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

Kyung-Ah Shim[1], Jeongsu Kim[1], and Hyeokdong Kwon[1]

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

Abstract. We present the RLWE-based signature scheme, NCC-Sign, using non-cyclotomic polynomials and cyclotomic trinomials, which follows the design paradigm of Dilithium based on Bai and Galbraith scheme with public key compression. NCC-Sign uses two types of polynomial rings, based on the non-cyclotomic polynomial $\phi(X) = X^p - X - 1$ for stronger security and the trinomial $\phi(X) = X^n - X^{n/2} + 1$ for efficiency. It provides higher security by choosing more conservative parameters whose classical Core-SVP estimates exceed or are close to 128, 192, and 256 bits at the three security levels, respectively. NCC-Sign using the non-cyclotomic polynomial provides stronger security guarantee than power-of-2 cyclotomic counterparts by minimizing structures that can be used by attackers. However, it is inefficient since it cannot use the optimized implementation techniques for the cyclotomic power-of-2 rings. Our cyclotomic trinomial counterpart meets both conservative security and efficiency requirements.

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 [2] 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 $\phi(X) = X^n + 1$, where n is a power of 2, and NTRU KEM uses the p-th cyclotomic polynomial $\phi(X) = X^p - 1$, where p is prime [8, 14, 16, 25, 19–21]. They achieve high speeds on several architectures as well as reasonably small signatures and key sizes.

There are advantages to choosing the cyclotomic polynomials, but there has been a potential threat of attacks using unnecessary algebraic structures [6, 10]. The attacks related to the additional algebraic structures exploit the fact that the field $\mathbb{Q}[X]/\phi(X)$ has many subfields for certain $\phi(X)$ [5, 3], some attacks use

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

the fact that a number field $\mathbb{Q}[X]/\phi(X)$ has small Galois group [11], and some attacks use the fact that there exist ring homomorphisms from $\mathbb{Z}_q[X]/\phi(X)$ to some smaller nonzero rings [17, 18, 12]. There is a sub-exponential time attack against NTRU assumptions with large moduli, which invalidated security guarantees of some FHE schemes [3, 23, 9]. There are also polynomial-time quantum attacks that 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 the plausible assumptions [7].

Although no attacks are known that perform significantly better on the schemes based on the structured lattices of cyclotomics, it remains possible that further cryptanalysis could exploit the structures. Therefore, it is necessary to think about countermeasures against the potential threats. As an opponent of these cyclotomics, a lattice-based KEM, NTRU Prime KEM, was 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 the NTRU Prime field [6], which aims to remove unnecessary structures exploited in attacks. Suggestions for the NTRU Prime field are as follows:

1. Choose $\phi(X)$ as a monic irreducible polynomial of prime degree p whose Galois group is isomorphic to S_p (the largest Galois group possible).
2. Choose a prime number q such 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 uses the non-cyclotomic polynomial $\phi(X) = X^p - X - 1$ for stronger security and the trinomial $\phi(X) = X^n - X^{n/2} + 1$ for efficiency. It is designed with the following design rationale.

[NCC-Sign Non-cyclotomic]

– **A RLWE-based Signature Scheme using the Non-cyclotomic Polynomial.** NCC-Sign follows the design paradigm of Dilithium based on Bai and Galbraith scheme with public key compression. It uses a prime-degree large Galois group inert modulus with $\phi(X) = X^p - X - 1$, which allows to eliminate the structures that were the causes of the previous attacks in the power-of-2 cyclotomic cases. Its existential unforgeability of our scheme is proved in (Q)ROM under the RLWE, RSIS and SelfTargetRSIS assumptions in a similar way to Dilithium [16, 25]. Unlike Dilithium based on the

MLWE/MSIS problems over the power-of-2 cyclotomic ring, NCC-Sign is based on the RLWE/RSIS problems over the non-cyclotomic polynomial ring. Such a choice leads to selection of more conservative parameters for higher security.

