{r}$ and $\mathbf{z}$ be elements of R_q where $\|\mathbf{z}\|_\infty \leq \alpha/2$, and let $\mathbf{h}$, $\mathbf{h}'$ be vectors of bits (polynomials in R_q where coefficients are 0 or 1). Then the $\mathsf{HighBits}_q$, $\mathsf{MakeHint}_q$, and $\mathsf{UseHint}_q$ algorithms satisfy the following properties:*

1. $\mathsf{UseHint}_q(\mathsf{MakeHint}_q(\mathbf{z}, \mathbf{r}, \alpha), \mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \alpha)$.
2. *Let* $\mathbf{v}_1 = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha)$. *Then* $\|\mathbf{r} - \mathbf{v}_1 \cdot \alpha\|_\infty \leq \alpha + 1$. *Furthermore, if the number of 1's in* $\mathbf{h}$ *is* ω, *then all except at most* ω *coefficients of* $\mathbf{r} - \mathbf{v}_1 \cdot \alpha$ *will have magnitude of at most* $\alpha/2$ *after centered reduction modulo* q.
3. *For any* $\mathbf{h}$, $\mathbf{h}'$, *if* $\mathsf{UseHint}_q(\mathbf{h}, \mathbf{r}, \alpha) = \mathsf{UseHint}_q(\mathbf{h}', \mathbf{r}, \alpha)$, *then* $\mathbf{h} = \mathbf{h}'$

Lemma 2 ([19, 42]). *If $\|\mathbf{s}\|_\infty \leq \beta$ and $\|\mathsf{LowBits}_q(\mathbf{r}, \alpha)\|_\infty < \alpha/2 - \beta$, then*

$$\mathsf{HighBits}_q(\mathbf{r}, \alpha) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{s}, \alpha).$$

Suppose a signature $\sigma = (\tilde{c}, \mathbf{z}, \mathbf{h})$ is generated from a signing algorithm. We first note that

$$\begin{aligned}
&\mathbf{az} - \mathbf{ct}_1 \cdot 2^d = \mathbf{a}(\mathbf{y} + \mathbf{cs}_1) - \mathbf{ct}_1 \cdot 2^d = \mathbf{ay} + \mathbf{cas}_1 - \mathbf{ct}_1 \cdot 2^d \\
=&\mathbf{ay} + \mathbf{cas}_1 - \mathbf{c}(\mathbf{t}_1 \cdot 2^d + \mathbf{t}_0) + \mathbf{ct}_0 = \mathbf{ay} + \mathbf{cas}_1 - \mathbf{ct} + \mathbf{ct}_0 \\
=&\mathbf{ay} + \mathbf{cas}_1 - \mathbf{c}(\mathbf{as}_1 + \mathbf{s}_2) + \mathbf{ct}_0 = \mathbf{ay} - \mathbf{cs}_2 + \mathbf{ct}_0.
\end{aligned}$$

Therefore, if $(\mathbf{z}, \mathbf{h}) \neq \perp$, then

$$\mathbf{w}_1' = \mathsf{UseHint}_q(\mathbf{h}, \mathbf{az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{ay} - \mathbf{cs}_2, 2\gamma_2)$$

from Lemma 1 and the fact that $\|\mathbf{ct}_0\|_\infty < \gamma_2$. As we are setting β to satisfy that $\|\mathbf{cs}_2\|_\infty \leq \beta$ and the signing algorithm makes sure that $\|\mathsf{LowBits}_q(\mathbf{ay} - \mathbf{cs}_2)\| < \gamma_1 - \beta$ for $(\mathbf{z}, \mathbf{h}) \neq \perp$, from Lemma 2, we get

$$\mathbf{w}_1' = \mathsf{HighBits}_q(\mathbf{ay} - \mathbf{cs}_2, 2\gamma_2) = \mathsf{HighBits}_q(\mathbf{ay}, 2\gamma_2) = \mathbf{w}_1,$$

which shows the correctness of our scheme.

3 Security and Parameter Selections

Now, we prove unforgeability of our scheme in QROM under the hardness assumptions of RLWE, RSIS and SelfTargetRSIS problems. We then select concrete and conservative parameters at three NIST security levels based on the security proofs and cost analysis against the lattice attacks on known cost models.

3.1 Existential Unforgeability

We adapt the security proof of Dilithium [19, 42] to our case: $l = k = 1$ and $R = \mathbb{Z}[X]/(X^p - X - 1)$. We follow the proof in [19, 42] and slightly change the bound due to our choice of ring R. The ring R_q is a ring R/qR where q is an inert prime over R which means both $R = \mathbb{Z}[X]/(X^p - X - 1)$ and $R_q = \mathbb{Z}_q[X]/(X^p - X - 1)$ are fields. Note that χ is a noise distribution. We let H to be a random oracle that maps its input to an element in B_τ. We use the following hardness assumptions.

Definition 1 ($\mathsf{Ring\text{-}LWE}_{q,\chi}$ **Problem).** *Let q be a positive integer. For a probability distribution χ over R_q, sample $\mathbf{a} \xleftarrow{\$} R_q$ and a vector $\mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi$, and output $(\mathbf{a}, \mathbf{as}_1 + \mathbf{s}_2)$.*

Definition 2 (Decision $\mathsf{Ring\text{-}LWE}_{q,\chi}$ **Problem).** *Given a pair $(\mathbf{a}, \mathbf{t})$ decide, with non-negligible advantage, whether it came from the RLWE distribution or it was generated uniformly at random from $R_q \times R_q$. The advantage of the adversary $\mathcal{A}$ in solving decisional RLWE problem over the ring R_q is*

$$\begin{aligned}\mathrm{Adv}_{q,\chi}^{\mathsf{Ring\text{-}LWE}}(\mathcal{A}) := \Big| & Pr[b = 1 \,|\, \mathbf{a}, \mathbf{t} \xleftarrow{\$} R_q; b \leftarrow \mathcal{A}(\mathbf{a}, \mathbf{t}) \\ & - Pr[b = 1 \,|\, \mathbf{a} \xleftarrow{\$} R_q, \mathbf{s}_1, \mathbf{s}_2 \leftarrow \chi; b \leftarrow \mathcal{A}(\mathbf{a}, \mathbf{as}_1 + \mathbf{s}_2)]\Big|.\end{aligned}$$

We say RLWE is hard when the above advantage is negligible for all (quantum) probabilistic polynomial-time algorithm $\mathcal{A}$.

Definition 3 ($\mathsf{Ring\text{-}SIS}_{q,l,\gamma}$ **Problem).** *The advantage of the adversary $\mathcal{A}$ to solve RSIS problem over the ring R_q is*

$$\mathrm{Adv}_{l,\gamma}^{\mathsf{Ring\text{-}SIS}}(\mathcal{A}) := \Pr\left[0 < \|\mathbf{y}\|_\infty \leq \gamma \wedge \begin{bmatrix}\mathbf{a}_1 \dots \mathbf{a}_l\ 1\end{bmatrix} \cdot \mathbf{y} = 0 \,\middle|\, \mathbf{a}_1 \dots, \mathbf{a}_l \xleftarrow{\$} R_q; \mathbf{y} \leftarrow \mathcal{A}(\mathbf{a}_1, \dots, \mathbf{a}_l)\right].$$

Definition 4 ($\mathsf{SelfTargetRSIS}_{q,\gamma,\mathrm{H}}$ **Problem).** *For the cryptographic hash function H, the advantage of $\mathcal{A}$ to solve SelfTargetRSIS problem $Adv_{H,\gamma}^{\mathsf{SelfTargetRSIS}}(\mathcal{A})$ is defined as*

$$\Pr\left[\begin{array}{c} 0 \leq \|\mathbf{y}\|_\infty \leq \gamma \wedge \\ H(\mu \| \begin{bmatrix}\mathbf{a}_1\ \mathbf{a}_2\ 1\end{bmatrix} \cdot \mathbf{y}) = \mathbf{c} \end{array} \,\middle|\, \mathbf{a}_1, \mathbf{a}_2 \xleftarrow{\$} R_q; \left(\mathbf{y} := \begin{bmatrix}\mathbf{r}_1 \\ \mathbf{c} \\ \mathbf{r}_2\end{bmatrix}, \mu\right) \leftarrow \mathcal{A}^{|H(\cdot)\rangle}(\mathbf{a}_1, \mathbf{a}_2) \right].$$

We note that there is a classical reduction from RSIS to SelfTargetRSIS [19, 42].