- **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 [6, 10] presented evidences that non-cyclotomic scheme had lower risks against the related classical and quantum attacks than the cyclotomic counterparts. NCC-Sign 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 provides speed-up ranging from 9% to 15% in the rejection sampling phase, depending on the parameter sets.
- **Flexible Choice of Parameters and Conservative Security.** 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. NCC-Sign provides flexibility in the parameter selection without any jumps appearing in the schemes. We select flexible conservative parameter sets whose classical Core-SVP estimates nearly equal or exceed 128, 192, and 256 bits at the three security levels, 1, 3, and 5, respectively. Particularly, the expected number of repetitions in the rejection samplings of NCC-Sign are up to 50% of Dilithium.
- **Protection against Side-Channel Attacks.** NCC-Sign uses uniform sampling to prevent the side-channel attacks targeting the discrete Gaussian distribution. All key dependent operations in our scheme are performed in a constant-time manner.

NCC-Sign using non-cyclotomic rings ensures stronger security guarantee against the potential threat, but is inefficient since it cannot use the optimized implementation techniques for the power-of-2 cyclotomic rings. Our cyclotomic trinomial counterpart meets both conservative security and efficiency requirement.

[NCC-Sign Trinomial]

- **A RLWE-based Signature Scheme using the Trinomial.** Our trinomial counterpart uses the m-th cyclotomic trinomial $\phi(X) = X^n - X^{n/2} + 1$, where $m = 2^a \cdot 3^b, a, b \geq 1$ and $n = \phi(m) = m/3$. The use of the trinomials with the extended form of degree $n = 2^a \cdot 3^b$ provides sufficient candidates of possible degrees which allows for choosing more conservative parameters for higher security.
- **Flexible Choice of Parameters and Conservative Security.** The trinomial with the extended form of degree allows use to select flexible parameter

selection for conservative security avoiding the parameter jump appeared in the RLWE-based schemes on the power-of-2 rings. While the classical Core-SVP estimates of Dilithium parameter sets are 123, 182, and 252 bits at the three security levels, respectively, those of our parameter sets exceed or are close to 128, 192, and 256 bits at the three security levels, respectively.

- **Efficient implementation.** NCC-Sign using the trinomial is faster than that using the non-cyclotomic polynomial. Although it uses larger degrees than Dilithium for higher security, its reference implementation is faster than that of Dilithium. However, our AVX2-optimized implementation is 1.8x to 2.2x times slower than Dilithium. Although the improvements in SHAKE optimization due to the module structure and NTT optimization are significant in AVX2 optimized implementation of Dilithium, the optimization techniques cannot be applied to our scheme. We believe our AVX2 optimized implementation of our scheme has room for improvement, especially in NTT optimization.
- **Protection against Side-Channel Attacks.** Our trinomial counterpart also uses uniform sampling and its all key dependent operations are performed in a constant-time manner.

2 Signature Scheme: NCC-Sign

2.1 Basic Operations

Throughout this document, in NCC-Sign Non-cyclotomic, 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. In NCC-Sign Trinomial, we let $R := \mathbb{Z}[X]/(X^n - X^{n/2} + 1)$ and $R_q := \mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ for $n = 2^a \cdot 3^b$ and prime number q. 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 $w \in \mathbb{Z}_q$, we 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 $\tilde{S}_\eta := \left\{ \mathbf{w} \mod^{\pm} 2\eta \ : \ \mathbf{w} \in R \right\}$. 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 [16, 25]. 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 $\mathsf{SampleInBall}$ 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** $\mathbf{c}$

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

$r := r \mod^+ q$
$r_0 := r \mod^\pm \alpha$
if $r - r_0 = q-1$ **then**
$\quad r_1 := 0$
$\quad r_0 := r_0 - 1$
else
$\quad r_1 := (r - r_0)/\alpha$
return (r_1, r_0)

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

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

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

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

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

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

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

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

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

$r_1 := \mathsf{HighBits}_q(r,\alpha)$
$v_1 := \mathsf{HighBits}_q(r+z,\alpha)$
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. $\mathsf{ExpandA}$, $\mathsf{ExpandS}$ and $\mathsf{ExpandMask}$ 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

The KeyGen, Sign and Verify algorithms of NCC-Sign are presented in Algorithm 8, 9, and 10, respectively.

Algorithm 8: KeyGen

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

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

$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$ (Trinomial version: $\hat{\mathbf{a}} \in T_q := \mathsf{ExpandA}(\zeta)$)
$\mu \in \{0,1\}^{512} := \mathrm{H}(ph \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho \in \{0,1\}^{512} := \mathrm{H}(K \parallel \mu)$ (or $\rho \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
  $\mathbf{y} \in \tilde{S}_{\gamma_1} := \mathsf{ExpandMask}(\rho, \kappa)$
  $\mathbf{w} := \mathbf{a}\mathbf{y}$ (Trinomial version: $\mathbf{w} := \mathsf{INTT}(\hat{\mathbf{a}} \circ \mathsf{NTT}(\mathbf{y}))$)
  $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
  $\tilde{c} \in \{0,1\}^{256} := \mathrm{H}(\mu \parallel \mathbf{w}_1)$
  $\mathbf{c} \in B_\tau := \mathsf{SampleInBall}_{p,\tau}(\tilde{c})$ (Trinomial version additionally stores $\hat{\mathbf{c}} := \mathsf{NTT}(\mathbf{c})$)
  $\mathbf{z} := \mathbf{y} + \mathbf{c}\mathbf{s}_1$ (Trinomial version: $\mathbf{z} := \mathbf{y} + \mathsf{INTT}(\hat{\mathbf{c}} \circ \mathsf{NTT}(\mathbf{s}_1))$)
  $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)$ (Trinomial version performs $\mathsf{INTT}(\hat{\mathbf{c}} \circ \mathsf{NTT}(\mathbf{s}_2))$ to compute $\mathbf{c}\mathbf{s}_2$)
  if $\|\mathbf{z}\|_\infty \geq \gamma_1 - \beta$ **or** $\|\mathbf{r}_0\|_\infty \geq \gamma_2 - \beta$ **then**
    $(\mathbf{z}, \mathbf{h}) := \perp$
  else
    $\mathbf{h} := \mathsf{MakeHint}_q(-\mathbf{c}\mathbf{t}_0, \mathbf{w} - \mathbf{c}\mathbf{s}_2 + \mathbf{c}\mathbf{t}_0, 2\gamma_2)$ (Trinomial version performs $\mathsf{INTT}(\hat{\mathbf{c}} \circ \mathsf{NTT}(\mathbf{t}_0))$ to compute $\mathbf{c}\mathbf{t}_0$)
    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: $\mathsf{Verify}(pk, M, \sigma = (\tilde{c}, \mathbf{z}, \mathbf{h}))$

$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$ (Trinomial version: $\hat{\mathbf{a}} \in T_q := \mathsf{ExpandA}(\zeta)$)
$\mu \in \{0,1\}^{512} := \mathrm{H}(\mathrm{H}(\zeta \parallel \mathbf{t}_1) \parallel M)$
$\mathbf{c} := \mathsf{SampleInBall}(\tilde{c})$ (Trinomial version additionally stores $\hat{\mathbf{c}} := \mathsf{NTT}(\mathbf{c})$)
$\mathbf{w}'_1 := \mathsf{UseHint}_q(\mathbf{h}, \mathbf{az} - \mathbf{ct}_1 \cdot 2^d, 2\gamma_2)$ (Trinomial version performs $\mathsf{INTT}(\hat{\mathbf{a}} \circ \mathsf{NTT}(\mathbf{z}))$ and $\mathsf{INTT}(\hat{\mathbf{c}} \circ \mathsf{NTT}(2^d \cdot \mathbf{t}_1))$ to compute $\mathbf{az}$ and $\mathbf{ct}_1 \cdot 2^d$, respectively)
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.

We let T_q to be the domain of NTT representations, and $\circ$ to be the coordinate-wise multiplication in T_q. NTT and INTT are NTT and inverse NTT operations, respectively. We note that a random string sampled from $\mathsf{ExpandA}$ can be interpreted as a random element in T_q as well. Therefore, we can write $\hat{\mathbf{a}} \in T_q := \mathsf{ExpandA}(\zeta)$ to sample the NTT representation of a random element $\mathbf{a} \in R_q$. The security proof of NCC-Sign in §3.1 requires the invertibility of $\mathbf{a}$. In fact, in Non-cyclotomic version, $\mathbf{a}$ is always invertible, but in Trinomial version, its invertibility must be checked. For this, $\mathsf{ExpandA}$ iteratively samples $\hat{\mathbf{a}}$ until none of the coefficients of $\hat{\mathbf{a}}$ is zero, which makes $\mathbf{a}$ invertible in step 3 of Algorithm 8.