Sketch of Security Proofs. We assume that a public key is given without the compression. Proving security in this case also shows the security when compression is used. In [5], the authors showed that, for existential unforgeability against chosen-message attacks (UF-CMA), existential unforgeability against no-message attacks (UF-NMA) is sufficient if the underlying identification scheme is accepting honest-verifier zero-knowledge (acHVZK): in the identification scheme, *accepting* transcript can be simulated without the secret information ([5] and [16] are concurrent works that fixed the original security proofs of Dilithium and Lyubashevsky [31, 35]). Therefore, we show that our scheme achieves acHVZK and UF-NMA in (Q)ROM.

UF-NMA security. In order to prove UF-NMA of our scheme based on RLWE and SelfTargetRSIS assumptions, firstly using RLWE assumption, we replace the public key by random elements of R_q, $(\mathbf{a}, \mathbf{t})$. Then, the adversary $\mathcal{A}$ receives $(\mathbf{a}, \mathbf{t})$ and needs to output valid message/signature pair M and $(\mathbf{z}, \mathbf{h}, \mathbf{c})$ such that

$$\|\mathbf{z}\|_\infty < \gamma_1 - \beta,\ \mathrm{H}(\mu \| \mathsf{UseHint}_q(\mathbf{h}, \mathbf{az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2)) = \mathbf{c},$$

and the number of 1's in $\mathbf{h}$ is less than ω. Lemma 1 implies

$$2\gamma_2 \cdot \mathsf{UseHint}_q(\mathbf{h}, \mathbf{az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2) = \mathbf{az} - \mathbf{ct}_1 \cdot 2^d + \mathbf{v},$$

where $\|\mathbf{v}\|_\infty \leq 2\gamma_2 + 1$. Let $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + t_0$ where $\|\mathbf{t}_0\|_\infty \leq 2^{d-1}$. Then

$$\mathbf{az} - \mathbf{ct}_1 \cdot 2^d + \mathbf{v} = \mathbf{az} - \mathbf{c}(\mathbf{t} - \mathbf{t}_0) + \mathbf{v} = \mathbf{az} - \mathbf{ct} + (\mathbf{ct}_0 + \mathbf{v}) = \mathbf{az} - \mathbf{ct} + \mathbf{v}',$$

where $\|\mathbf{v}'\|_\infty \leq 2\tau 2^{d-1} + 2\gamma_2 + 1$. It follows that using adversary, we find $\mathbf{z}, \mathbf{c}, \mathbf{v}', M$ such that $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$, $\|\mathbf{c}\|_\infty = 1$, $\|\mathbf{v}'\|_\infty \leq 2\tau \cdot 2^{d-1} + 2\gamma_2 + 1$, $M \in \{0,1\}^*$, such that

$$\mathrm{H}(\mu \| \frac{1}{2\gamma_2} \begin{bmatrix}\mathbf{a} - \mathbf{t}\ 1\end{bmatrix} \cdot \begin{bmatrix}\mathbf{z} \\ \mathbf{c} \\ \mathbf{v}'\end{bmatrix}) = \mathbf{c}.$$

Let $\mathrm{H}(\mu\|\mathbf{x}) = \mathrm{H}'(\mu\|2\gamma_2 \cdot \mathbf{x})$. Then $\mathrm{H}'(\mu\| \begin{bmatrix}\mathbf{a} & -\mathbf{t} & 1\end{bmatrix} \cdot \begin{bmatrix}\mathbf{z} \\ \mathbf{c} \\ \mathbf{v}'\end{bmatrix}) = \mathbf{c}$ and this solves the SelfTargetRSIS problem with $\gamma = \max\{\gamma_1 - \beta, 2\tau \cdot 2^{d-1} + 2\gamma_2 + 1\}$.

Zero-knowledgeness. Now we prove that our scheme is acHVZK. Assume that public key is $\mathbf{t}$ (rather than $\mathbf{t}_1$). We note that $\mathbf{t}_0$ is used in simulation. It is clear that if our scheme is zero-knowledge with $\mathbf{t}$ then it is zero-knowledge with $\mathbf{t}_1$. Let $\mathbf{w} = \mathbf{ay}$ and $\mathbf{z} = \mathbf{y} + \mathbf{cs}_1$. Then $\mathbf{w} - \mathbf{cs}_2 = \mathbf{ay} - \mathbf{cs}_2 = \mathbf{az} - \mathbf{ct}$ since

$$\mathbf{az} - \mathbf{ct} = \mathbf{a}(\mathbf{y} + \mathbf{cs}_1) - \mathbf{ct} = \mathbf{ay} + \mathbf{acs}_1 - \mathbf{ct} = \mathbf{ay} - \mathbf{c}(\mathbf{t} - \mathbf{as}_1) = \mathbf{w} - \mathbf{cs}_2.$$

Now, $\Pr[\mathbf{z}, \mathbf{c}] = \Pr[\mathbf{c}] \Pr[\mathbf{y} = \mathbf{z} - \mathbf{cs}_1 \mid \mathbf{c}]$ where $\|\mathbf{z}\|_\infty \leq \gamma_1 - \beta$. If $\|\mathbf{cs}_i\|_\infty \leq \beta$, then $\|\mathbf{z} - \mathbf{cs}_i\|_\infty \leq \gamma_1 - 1$. Since $\mathbf{y}$ is chosen uniformly random from $\tilde{S}_{\gamma_1}$, the probability is the same for all $(\mathbf{z}, \mathbf{c})$. For the simulation, we pick uniformly random

$$(\mathbf{z}, \mathbf{c}) \in S_{\gamma_1 - \beta - 1} \times B_\tau$$

and check $\|\mathbf{r}_0\|_\infty = \|\mathsf{LowBits}_q(\mathbf{w} - \mathbf{cs}_2, 2\gamma_2)\|_\infty = \|\mathsf{LowBits}_q(\mathbf{az} - \mathbf{ct}, 2\gamma_2)\|_\infty \leq \gamma_2 - \beta$. Since $\mathbf{h}$ can be constructed when $(\mathbf{z}, \mathbf{c})$ is sampled, and such simulation's output is indistinguishable from the honestly generated *accepting* transcript, our underlying identification scheme is acHVZK.