3 Security Analysis and Parameter Selections

3.1 Existential Unforgeability

For security proof, 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*

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

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*

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

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*

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

We note that there is a classical reduction from RSIS to SelfTargetRSIS [16, 25].

For security analysis of our scheme, we need the following lemmas in [25, 16].

Lemma 1 ([16, 25]). *Suppose that q and α are positive integers satisfying $q > 2\alpha$, $q \equiv 1 \pmod{\alpha}$ and α is 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 ([16, 25]). *If $\|\mathbf{s}\|_\infty \leq \beta$ and $\|\mathsf{LowBits}_q(\mathbf{r}, \alpha)\|_\infty < \alpha/2 - \beta$, then*

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

Sketch of Security Proofs. 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) [4], . The security of Dilithium is analyzed in [22] where Fiat-Shamir signatures are analyzed and it is shown that deterministic UF-NMA secure signature schemes are also UF-CMA-secure in the QROM. However, the gap in CMA to NMA reduction is found and fixed in [4, 15]. The gap lies in the fact that non-accepting transcripts disturbs the distribution of random oracle answers towards accepting transcripts. To fix this gap, [15] considers HVZK simulator for reprogramming accepted and rejected transcripts and [4] uses additional hybrid step that removes rejected transcripts. In the end, with worse bound, [4, 15] provide the security proof of Dilithium which need larger commitment min-entropy.

Since NCC-Sign is a ring-version of Dilithium over the non-cyclotomic ring and the cyclotomic trinomial ring, the security proof of Dilithium can be applied to NCC-Sign as it is. What we have to do is to check the bound when two polynomials are multiplied and when $k = l = 1$ is applied. NCC-Sign has slightly worse bound than Dilithium because of the coefficient growth when two polynomials are multiplied. We sketch that NCC-Sign achieves acHVZK and UF-NMA in (Q)ROM. We assume that a public key is given without the compression. Proving security in this case also shows the security when compression is used. We let H to be a random oracle that maps its input to an element in B_τ.

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 $\|\mathbf{v}\|_\infty \le 2\gamma_2 + 1$. Let $\mathbf{t} = \mathbf{t}_1 \cdot 2^d + t_0$ where $\|\mathbf{t}_0\|_\infty \le 2^{d-1}$. 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 $\|\mathbf{v}'\|_\infty \le 2\tau 2^{d-1} + 2\gamma_2 + 1$. It follows that using adversary, we find $\mathbf{z}, \mathbf{c}, \mathbf{v}', M$ such that $\|\mathbf{z}\|_\infty < \gamma_1 - \beta$, $\|\mathbf{c}\|_\infty = 1$, $\|\mathbf{v}'\|_\infty \le 2\tau \cdot 2^{d-1} + 2\gamma_2 + 1$, $M \in \{0,1\}^*$, 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, $\Pr[\mathbf{z},\mathbf{c}] = \Pr[\mathbf{c}]\Pr[\mathbf{y} = \mathbf{z} - \mathbf{c}\mathbf{s}_1 \mid \mathbf{c}]$ where $\|\mathbf{z}\|_\infty \le \gamma_1 - \beta$. If $\|\mathbf{c}\mathbf{s}_i\|_\infty \le \beta$, then $\|\mathbf{z} - \mathbf{c}\mathbf{s}_i\|_\infty \le \gamma_1 - 1$. 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 $\|\mathbf{r}_0\|_\infty = \|\mathsf{LowBits}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)\|_\infty = \|\mathsf{LowBits}_q(\mathbf{a}\mathbf{z} - \mathbf{c}\mathbf{t}, 2\gamma_2)\|_\infty \le \gamma_2 - \beta$. 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.

At last, we compute the commitment min-entropy in the security proof of NCC-Sign, which requires the invertibility of $\mathbf{a}$. When $\phi = X^p - X - 1$, $\mathbf{a}$ is always invertible, but it is not in the case that $\phi = X^n - X^{n/2} + 1$. The key generation in Algorithm 8 checks its invertibility in NCC-Sign Trinomial.