3.2 Security Estimates for RLWE and RSIS

We follow the core-SVP method: BKZ-b calls the SVP oracle of dimension b which costs in time $\approx 2^{0.292b}$. For a given basis $(\mathbf{c}_1, ..., \mathbf{c}_n)$ as input, $\mathbf{c}_k(i)$ is a projection of $\mathbf{c}_k$ orthogonally to the vectors $(\mathbf{c}_1, ..., \mathbf{c}_i)$, let $\ell_i = \log_2 \|\mathbf{c}_i(i-1)\|$. BKZ preserves the determinant of the $\mathbf{c}_i$'s, and the sum of the ℓ_is remains constant. After small number of SVP calls inside the BKZ algorithm, we expect the local slope of the ℓ_is converges to

$$\mathrm{slope}(b) = \frac{1}{b-1} \log_2 \left(\frac{b}{2\pi e} (\pi \cdot b)^{1/b} \right).$$

After the BKZ reduction, ℓ_is are of the following forms:

- The first ℓ_is are constant equal to $\log_2 q$ (possibly empty).
- Then they decrease linearly, with slope $\mathrm{slope}(b)$.
- The last ℓ_is are constant equal to 0 (possibly empty).

Throughout this section, we write $\mathrm{vec}(\mathbf{x}) = [x_0, x_1, \cdots, x_{p-1}]^T$ when $\mathbf{x} = x_0 + x_1 X + \ldots x_{p-1} X^{p-1} \in R_q$, and $\mathrm{rot}(\mathbf{x})$ is a matrix whose k-th column vector is $\mathrm{vec}(X^{k-1} \cdot \mathbf{x})$. Also, $\mathrm{rot}(\mathbf{x})_{[1:m]}$ is a $m \times p$ matrix consisting of first m rows of a matrix $\mathrm{rot}(\mathbf{x})$.

Solving RLWE. Any RLWE instance over R can be viewed as a LWE instance. Let $(\mathbf{a}, \mathbf{b}) \in R_q^2$ be a RLWE instance over R_q, where $\mathbf{b} = \mathbf{a} \cdot \mathbf{s}_1 + \mathbf{s}_2$.

Main lattice attack is a primal attack which finds short vectors in the following lattice L of dimension $d = p + m + 1$ and determinant q^m which has the solution vector $(\mathrm{vec}(\mathbf{s}_2), \mathrm{vec}(\mathbf{s}_1), 1)$: $L = \begin{bmatrix} qI_m & -\mathrm{rot}(\mathbf{a})_{[1:m]} & \mathbf{b} \\ & I_p & 0 \\ & & 1 \end{bmatrix}$. It is known that one can expect to find the solution if $2^{\ell_{d-b}}$ is greater than the expected norm of $(\mathrm{vec}(\mathbf{s}_2), \mathrm{vec}(\mathbf{s}_1), 1)$ after projection orthogonally to the first $d - b$ vectors, which is $\varsigma\sqrt{b}$, where ς is a standard deviation of coordinates of $\mathbf{s}_1, \mathbf{s}_2$. When it is uniform on $[-1, 0, 1]$, it is $\sqrt{2/3} \approx 0.816$. For $[-2, -1, 0, 1, 2]$, it is about 1.414 and for $[-4, -3, -2, -1, 0, 1, 2, 3, 4]$, it is about 2.582. We also assume that the number of SVP calls inside BKZ is larger than d which equals to $p + m + 1$.

Solving RSIS and SelfTargetRSIS. For the RSIS and SelfTargetRSIS problem, we consider those problems as a RSIS problem. For the RSIS problem, given uniformly sampled polynomials $\mathbf{a}_i \in R_q$, $i = 1, ..., k$, it is required to find small polynomials $\mathbf{y}_i$, $i = 0, ..., k$, s.t. $\mathbf{y}_0 + \sum_{i=1}^{k} \mathbf{y}_i \mathbf{a}_i = 0$ and $\|\mathbf{y}_i\|_\infty \leq \gamma$. Using rotation matrix, the RSIS problem can be solved by lattice reduction algorithms finding short vectors in the following lattice basis of determinant q^p which has the solution vector $(-\mathrm{vec}(\mathbf{y}_0),\ \mathrm{vec}(\mathbf{y}_1),\ \cdots,\ \mathrm{vec}(\mathbf{y}_k))$:

$$L = \begin{bmatrix} qI_p & \mathrm{rot}(\mathbf{a}_1) & \cdots & \mathrm{rot}(\mathbf{a}_k) \\ & I & & \\ & & \ddots & \\ & & & I \end{bmatrix}.$$

To find the solution vector of the lattice, one uses the BKZ algorithm of block size b after choosing w columns among rotated vectors to obtain a lattice of dimension $d = w + p$. As is explained above, after the BKZ algorithm, one can obtain ℓ_is. Let i be the smallest index such that ℓ_i is below $\log_2 q$ and j be the largest index such that ℓ_j is above 0. Then, from the BKZ algorithm, one obtains $\sqrt{4/3}^b$ short vectors of length 2^{ℓ_i} after projection to the first $i - 1$ vectors. Now we assume that our short vectors have coordinates that satisfy the followings:

- the first $i - 1$ coordinates are uniform modulo q.
- the next $j - i + 1$ coordinates have similar magnitude and sampled from Gaussian distribution of standard deviation σ where $\sigma = 2^{\ell_i}/\sqrt{j - i + 1}$.
- the last $w - j$ coordinates are zeroes.

If those j coordinates are all have absolute values less than γ, then the vector is considered as a solution vector. Time complexity of the algorithm finding a SIS solution is the cost of BKZ-b multiplied by the inverse of the success probability of finding such vectors within the $\sqrt{4/3}^b$ vectors. Similar to Dilithium, we also consider the forget q case. In this case, the lattice basis is first multiplied by some random unimodular matrices to remove the first q-vectors. Then the BKZ algorithm is applied and we assume that q-vectors are not found. The above

analysis is applied in the same way to $i = 1$. As in the RLWE case, we assume that the cost of BKZ-b is the cost of SVP_b multiplied by the dimension d.

Other Attacks. There exist other attacks like algebraic attacks. However, we do not consider algebraic attacks since they usually need many samples. Our signature scheme only offer one RLWE sample, which translates to p LWE samples. Since hybrid attacks are especially suitable to sparse secret, we do not consider these attacks.

3.3 Parameter Selection for Non-Cyclotomic Case

Based on the security estimates for RLWE and RSIS, we choose secure parameter sets for our scheme at the three security levels. We first describe how to choose p and q. We can find enough list of candidate inert primes for each prime p, and find suitable primes p and q in the list satisfying $q \equiv 1 \mod 2\gamma_2$. This condition is needed for the correct verification and $q-1$ needs to have small even divisor. In this reason, we choose γ_2 as a $q-1$ divided by suitable even number like 90, 56, 42. The concrete choice depends on the exact value of q and it affects the cost to the SIS problem. Larger γ_2 is good for efficiency but bad for the security. We choose suitable p and q such that the expected number of repetitions in the rejection sampling is not too large for efficiency. In Table 1, we list some inert primes q for a given p.

p	q
1021	8348477, 8339581, 8333113
1429	8380087, 8376649, 8333131, 8332559
1913	8361623, 8343469, 8334383

Table 1: Some inert primes q for a given p

Our parameter choice is different from Dilithium [19, 42] and NTRU Prime KEM [7, 11].

- In NTRU Prime KEM [7, 11], the smallest p is 653 with $q = 4621$, but we need a larger p corresponding to much larger q. The main reason for this difference comes from the rejection sampling required in the signature scheme, while it is not needed in KEM. The rejection sampling in signing makes the distribution of a signature independent from the secret key. For efficient rejection sampling, the larger q the better: it lowers the rejection probability. With larger q, we need larger p to thwart the lattice attacks.
- The size of q in our scheme is similar to that of Dilithium [19, 42]. While Dilithium uses a single prime q for the modulus at all security levels, our q is different at each security level. This is because we need an inert modulus q for each prime p.