Commitment min-entropy. Let $R = \mathbb{Z}[X]/\phi$ and $R_q = \mathbb{Z}_q[X]/\phi$. We consider the min-entropy of the commitment in the underlying sigma protocol. First, we assume that $\mathbf{a}$ is invertible. Now, for every possible $\mathbf{w}_1$, we need to compute the probability that $\mathsf{HighBits}_q(\mathbf{a}\mathbf{y}, 2\gamma_2)$ equals $\mathbf{w}_1$ when y is uniformly sampled from $\tilde{S}_{\gamma_1}$ for $\mathbf{a},\mathbf{y} \in R_q$ where $|\tilde{S}_{\gamma_1}| = (2\gamma_1)^n$. To compute the probability, we consider the set $T_{\mathbf{w}_1} = \{\mathbf{w} \in R_q \mid \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2) = \mathbf{w}_1\}$. Note that $|T_{\mathbf{w}_1}| \approx (2\gamma_2)^n$. We have $Pr_{y \leftarrow \tilde{S}_{\gamma_1}}[\mathsf{HighBits}_q(\mathbf{a}\mathbf{y}, 2\gamma_2) = \mathbf{w}_1] = Pr_{y \leftarrow \tilde{S}_{\gamma_1}}[\mathbf{a}\mathbf{y} = \mathbf{w},\ \mathbf{w} \in T_{\mathbf{w}_1}] = Pr_{y \leftarrow \tilde{S}_{\gamma_1}}[\mathbf{y} = \mathbf{w}\mathbf{a}^{-1},\ \mathbf{w} \in T_{\mathbf{w}_1}] \le \frac{|T_{\mathbf{w}_1}|}{|\tilde{S}_{\gamma_1}|} \approx \left(\frac{\gamma_2}{\gamma_1}\right)^n$, and the minimum entropy is about $n/2$ to n depending on the parameter set. This minimum entropy is larger than 500 already for the our parameter at the security level 1 in §3.3.

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

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

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 slope(b).
- The last ℓ_is are constant equal to 0 (possibly empty).

Throughout this section, we write $\text{vec}(\mathbf{x}) = [x_0, x_1, \cdots, x_{p-1}]^T$ when $\mathbf{x} = x_0 + x_1 X + \ldots x_{p-1}X^{p-1} \in R_q$, and $\text{rot}(\mathbf{x})$ 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 NCC-Sign

According to our security proof, NCC-Sign 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 $k = 1$, $\zeta' = \max\{2(\gamma_1 - \beta), 4\gamma_2 + 2\}$

[NCC-Sign Non-cyclotomic]
We aim to choose conservative parameter sets whose classical Core-SVP estimates are exceed 128, 192, and 256 at the three security levels, respectively.

Table 1: Some inert primes q for a given p

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

- 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 Table 1, we list some inert primes q for a given p. We choose suitable p and q such that the expected number of repetitions in the rejection samplings is not too large for efficiency.

- The first parameter sets using $\eta = 2$ is given in Table 2, where their Core-SVP estimates are 135, 194, and 261 bits at the three security levels, respectively, and the LWE cost is higher than the SIS cost. We choose another parameter sets using $\eta = 1$ given in Table 3 to balance the security of SIS and LWE problems, where the SIS cost is only slightly higher than the LWE cost.
- Actually, 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. In the second parameter set of $\eta = 1$, the expected numbers of repetitions in the rejection samplings are 1.58, 1.74, and 1.98 at the three security levels, respectively, which are reduced by half, compared to the first parameter sets.
- In the Tables, Exp. reps. represents the expected numbers of repetitions in rejection samplings. The expected numbers in the first parameter set is 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 paragraph. In Table 2, they are calculated by $e^{n\beta(1/\gamma_1+1/\gamma_2)}$ as in Dilithium.
- We estimate cost of NCC-Sign parameters in the Core-SVP model. LWE and SIS security is estimated using the script from `https://github.com/pq-crystals/security-estimates`. LWE cost by the lattice estimator is calculated from `https://github.com/malb/lattice-estimator`.
- The use of $\eta = 1$ means that the ternary secret and error are used. It is known that hybrid attacks are more effective to the ternary secret case. We provide cost analysis of the second parameter sets against the hybrid attacks. For security of the second parameter set 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 Table 3, 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).

Table 2: Parameter set of NCC-Sign Non-cyclotomic ($\eta = 2$).

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: Balanced parameter set of NCC-Sign Non-cyclotomic ($\eta = 1$)

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. [$\approx e^{n\beta(1/\gamma_1+1/\gamma_2)}$]	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	129.1	187.8	252.2
(method)	(usvp)	(dual hybrid)	(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

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 4, 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 [16, 25].

Table 4: Optimization effects for our parameter sets using new SampleInBall.

Parameter	p	τ	κ	p_1, p_2	β_1, β_2	Exp. reps.	Speed-up
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

[NCC-Sign Trinomial]
NCC-Sign Trinomial uses the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$, where $\varphi(X) = X^n - X^{n/2} + 1$ is the m-th cyclotomic polynomial with m of the form $m = 2^a \cdot 3^b$, $a, b \geq 1$ and $n = \varphi(n) = m/3$. We aim to choose conservative parameter sets whose classical Core-SVP estimates are closest to or exceed 128, 192, and 256 at the three security levels, respectively. Depending on the complexity of the LWE problem and the SIS problem at the required security levels, we choose appropriate parameter sets so that the expected number of repetition in the rejection samplings is not too large.

- The degree of the polynomial has of the form $2^a \cdot 3^b$ which allows 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 choose 1152, 1536, and 2304, depending on the required security levels, where $1152 = 2^7 \cdot 3^2$ and $1536 = 2^9 \cdot 3$. For the 256-bit security level, we choose two types of degrees as $n = 2048 = 2^{11}$ and $n = 2304 = 2^8 \cdot 3^2$.
- Our modulus is chosen prime q that satisfies the NTT condition in NTTRU KEM [13] using radix-3 and radix-2 NTT. NTTRU KEM [13] shows 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. In the case of $n = 1152, 1536$, and 2304, q is larger than 2^{23} which lead to efficient rejection samplings. The parameter set of $n = 2048$ has smaller output sizes than that with $n = 2304$, but the number of repetitions in the rejection samplings is 2 times larger than that of $n = 2304$. It is because that, compared to $n = 2304$, the modulus q is smaller than 2^{23} and the size of γ_2 is reduced by half.
- The concrete parameter set is presented in Table 5. As in the case of non-cyclotomic, LWE and SIS security is estimated using the script from `https://github.com/pq-crystals/security-estimates`. LWE cost by the lattice estimator is calculated from `https://github.com/malb/lattice-estimator`.

Our parameter choice is different from Dilithium [16, 25] and NTRU Prime KEM [6, 10].

- In NTRU Prime KEM [6, 10], the smallest p is 653 with $q = 4621$, but NCC-Sign Non-cyclotomic needs 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.
- While Dilithium uses a single prime q for the modulus at all security levels, q in NCC-Sign is different at each security level since it needs an inert modulus q for each prime p in NCC-Sign Non-cyclotomic and an appropriate q for the use of incomplete NTT in NCC-Sign Trinomial.

Table 5: Parameter set of NCC-Sign Trinomial.

Parameter/Security Level	1	3	$5'$	5
n	1152	1536	2048	2304
q	8401537	8397313	8380417	8404993
d [dropped bits from t] ($2^d\tau < \gamma_2$)	12	12	11	13
τ [# of ± 1's in c]	25	29	32	32
challenge entropy [$\log \binom{p}{\tau} + \tau$]	195	232	265	271
γ_1 [y coefficient range]	2^{18}	2^{18}	2^{18}	2^{19}
γ_2 [low-order rounding range]	131274	131208	130944	262656
η [secret key range]	1	1	1	1
β	50	58	64	64
ω [max # of 1's in hint]	80	80	80	80
Exp. reps. [$\approx e^{n\beta(1/\gamma_1+1/\gamma_2)}$]	1.93	2.76	4.49	2.32
Key/Signature Size				
pk size	1760	2336	3104	3200
sk size	2400	3168	3936	4992
sig size	2912	3872	5152	6080
SIS Hardness (Core-SVP)				
BKZ block-size b to break SIS	462	671	963	1005
Best Known Classical bit-cost	135	196	281	293
Best Known Quantum bit-cost	122	177	255	266
LWE Hardness (Core-SVP)				
BKZ block-size b to break LWE	451	652	934	1078
Best Known Classical bit-cost	131	190	273	315
Best Known Quantum bit-cost	119	172	247	285
Lattice estimator (Core-SVP)				
BKZ block-size b to break LWE	452	652	930	1072
Classical bit-cost	132	190.7	271.7	313.3
(method)	(usvp)	(dual hybrid)	(dual hybrid)	(dual hybrid)