According to our security proof, our scheme is secure as long as the following problems are hard:

- RLWE_D where D is a uniform distribution over S_η
- SelfTargetRSIS with $k = 2, \zeta$ where $\zeta = \max\{\gamma_1 - \beta, 2\gamma_2 + 1 + 2^d \cdot \tau\}$
- RSIS with $k = 1, \zeta'$ where $\zeta' = \max\{2(\gamma_1 - \beta), 4\gamma_2 + 2\}$

Classically, SelfTargetRSIS with ζ can be reduced from RSIS with 2ζ. Thus for the concrete parameters, we consider RSIS with $k = 2, 2\zeta$ instead of the SelfTargetRSIS problem for simplicity. Thus, we consider the following problems for the concrete parameters:

- RLWE_D where D is a uniform distribution over S_η
- RSIS with $k = 2$, $\zeta = \max\{2(\gamma_1 - \beta), 4\gamma_2 + 2 + 2^{d+1} \cdot \tau\}$
- RSIS with $k = 1$, $\zeta' = \max\{2(\gamma_1 - \beta), 4\gamma_2 + 2\}$

After selecting suitable primes p and q, we choose γ_1 as a power of two and γ_2 such that $2\gamma_2 \mid q-1$ and $2\gamma_2 \approx \gamma_1$. We also use $\eta = 2$. Larger η makes the underlying LWE problem harder, at the cost of less efficient rejection sampling since $\beta = 2\tau\eta$ in the expected number of the rejection sampling. The first parameter set and more conservative second parameter set are given in Table 2 and Table 3, respectively. In Table 2 and Table 3, Exp. reps. represents the expected numbers of repetitions in rejection samplings calculated by $e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}$ from the proposed SampleInBall algorithm which will be analyzed in the next subsection.

We provide balanced parameter sets. Since LWE cost is always higher than the SIS cost, we lower the η to balance the security of SIS and LWE. Namely, we use $\eta = 1$ in the parameter sets of Table 7, where one can see that SIS cost is only slightly higher than the LWE cost. In the third balanced parameter set, we use $\eta = 1$ to balance the security of LWE and SIS, which means that the ternary secret and error are used. It is known that hybrid attacks are more effective to the ternary secret case. In this reason, we review the security of the balanced parameter sets against the hybrid attacks. We use the code published in `https://github.com/bencrts/hybrid_attacks`, which uses hybrid-decoding and hybrid-dual attacks. In the hybrid attacks, the followings are assumed with realistic cost model:

- Reduced basis follows geometric series assumption.
- Square-root speed-up is obtained for the meet in the middle attack with success probability 1.
- There are no cost for the memory.

In the third parameter set with $\eta = 1$, the expected numbers of repetitions in rejection samplings are 1.58, 1.74 and 1.98 at the three security levels, respectively, which are smallest among the three parameter sets. Thus, the third parameter set provides the fastest signing among the three parameter sets.

We estimate their cost in the Core-SVP model. LWE and SIS security is estimated using the script from https://github.com/pq-crystals/security-estimates.

Parameter/Security Level	1	3	5
p	1021	1429	1913
q	8339581	8376649	8343469
d [dropped bits from t] ($2^d\tau < \gamma_2$)	11	12	12
τ [# of ± 1's in c]	25	29	32
challenge entropy [$\log \binom{p}{\tau} + \tau$]	190	228	259
γ_1 [y coefficient range]	2^{17}	2^{18}	2^{19}
γ_2 [low-order rounding range]	$(q-1)/90$ $(= 92662)$	$(q-1)/56$ $(= 149583)$	$(q-1)/42$ $(= 198654)$
η [secret key range]	2	2	2
β	100	116	128
ω [max # of 1's in hint]	80	80	80
Exp. reps. [$\approx e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}$]	6.6	5.7	5.5
Key/Signature Size			
Public key size	1564	1997	2663
Secret key size	2266	3312	4402
Signature size	2458	3605	5055
SIS Hardness (Core-SVP)			
BKZ block size b to break SIS	417	599	796
Best known classical bit cost	121	175	232
Best known quantum bit cost	110	158	211
LWE Hardness (Core-SVP)			
BKZ block size b to break LWE	421	648	932
Best known classical bit cost	123	189	272
Best known quantum bit cost	111	171	247
LWE Estimator			
Cost to SIS (BKZ b)	133.9 (411)	198.1 (629)	259.8 (839)
Quantum cost to SIS	115.9	173.9	229.7
Cost to LWE by estimator (BKZ b)	147.7 (413)	211.5 (641)	291.3 (924)
Quantum cost to LWE	123.0	182.0	255.6

Table 2: First parameter set of non-cyclotomic case

Parameter/Security Level	1^c	3^c	5^c
p	1201	1607	2039
q	17279291	17305741	17287423
d [dropped bits from t] ($2^d\tau < \gamma_2$)	12	13	13
τ [# of ± 1's in c]	32	32	32
challenge entropy [$\log \binom{p}{\tau} + \tau$]	241	254	265
γ_1 [y coefficient range]	2^{19}	2^{19}	2^{19}
γ_2 [low-order rounding range]	$(q-1)/70$ ($= 246847$)	$(q-1)/60$ ($= 288429$)	$(q-1)/58$ ($= 298059$)
η [secret key range]	2	2	2
β	128	128	128
ω [max # of 1's in hint]	80	80	80
Exp. reps. [$\approx e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}$]	2.5	3.02	3.95
Key/Signature Size			
Public key size	1984	2443	3091
Secret key size	2800	3914	4940
Signature size	3186	4251	5385
SIS Hardness (Core-SVP)			
BKZ block size b to break SIS	463	666	895
Best known classical bit cost	135	194	261
Best known quantum bit cost	122	176	237
LWE Hardness (Core-SVP)			
BKZ block size b to break LWE	491	711	956
Best known classical bit cost	143	207	279
Best known quantum bit cost	130	188	253
LWE Estimator			
Cost to SIS (BKZ b)	155.5 (484)	218.1 (697)	289.7 (941)
Quantum cost to SIS	135.3	192.0	256.8
Cost to LWE (BKZ b)	167.3 (483)	229.3 (704)	298.1 (949)
Quantum cost to LWE	141.1	198.4	262.0

Table 3: Second parameter set of non-cyclotomic case ($\eta = 2$)

Parameter/Security Level	$1^{c,1}$	$3^{c,1}$	$5^{c,1}$
p	1201	1607	2039
q	17279291	17305741	17287423
d [dropped bits from t] $(2^d\tau < \gamma_2)$	12	13	13
τ [# of ± 1's in c]	32	32	32
challenge entropy $[\log \binom{p}{\tau} + \tau]$	241	254	265
γ_1 [y coefficient range]	2^{19}	2^{19}	2^{19}
γ_2 [low-order rounding range]	$(q-1)/70 =$ 246847	$(q-1)/60 =$ 288429	$(q-1)/58 =$ 298059
η [secret key range]	1	1	1
β	64	64	64
ω [max # of 1's in hint]	80	80	80
Exp. reps. $[\approx e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}]$	1.58	1.74	1.98
pk size	1984	2443	3091
sk size	2703	3817	4843
sig size	3936	5255	6659
BKZ block-size b to break SIS	463	666	895
Best Known Classical bit-cost	135	194	261
Best Known Quantum bit-cost	122	176	237
Best Plausible bit-cost	96	138	185
BKZ block-size b to break LWE	450	656	884
Best Known Classical bit-cost	131	191	258
Best Known Quantum bit-cost	119	174	234
Core-SVP cost by Lattice estimator			
BKZ block-size b to break LWE	442	642	863
Classical bit-cost (method)	129.1 (usvp)	187.8 (dual hybrid)	252.2 (dual hybrid)
Hybrid-decoding attack cost			
BKZ block-size b to break LWE	445	655	890
Classical bit-cost	168.6	231.4	301.5
Hybrid-dual attack cost			
BKZ block-size b to break LWE	430	621	842
Classical bit-cost	156.1	213.2	277.1

Table 4: Balanced third parameter set of non-cyclotomic case ($\eta = 1$)

LWE cost by the lattice estimator is calculated from `https://github.com/malb/lattice-estimator`. Additionally, to consider hybrid attack also, we use lattice estimator (https://github.com/malb/lattice-estimator) to estimate the security of LWE. In Table 7, the model 'usvp' means that solving unique shortest vector problem is the best estimated strategy. For quantum security, we utilize the simple estimation method that uses classical security estimate with BKZ block size b. For this, we assume that solving the shortest vector problem in a lattice of dimension b costs $2^{0.292b}$ and $2^{0.265b}$ for classical and quantum attackers, respectively. Additionally, we assume the square-root quantum attacker for the rest attack cost. Namely, we estimate the quantum cost from the classical cost: $2^{a+0.292b}$ (classical) becomes $2^{a/2+0.265b}$ (quantum).