[NCC-Sign vs. Dilithium.] We compare NCC-Sign and Dilithium in Table 6 in terms of the Core-SVP estimates for the LWE/SIS problems, output sizes and the expected number of repetitions. The classical Core-SVP estimates of Dilithium parameters are 123, 182, and 252 bits at the three security levels, respectively. However, the classical Core-SVP estimates of NCC-Sign parameters exceed or are close to 128, 192, and 256 bits at the three security levels and the expected number of repetitions in the rejection samplings are smaller than that of Dilithium.

Table 6: Comparison of Our Scheme and Dilithium

Scheme	Core-SVP/Size	1	3	5 (5′)
Dilithium	SIS	123	186	265
	LWE	123	182	252
	Repetitions	4.25	5.1	3.85
	Public key size	1312	1952	2592
	Signature size	2420	3293	4595
NCC-Sign Non-Cyclotomic ($\eta = 2$)	SIS	135	194	261
	LWE	143	207	279
	Repetitions	2.27	2.7	3.43
	Public key size	1984	2443	3091
	Signature size	3186	4251	5385
NCC-Sign Non-Cyclotomic ($\eta = 1$)	SIS	135	194	261
	LWE	131	191	258
	Repetitions	1.58	1.74	1.98
	Public key size	1984	2443	3091
	Signature size	3936	5255	6659
NCC-Sign Trinomial	SIS	135	196	293 (281)
	LWE	131	190	315 (273)
	Repetitions	1.93	2.71	2.32 (4.49)
	Public key size	1760	2336	3200 (3104)
	Signature size	2912	3872	6080 (5152)

3.4 Computational Efficiency

We measure the performance of each algorithm on Intel i9-10980XE @3.00 GHz for the non-cyclotomic case and an Intel Xeon(R) Gold 6234 @3.3 GHz for the trinomial case. The results presented in Table 7 and Table 8 include the numbers of CPU cycles required by the key generation, signing, and verification. Each result of signing, verification and key generation is a median of 10,000 measurements. The source code was developed on Ubuntu 22.04 LTS, compiled using gcc 11.4.0, and optimization level -O3 applied. Hyperthreading and Turbo Boost are switched off.

[NCC-Sign Non-cyclotomic] NCC-Sign Non-cyclotomic uses the NTT-based polynomial multiplication for NTT-unfriendly ring, $\mathbb{Z}_q[X]/(X^p - X - 1)$, as in [24]. Table 7 shows the reference implementation results of the NCC-Sign Non-cyclotomic.

Table 7: Performance of NCC-Sign Non-cyclotomic (Reference code).

Security Level	1^c	3^c	5^c
KeyGen	979,979	1,001,022	1,034,193
Sign	7,269,506	8,752,038	10,719,703
Verify	1,863,350	1,884,647	1,926,235
Security Level	$1^{c,1}$	$3^{c,1}$	$5^{c,1}$
KeyGen	994,925	979,361	1,036,603
Sign	5,125,748	5,591,804	6,251,357
Verify	1,862,862	1,904,330	1,926,090

[NCC-Sign Trinomial] NCC-Sign Trinomial uses the NTT-based polynomial multiplication for NTT-friendly rings $\mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ as in [13]. Table 8 shows the reference and AVX2-optimized implementation results of the NCC-Sign Trinomial.

Table 8: Performance of NCC-Sign Trinomial and Dilithium.

Scheme	Security Level	1	3	5
Performance (Reference Code, median cycles)				
NCC-Sign-Trinomial	KeyGen	240,496	324,140	488,168
	Sign	616,746	1,245,144	1,781,784
	Verify	339,698	460,808	722,320
Dilithium	KeyGen	283,234	535,272	821,502
	Sign	973,868	1,658,510	2,206,464
	Verify	311,572	512,900	852,874
Performance (AVX2-optimized, median cycles)				
NCC-Sign-Trinomial	KeyGen	164,184	218,772	335,440
	Sign	290,396	553,728	838,432
	Verify	158,138	200,242	340,382
Dilithium	KeyGen	73,720	126,556	198,860
	Sign	178,436	289,862	353,008
	Verify	79,534	128,602	199,366

For comparison, we also present the implementation results of Dilithium. The performance of both algorithms was measured under the same environment and conditions.9 Although NCC-Sign Trinomial requires larger degrees than Dilithium for higher security, its reference implementation is faster than Dilithium. However, our AVX2-optimized implementation is 1.8x to 2.2x times slower. The improvement of SHAKE optimization and NTT optimization is significant in the AVX2-optimized implementation of Dilithium, but the optimization techniques cannot be applied to our scheme. We believe our AVX2 optimized implementation of our scheme has room for improvement, especially in NTT optimization.