Number of Repetitions. We analyze the probability of the rejection in Sign algorithm using the proposed SampleInBall. We choose the challenge polynomial $\mathbf{c} \in \mathcal{R}$ having τ non-zero coefficients. For optimization, our optimized SampleInBall algorithm chooses $\mathbf{c} \in R = \mathbb{Z}[X]/(X^p - X - 1)$ differently: choose two (or more) separate polynomials. Now, we calculate the probability that Step 12-13 pass in Sign algorithm and investigate optimization effects of our algorithm for the suggested parameter sets.

Let κ be a challenge entropy, $p_1 = (p-1)/2$, and $p_2 = (p+1)/2$ with $p_1 + p_2 = p$. First, choose τ_1, τ_2 such that

$$\log \binom{p_1}{\tau_1} + \tau_1 + \log \binom{p_2}{\tau_2} + \tau_2 > \kappa.$$

Then choose $\mathbf{c} = \mathbf{c}_2 + X^{p_2}\mathbf{c}_1$, where $\mathbf{c}_i$ is a degree-$(p_i - 1)$ polynomial of coefficients in $\{-1, 0, 1\}$ and the sum of absolute value of the coefficient is τ_i for $i = 1, 2$. Now, consider the product $\mathbf{c} \cdot \mathbf{s} \in R$, where $\mathbf{s}$ has also small coefficients whose absolute value is not greater than η.

Let $\mathbf{t} = \mathbf{s} \cdot X^i$ and t_j be the j-th coefficient of $\mathbf{t}$. Then, for $i = 0$, it is clear that $|t_j| \leq \eta$ for all j. For $i = 1$, it can be seen that $|t_j| \leq \eta$ for all j except that $|t_1| \leq 2\eta$. For $i = 2$, it can also be seen that $|t_j| \leq \eta$ for all j but $j = 1, 2$ where $|t_1|, |t_2| \leq 2\eta$. Similarly, for $\mathbf{t} = \mathbf{s} \cdot X^i$, it can be seen that $|t_j| \leq \eta$ for all j except $j = 1, 2, ..., i$. Thus, for $i < p_2$, $|t_j| \leq \eta$ for $j \geq p_2$ and $|t_j| \leq 2\eta$ for $j < p_2$.

Now let $\mathbf{t} = \mathbf{s} \cdot \mathbf{c}_2 \in R$ and t_j be the coefficient of $\mathbf{t}$. Since $\mathbf{c}_2$ has a degree less than p_2 and has only τ_2 non-zero coefficients, we know that $|t_j| \leq \tau_2\eta$ for $j \geq p_2$, and $|t_j| \leq 2\tau_2\eta$ for $j < p_2$. Let $\mathbf{u} = \mathbf{s} \cdot \mathbf{c} \in R$ and u_j be the coefficient of $\mathbf{u}$. Then it can be seen that $|u_j| \leq (2\tau_1 + \tau_2)\eta$ for $j \geq p_2$, and $|u_j| \leq 2(\tau_1 + \tau_2)\eta$ for $j < p_2$. Let $\beta_1 = 2(\tau_1 + \tau_2)\eta$ and $\beta_2 = (2\tau_1 + \tau_2)\eta$. Let $\mathbf{z}$ be the signature and z_j be the coefficient of $\mathbf{z}$. Then in the signature generation, we can check $|z_j| < \beta_1$ for $j < p_2$ and $|z_j| < \beta_2$ for $j \geq p_2$ instead of $|z_j| < \beta$. Since β_2 is smaller than β_1 and β_1 is only slightly larger than β, the rejection probability could become smaller. More concretely, the expected repetitions become

$$e^{(p_1\beta_2 + p_2\beta_1)(1/\gamma_1 + 1/\gamma_2)}$$

instead of $e^{p\beta(1/\gamma_1 + 1/\gamma_2)}$. In Table 5, we can see that this optimization offers speed-up ranging from 9% to 24%, depending on the two parameter sets. The

numbers in parentheses of Exp. reps. are the expected numbers of repetitions calculated by $e^{p\beta(1/\gamma_1+1/\gamma_2)}$ in [19, 42].

Parameter	p	τ	κ	p_1, p_2	β_1, β_2	Exp. reps.	Speed-up
1	1021	25	190	510,511	104,76	5.44 (6.6)	1.21
3	1429	29	228	714,715	120,88	4.76 (5.7)	1.19
5	1913	32	259	956,957	128,96	4.42 (5.5)	1.24
1^c	1201	32	241	600,601	132,98	2.27 (2.5)	1.09
3^c	1607	32	254	803,804	132,98	2.7 (3.02)	1.11
5^c	2039	32	265	1019,1020	132,98	3.43 (3.95)	1.15

Table 5: Optimization effects for our parameter sets.

3.4 Parameter Selection for Cyclotomic Trinomial Counterpart

NCC-Sign supports a cyclotomic trinomial counterpart, where the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ is the m-th cyclotomic polynomial with m of the form $m = 2^a \cdot 3^b$, $a, b > 1$. It is given by $\phi(X) = X^n - X^{n/2} + 1$ and $n = \varphi(n) = m/3$.

The degree of the polynomial has of the form $2^a \cdot 3^b$ which allows us to choose flexible parameters. Possible degrees of the polynomial of the form $2^a \cdot 3^b$ are 512, 576, 648, 729, 768, 864, 972, 1024, 1152, 1296, 1458, 1536, 1728, 1944, 2048, 2187, and 2304. We select n as 1024, 1458 and 1944 in the first parameter set. For the use of NTT, we choose 1152, 1536, and 2304, depending on the required security level, where $1152 = 2^7 \cdot 3^2$ and $1536 = 2^9 \cdot 3$. For the 256-bit security level, we choose n so that it has less powers of 3: choose n as 2304 rather than 1944 since $1944 = 2^3 \cdot 3^5$ and $2304 = 2^8 \cdot 3^2$. We expect that using $n = 2304$ would be faster. We use the two types of modulus as follows:

- a power-of-2 modulus $q = 2^{23}$
- a modulus q for the use of NTT: we use a suitable prime modulus q larger than 2^{23}

We aim to choose the parameter set for the use of NTT that are closest to or exceed 128, 192, and 256 at the three security levels. The concrete parameter sets based on security analysis similar to the non-cyclotomic case are presented in Table 6 and Table 7. The parameter set of a power-of-2 modulus $q = 2^{23}$ is given in Table 6 and, for more speed up, the parameter set for the use of NTT is given in Table 7. As in the non-cyclotomic case, we estimate the security in the Core-SVP model. LWE and SIS security is estimated using the script from https://github.com/pq-crystals/security-estimates. Additionally, to consider hybrid attack also, we use lattice estimator (https://github.com/malb/lattice-estimator) to estimate the security of LWE.

In NTTRU KEM [36], the authors show that with appropriately chosen q, NTT over the ring $\mathbb{Z}_{7681}[X]/(X^{768} - X^{384} + 1)$ is as fast as that over power-of-2 rings. NTTRU is the fastest of any lattice-based NIST submissions. Unlike relatively small n and q in NTTRU, our modulus is large close to 2^{23} for efficient rejection samplings. We will investigate how much improvement the use of NTT will achieve for the parameter set with the modulus for NTT.

Parameter/Security Level	1	3	5
n	1024	1458	1944
q	2^{23}	2^{23}	2^{23}
d [dropped bits from t] ($2^d\tau < \gamma_2$)	12	12	13
τ [# of ± 1's in c]	25	29	32
challenge entropy [$\log \binom{p}{\tau} + \tau$]	190	230	263
γ_1 [y coefficient range]	2^{18}	2^{18}	2^{19}
γ_2 [low-order rounding range]	2^{17}	2^{17}	2^{18}
η [secret key range]	2	2	1
β	100	116	64
ω [max # of 1's in hint]	80	80	80
Exp. reps. [$\approx e^{n\beta(1/\gamma_1+1/\gamma_2)}$]	3.23	6.92	2.04
Key/Signature Size			
pk size	1440	2037	2462
sk size	2400	3377	4227
sig size	2529	3678	5135
SIS Hardness (Core-SVP)			
BKZ block-size b to break SIS	395	628	813
Best Known Classical bit-cost	115	183	237
Best Known Quantum bit-cost	104	166	215
LWE Hardness (Core-SVP)			
BKZ block-size b to break LWE	422	665	875
Best Known Classical bit-cost	123	194	256
Best Known Quantum bit-cost	111	176	232
Lattice estimator (Core-SVP)			
BKZ block-size b to break LWE	400	632	855
Classical bit-cost (method)	116.8 (usvp)	184.5 (usvp)	249.7 (dual hybrid)

Table 6: Parameter set of cyclotomic trinomial counterpart

Parameter/Security Level	1	3	5
n	1152	1536	2304
q	8401537	8397313	8404993
d [dropped bits from t] ($2^d\tau < \gamma_2$)	12	12	13
τ [# of ± 1's in c]	25	29	32
challenge entropy [$\log \binom{p}{\tau} + \tau$]	195	232	271
γ_1 [y coefficient range]	2^{18}	2^{18}	2^{19}
γ_2 [low-order rounding range]	131274	131208	262656
η [secret key range]	1	1	1
β	50	58	64
ω [max # of 1's in hint]	80	80	80
Exp. reps. [$\approx e^{n\beta(1/\gamma_1+1/\gamma_2)}$]	1.93	2.76	2.32
Key/Signature Size			
pk size	1760	2336	3200
sk size	2400	3168	4992
sig size	2912	3872	6080
SIS Hardness (Core-SVP)			
BKZ block-size b to break SIS	462	671	1005
Best Known Classical bit-cost	135	196	293
Best Known Quantum bit-cost	122	177	266
LWE Hardness (Core-SVP)			
BKZ block-size b to break LWE	451	652	1078
Best Known Classical bit-cost	131	190	315
Best Known Quantum bit-cost	119	172	285
Lattice estimator (Core-SVP)			
BKZ block-size b to break LWE	452	652	1072
Classical bit-cost (method)	132 (usvp)	190.7 (dual hybrid)	313.3 (dual hybrid)

Table 7: Parameter set of cyclotomic trinomial counterpart for the use of NTT

4 Implementation Details

We describe implementation details of non-cyclotomic case. We first explain a new optimized hashing to a ball using two separate polynomials and investigate its improvements. We also find modulus of special forms to improve modular reductions. We then describe polynomial multiplications and modular reductions. Our scheme follows the bit packing method similar to that in [19, 42].

4.1 Polynomial Multiplications

Algorithm 11: Toom-Cook Algorithm [15], [30]

Require: Two polynomials $A(x)$ and $B(x)$ of degree $N = 1023$
Ensure : $C(x) = A(x)B(x)$

Splitting
// $A_3, \cdots, A_0, B_3, \cdots, B_0$ are degree 255 polynomials
$A(y) = A_3y^3 + A_2y^2 + A_1y + A_0$ // $y = x^{256}$
$B(y) = B_3y^3 + B_2y^2 + B_1y + B_0$ // $y = x^{256}$

Evaluation
// Evaluation of the polynomials at $y = \{0, \pm1, \pm0.5, 2, \infty\}$.
// Using Karatsuba multiplication to get $w_1, \cdots, w_7$.
$w_1 = A(\infty)B(\infty) = A_3B_3$
$w_2 = A(2)B(2) = (A_0 + 2A_1 + 4A_2 + 8A_3)(B_0 + 2B_1 + 4B_2 + 8B_3)$
$w_3 = A(1)B(1) = (A_0 + A_1 + A_2 + A_3)(B_0 + B_1 + B_2 + B_3)$
$w_4 = A(-1)B(-1) = (A_0 - A_1 + A_2 - A_3)(B_0 - B_1 + B_2 - B_3)$
$w_5 = A(0.5)B(0.5) = (8A_0 + 4A_1 + 2A_2 + A_3)(8B_0 + 4B_1 + 2B_2 + B_3)$
$w_6 = A(-0.5)B(-0.5) = (8A_0 - 4A_1 + 2A_2 - A_3)(8B_0 - 4B_1 + 2B_2 - B_3)$
$w_7 = A(0)B(0) = A_0B_0$

Interpolation
$w_2 = w_2 + w_5$
$w_6 = w_6 - w_5$
$w_4 = (w_4 - w_3)/2$
$w_2 = w_5 - w_1 - 64w_7$
$w_3 = w_3 + w_4$
$w_5 = 2w_5 - w_6$
$w_2 = w_2 - 65w_3$
$w_3 = w_3 - w_7 - w_1$
$w_2 = w_2 + 45w_3$
$w_5 = (w_5 - 8w_3)/24$
$w_6 = w_6 + w_2$
$w_2 = (w_2 + 16w_4)/18$
$w_4 = -(w_4 + w_2)$
$w_6 = (30w_2 - w_6)/60$
$w_2 = w_2 - w_6$
return $C(y) = w_1y^6 + w_2y^5 + w_3y^4 + w_4y^3 + w_5y^2 + w_6y + w_7$

We cannot apply NTT to our scheme. The next best alternative is the 4-way Toom-Cook multiplication and Karatsuba multiplication used in [15], [30]. At first, 4-way Toom-Cook multiplication is performed in three steps : Splitting, Evaluation, Interpolation. Next, Karatsuba multiplication is used in the Evaluation step. To use these multiplication methods, the degree of polynomial must be $16l - 1$ for some integer l. Thus, for polynomial multiplication, we choose $N = 1023, 1439, 1919$ which is closest to $p = 1021, 1429, 1913$ (coefficients of degree k is 0 for $p \leq k \leq N$).