3.5 Application

We provide a pilot testing for feasibility in the CA system using NCC-Sign Trinomial as a digital signature scheme of CA and users. We measure the performance of each algorithm in the CA system on an Intel(R) Core (TM) i9-13900K @2.99 GHz. Each result of each algorithm is an average of 10,000 measurements. As seen in Fig. 1, key pair generation, certificate generation and certificate verification take 0.0572 ms, 0.2077 ms, and 0.3012 ms, respectively. In the cert verify, the system verifies the root CA's signature and the CA's signature.

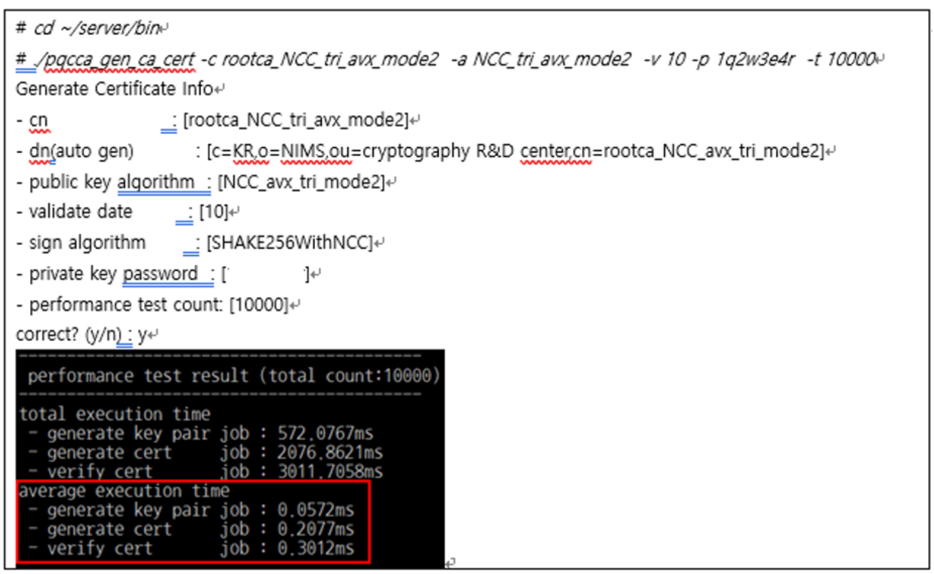

Fig. 1: Performance of NCC-Sign Trinomial in the CA system.

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. 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
3. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions. In: Annual International Cryptology Conference. pp. 153–178. Springer (2016)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)
9. 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)
10. 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
11. Campbell, P., Groves, M., Shepherd, D.: Soliloquy: A cautionary tale. In: ETSI 2nd Quantum-Safe Crypto Workshop. vol. 3, pp. 1–9 (2014)
12. 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)
13. Chung, C.M.M., Hwang, V., Kannwischer, M.J., Seiler, G., Shih, C.J., Yang, B.Y.: Ntt multiplication for ntt-unfriendly rings: New speed records for saber and ntru on cortex-m4 and avx2. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 159–188 (2021)
14. 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)
15. Devevey, J., Fallahpour, P., Passelegue, A., Stehlé, D.: A detailed analysis of fiat-shamir with aborts. Cryptology ePrint Archive (2023)
16. 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)

17. Eisenträger, K., Hallgren, S., Lauter, K.: Weak instances of PLWE. In: International Conference on Selected Areas in Cryptography. pp. 183–194. Springer (2014)
18. 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)
19. 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)
20. 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)
21. 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)
22. 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)
23. 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)
24. Lyubashevsky, V., Seiler, G.: Nttru: truly fast ntru using ntt. Cryptology ePrint Archive (2019)
25. 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`