- **Splitting.** We split polynomial into four small polynomials. For example, if $A(x), B(x)$ are a degree 1023 polynomials then $A(y) = A_3y^3 + A_2y^2 + A_1y + A_0, B(y) = B_3y^3 + B_2y^2 + B_1y + B_0$, where $y = x^{256}$.
- **Evaluation.** We evaluate 7 values of two polynomials at $y = \{0, \pm 1, \pm 0.5, 2, \infty\}$. After Evaluation, multiplication two polynomials for each values using Karatsuba multiplication.
- **Interpolation.** We calculate $C(y) = A(y)B(y) = w_1y^6 + w_2y^5 + w_3y^4 + w_4y^3 + w_5y^2 + w_6y + w_7$ using Evaluation values at $y = \{0, \pm 1, \pm 0.5, 2, \infty\}$.

Algorithm 11 is the details of Splitting, Evaluation and Interpolation for $N = 1023$. Algorithm 12 is the details of Karatsuba multiplication.

Algorithm 12: Karatsuba Multiplication [15], [30]

Require: Two polynomials $A(x)$ and $B(x)$ of degree $N = 255$
Ensure : $C(x) = A(x)B(x)$ of degree $N = 510$ polynomial
// Splitting two polynomials
1 $A(y) = A_3y^3 + A_2y^2 + A_1y + A_0$ // $y = x^{64}$
2 $B(y) = B_3y^3 + B_2y^2 + B_1y + B_0$ // $y = x^{64}$
// $A(y)B(y) = (A_3B_3)y^6 + (A_3B_2 + A_2B_3)y^5 + (A_3B_1 + A_2B_2 + A_1B_3)y^4 + (A_3B_0 + A_2B_1 + A_1B_2 + A_0B_3)y^3 + (A_2B_0 + A_1B_1 + A_0B_2)y^2 + (A_1B_0 + A_0B_1)y + (A_0B_0)$
3 $w_1 = A_3B_3$
4 $w_3 = A_2B_2$
5 $w_5 = A_1B_1$
6 $w_7 = A_0B_0$
7 $w_2 = (A_3 + A_2)(B_3 + B_2) - w_1 - w_3$
8 $w_6 = (A_1 + A_0)(B_1 + B_0) - w_5 - w_7$
9 $w_8 = (A_3 + A_1)(B_3 + B_1)$
10 $w_9 = (A_2 + A_0)(B_2 + B_0)$
11 $w_4 = (A_3 + A_2 + A_1 + A_0)(B_3 + B_2 + B_1 + B_0)$
12 $w_5 = w_5 + w_9 - w_7 - w_3$
13 $w_3 = w_3 + w_8 - w_1 - w_5$
14 $w_4 = w_4 - w_8 - w_9 - w_2 - w_6$
15 **return** $C(y) = w_1y^6 + w_2y^5 + w_3y^4 + w_4y^3 + w_5y^2 + w_6y + w_7$

Algorithm 13: Signed Montgomery Reduction ($\beta = 2^{32}$) [41]

Require: $0 < q < \frac{\beta}{2}$ odd, $-\frac{\beta}{2}q \leq a = a_1\beta + a_0 < \frac{\beta}{2}q$ where $0 \leq a_0 < \beta$
Ensure : $r^{'} \equiv \beta^{-1}a \pmod{q}$, $-q < r' < q$
1 $m \leftarrow a_0 q^{-1} \bmod {}^{\pm}\beta$
2 $t_1 \leftarrow \lfloor \frac{mq}{\beta} \rfloor$
3 $r^{'} \leftarrow a_1 - t_1$

4.2 Modular Reductions

Our scheme performs polynomial multiplications over the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^p - X - 1)$. Using Montgomery reduction [37], our implementation avoids divisions and provides fast modular reductions. After coefficients of each polynomial are converted into Montgomery domain, the multiplication is conducted with the corresponding reduction to have the coefficients in $[0, q-1]$. After the multiplication is finished, the coefficients of each polynomial are converted to the original domain with coefficients of $[\frac{-q+1}{2}, \frac{q-1}{2}]$ by using the Algorithm 13. This is because infinity norm of polynomials is checked after multiplication. Original output of Algorithm 13 is in $(-q, q)$, however, our input is in $[0, q-1]$ so that the output is in $[\frac{-q+1}{2}, \frac{q-1}{2}]$.

p	q	$q-1$
1021	8290297 $(= 2^{23} - 2^{16} - 2^{15} - 2^3 + 1)$	$2^3 * 3^3 * 7 * 5483$
1447	8126431 $(= 2^{23} - 2^{18} - 2^5 - 2^1 + 1)$	$2 * 3 * 5 * 13 * 67 * 311$
1913	6287329 $(= 2^{23} - 2^{21} - 2^{12} - 2^5 + 1)$	$2^5 * 3^3 * 19 * 383$
1279	16736257 $(= 2^{24} - 2^{15} - 2^{13} + 1)$	$2^{13} * 3^2 * 227$
1621	16252861 $(= 2^{24} - 2^{19} - 2^6 - 2^2 + 1)$	$2^2 * 3 * 5 * 13 * 67 * 311$
2099	16515073 $(= 2^{24} - 2^{18} + 1)$	$2^{18} * 3^2 * 7$

Table 8: Modulus of specific form

A Special Form of q. We find several modulus q of special form which might be beneficial for the performance: q has small weight, which would be good for the modular reduction.

- **Low-weight q.** In Dilithium [19, 42], the modulus $q = 8380417 (= 2^{23} - 2^{13} + 1)$ is used. When this modulus is used, the modular reduction by q can be computed using only small number of shifts and additions. In our case, due to the inert condition of p and q, it is hard to find such special modulus. However, it was possible to find similar form modulus. For example, we could find $(p, q) = (1021, 8290297)$, where

$$q = 2^{23} - 2^{16} - 2^{15} - 2^3 + 1.$$

  Note that $q - 1 = 2^3 * 3^3 * 7 * 5483$. We list some of similar modulus q in Table 8.

4.3 Reference Implementation

Our implementation specifications are as follows:

- **Target Platform.** The computer we have used is equipped with an Intel(R) Core(TM) i7-12700K CPU at the constant clock frequency of 3.60GHz running Ubuntu 18.04.
- The results presented in Table 9 and Table 10 include the numbers of CPU cycles required by the key generation, signing and verification.
- Each result is an average of 100,000 measurements for each function using the C programming language with GNU GCC version 7.5.0 compiler.
- Signing performance of the non-cyclotomic scheme for the third parameter set is faster than those of the other parameters. The third parameter set uses larger modulus and $\eta = 1$, which leads to the smaller number of expected repetitions in the rejection sampling.
- Reference implementation of the cyclotomic trinomial scheme for the first parameter set is as fast as Dilithium. Performance of the scheme using NTT is expected to be faster than Dilithium.

Our reference implementation uses the SampleInBall algorithm in Dilithium [19, 42]. The new optimized SampleInBall algorithm and special forms of q will be used in our optimized implementation using AVX2.

Algorithm/Security Level	1	3	5
KeyGen	1,257,562	2,386,408	4,202,722
Sign	16,174,808	28,184,328	49,062,056
Verify	2,444,616	4,765,774	8,342,102

Table 9: Performance of the non-cyclotomic scheme for the first parameters

Algorithm/Security Level	1^c	3^c	5^c
KeyGen	1,727,508	2,965,942	4,700,228
Sign	11,768,076	20,816,964	42,227,652
Verify	3,400,702	5,876,246	9,324,876

Table 10: Performance of the non-cyclotomic scheme for the second parameters

Algorithm/Security Level	$1^{c,1}$	$3^{c,1}$	$5^{c,1}$
KeyGen	1,788,898	3,026,796	4,773,352
Sign	7,116,832	12,780,647	28,418,546
Verify	3,519,616	5,978,144	9,501,792

Table 11: Performance of the non-cyclotomic scheme for the third parameters

Algorithm/Security Level	1	3	5
KeyGen	147,168	233,524	362,170
Sign	831,534	2,765,390	2,932,946
Verify	221,306	384,872	611,102

Table 12: Performance of the cyclotomic trinomial scheme for the first parameters

5 Conclusion

We proposed an RLWE-based signature scheme, NCC-Sign, that supports two types of signature schemes based on the non-cyclotomic polynomial $\phi(X) = X^p - X - 1$ for intermediate security guarantee and its cylotomic trinomial counterpart $\phi(X) = X^n - X^{n/2} + 1$ for faster performance. Unlike Dilithium, in our second and third parameter sets, the classical Core-SVP estimates exceed 128, 192 and 256 bits at the three security levels and the expected number of repetitions in the rejection samplings are 1.58, 1.74 and 1.98 at the three security levels in the third parameter set, respectively. However, unlike the schemes which use either cyclotomic polynomials to enable the use of NTT and power-of-two moduli for efficient coefficient-wise operations, it is a challenging task to implement our scheme without using these optimization techniques. Our future work is to show how to efficiently implement our scheme by solving these problems. In the cylotomic trinomial case, the future work includes the followings:

- implementation using NTT over the ring $\mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ expected to be as fast as or faster than Dilithium,
- suggestion of more parameter sets appropriate for the security levels,
- the use of the Gaussian distribution to reduce the output size and trade-off between the output size and performance in Dilithium-G [18, 17],
- consideration of the NTRU-based key generation for more compact construction and to remove computational overhead for recovering the public key from a given seed.

References

1. NIST post-quantum cryptography standardization round 3 submissions. https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
2. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions. In: Annual International Cryptology Conference. pp. 153–178. Springer (2016)
3. Alkim, E., Barreto, P.S., Bindel, N., Krämer, J., Longa, P., Ricardini, J.E.: The lattice-based digital signature scheme qTESLA. In: International Conference on Applied Cryptography and Network Security. pp. 441–460. Springer (2020)
4. Bai, S., Galbraith, S.D.: An improved compression technique for signatures based on learning with errors. In: Cryptographers' Track at the RSA Conference. pp. 28–47. Springer (2014)
5. Barbosa, M., Barthe, G., Doczkal, C., Don, J., Fehr, S., Grégoire, B., Huang, Y.H., Hülsing, A., Lee, Y., Wu, X.: Fixing and mechanizing the security proof of fiat-shamir with aborts and dilithium. Cryptology ePrint Archive (2023)
6. Bauch, J., Bernstein, D.J., Valence, H.d., Lange, T., Vredendaal, C.v.: Short generators without quantum computers: the case of multiquadratics. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 27–59. Springer (2017)
7. Bernstein, D.J., Chuengsatiansup, C., Lange, T., Vredendaal, C.v.: NTRU prime: reducing attack surface at low cost. In: International Conference on Selected Areas in Cryptography. pp. 235–260. Springer (2017)
8. Biasse, J.F., Song, F.: Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In: Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms. pp. 893–902. SIAM (2016)
9. Bos, J., 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). pp. 353–367. IEEE (2018)
10. Bos, J.W., Lauter, K., Loftus, J., Naehrig, M.: Improved security for a ring-based fully homomorphic encryption scheme. In: IMA International Conference on Cryptography and Coding. pp. 45–64. Springer (2013)
11. Brumley, B.B., Chen, M.S., Chuengsatiansup, C., Lange, T., Marotzke, A., Tuveri, N., van Vredendaal, C., Yang, B.Y.: NTRU prime: round 3 20201007
12. Campbell, P., Groves, M., Shepherd, D.: Soliloquy: A cautionary tale. In: ETSI 2nd Quantum-Safe Crypto Workshop. vol. 3, pp. 1–9 (2014)
13. Chen, H., Lauter, K., Stange, K.E.: Security considerations for galois non-dual RLWE families. In: International Conference on Selected Areas in Cryptography. pp. 443–462. Springer (2016)
14. Chuengsatiansup, C., Prest, T., Stehlé, D., Wallet, A., Xagawa, K.: ModFalcon: Compact signatures based on module-NTRU lattices. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. pp. 853–866 (2020)
15. D'Anvers, J.P., Karmakar, A., Sinha Roy, S., Vercauteren, F.: Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In: International Conference on Cryptology in Africa. pp. 282–305. Springer (2018)

16. Devevey, J., Fallahpour, P., Passelègue, A., Stehlé, D.: A detailed analysis of fiat-shamir with aborts. Cryptology ePrint Archive (2023)
17. Devevey, J., Fawzi, O., Passelègue, A., Stehlé, D.: On rejection sampling in lyubashevsky's signature scheme. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 34–64. Springer (2022)
18. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. iacr tches 2018 (1), 238–268 (2018)
19. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
20. Eisenträger, K., Hallgren, S., Lauter, K.: Weak instances of PLWE. In: International Conference on Selected Areas in Cryptography. pp. 183–194. Springer (2014)
21. Elias, Y., Lauter, K.E., Ozman, E., Stange, K.E.: Provably weak instances of ring-LWE. In: Annual Cryptology Conference. pp. 63–92. Springer (2015)
22. Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., Yu, Y.: MITAKA: A simpler, parallelizable, maskable variant of. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 222–253. Springer (2022)
23. Fouque, P.A., 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. Submission to the NIST's post-quantum cryptography standardization process **36**(5) (2018)
24. 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. pp. 197–206 (2008)
25. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Annual International Cryptology Conference. pp. 112–131. Springer (1997)
26. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 530–547. Springer (2012)
27. 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)
28. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International algorithmic number theory symposium. pp. 267–288. Springer (1998)
29. Hülsing, A., Rijneveld, J., Schanck, J., Schwabe, P.: High-speed key encapsulation from NTRU. In: International Conference on Cryptographic Hardware and Embedded Systems. pp. 232–252. Springer (2017)
30. Karmakar, A., Mera, J.M.B., Roy, S.S., Verbauwhede, I.: Saber on ARM CCA-secure module lattice-based key encapsulation on ARM. Cryptology ePrint Archive (2018)
31. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 552–586. Springer (2018)
32. Kiyomoto, S., Takagi, T.: A compact digital signature scheme based on the module-LWR problem. In: Information and Communications Security: 22nd International

Conference, ICICS 2020, Copenhagen, Denmark, August 24–26, 2020, Proceedings. vol. 12282, p. 73. Springer (2020)
33. Lee, C., Pellet-Mary, A., Stehlé, D., Wallet, A.: An lll algorithm for module lattices. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 59–90. Springer (2019)
34. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: Proceedings of the forty-fourth annual ACM symposium on Theory of computing. pp. 1219–1234 (2012)
35. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
36. Lyubashevsky, V., Seiler, G.: Nttru: Truly fast ntru using ntt. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 180–201 (2019)
37. Montgomery, P.L.: Modular multiplication without trial division. Mathematics of computation **44**(170), 519–521 (1985)
38. Mukherjee, T., Stephens-Davidowitz, N.: Lattice reduction for modules, or how to reduce modulesvp to modulesvp. In: Annual International Cryptology Conference. pp. 213–242. Springer (2020)
39. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 271–288. Springer (2006)
40. NIST: NIST PQC standardization process: Announcing four candidates to be standardized, plus fourth round candidates. `https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4`
41. Seiler, G.: Faster avx2 optimized ntt multiplication for ring-LWE lattice cryptography. Cryptology ePrint Archive (2018)
42. Shi Bai, Léo Ducas, E.K.T.L.V.L.P.S.G.S., Stehlé, D.: CRYSTALS-Dilithium algorithm specifications and supporting documentation (version 3.1). `https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf`