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. Our scheme follows the Fiat-Shamir paradigm and combines the Bai-Galbraith scheme with several improvements from previous lattice-based schemes including CRYSTALS-Dilithium. It provides stronger security guarantee than cyclotomic counterparts and comparable key sizes and signature sizes to CRYSTALS-Dilithium. 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 based on the security proofs and cost analysis against the lattice attacks on known cost models. At last, we provide its performance evaluations.

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

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, CRYSTALS-Kyber, Saber, CRYSTALS-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 [11, 17, 18, 39, 22, 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 [8, 13]. The attacks exploited some additional structures use the fact that the field $\mathbb{Q}[X]/\phi(X)$ has many subfields for certain $\phi(X)$ [7, 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 [14], and some attacks using ring homomorphisms from $\mathbb{Z}_q[X]/\phi(X)$ to some smaller nonzero rings [19, 20, 15]. 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 [3, 33, 12]. There are polynomial-time quantum attacks broke Soiloquy, the cyclotomic case of Gentry's original fully homomorphic encryption (FHE) at STOC 2009 and the cyclotomic case of the Garg-Gentry-Halevi scheme under plausible assumptions [9].

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 [8] 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

Our scheme based on non-cyclotomic polynomials can get advantages in terms of security with relatively less structures than cylotomic cases. 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. We follow the design paradigm of CRYSTALS-Dilithium based on Bai and Galbraith scheme with public key compression. However, some critical distinctions exist between our scheme and CRYSTALS-Dilithium: our scheme is based on RLWE using non-cylcotomic polynomials instead of MLWE using the power-of-2 cylcotomic polynomial. The use of the non-cylcotomic polynomials leads to different selection of parameters and different implementation techniques. We also exploit a new optimized hashing to a ball using two separate polynomials. Consequently, our scheme provides stronger security guarantee than CRYSTALS-Dilithium and comparable key sizes and signature sizes.

Stronger Security Guarantee than Cylotomics. In the structured lattice-based schemes using cyclotomics, there have been proposed the potential attacks exploiting subfields, small Galois groups and ring homomorphisms, and polynomial-time quantum attacks on some HFE schemes. NTRU Prime KEM [8, 13] provide evidences that non-cyclotomic scheme has lower risks than the cyclotomics. We remove the additional structures that were the causes of the previous attacks to achieve stronger security guarantee by using the non-cyclotomic polynomial.

Security Proofs in ROM and QROM. Existential unforgeability of our scheme is proved in (quantum) random oracle model under the RLWE, RSIS and SelfTargetRSIS assumptions in a similar way to CRYSTALS-Dilithium [18, 39].

Flexible Choice of Parameters. In the lattice-based schemes based on the RLWE and MLWE problems using power-of-2 cyclotomics, the degree of polynomials n that must junp in increasingly by doubling or 256 bytes, respectively, due to the power of 2 restriction. Our scheme provides the flexibility for the parameter selections without the jumps that appear in the schemes.

Concrete/Conservative Parameters. We provide concrete parameters at NIST three security levels. We choose the parameters so that the rejection sampling in signing and the repeated number of rejections are the same level as CRYSTALS-Dilithium. Advanced attacks are still being proposed and predicted: recent improved dual lattice attacks [26, 35] considerably reduces the security levels of Kyber, Saber and CRYSTALS-Dilithium, the LWE/LWR-based schemes, bringing them below the thresholds defined by NIST. We suggest conservative parameters to allow security margins for future advances in cryptanalysis.

Protection against Side-Channel Attacks. The Fiat-Shamir with Aborts type signatures opt to sample their error vectors from a Gaussian distribution and used rejection sampling to hide the information about the secret-key in the signature. Most of the side channel analysis targetted the data dependent side-channel leakage from these Gaussian sampling, the rejection sampling components and the computation of the Number Theoretic Transform (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. All other operations such as polynomial multiplication and rounding are implemented in constant time.

Implementation. Compared to other lattice-based schemes which use either cylotomic 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. We use Toom-Cook and Karatsuba polynomial multiplications since NTT cannot be used to speed up polynomial multiplication in our case. We propose a new optimized hashing to a ball using two separate polynomials which offers speed-up ranging from 9% to 24%, depending on the parameter sets. Due to the lack of research on optimization for polynomial multiplication in non-cylotomic case, our scheme is less efficient than CRYSTALS-Dilithium, but we think that there still is room for optimization.

1.2 Related Works

The earlier lattice-based signatures, the GGH scheme [24] and NTRUSign [27], were completely broken by Nguyen and Regev [37] from the leakage of some secret information in lattice trapdoors. To prevent such leakage, Gentry, Peikert, and Vaikuntanathan [23] proposed a hash-and-sign type scheme secure under the hardness of worst-case lattice problems. At Eurocrypt 2012, Lyubashevsky [34] 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.* [25] proposed a compression technique without requiring Gaussian sampling based on the DCK and RSIS problem and Bai and Galbraith (BG) [6] 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 [5] based on RLWE and RSIS problems, CRYSTALS-Dilithium [18, 39] 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 [22] based on NTRU problem, its variant MITAKA [21] and ModFalcon [16] based on Module-NTRU problem. Recently, NIST recommended CRYSTALS-Dilithium and FALCON as digital signatures of NIST PQC Standardization [2].

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 $\tilde{S}_\eta := \left\{ \mathbf{w} \mod^{\pm} 2\eta \; : \; \mathbf{w} \in R \right\}$. 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.

Hashing to a Ball. We use multiple hashing algorithms that map strings in $\{0,1\}^*$ to random elements in desired domains such as S_η and R_q. We use $\mathsf{SampleInBall}$ algorithm to map 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 $\{-1, 0, 1\}$. 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 [18, 39]. 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.

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 c = c_0 c_1 ... c_{p-1} = 00...0
2 for i := p − τ to p − 1 do
3     j ← {0, 1, ..., i}
4     s ← {0, 1}
5     c_i := c_j
6     c_j := (−1)^s
7 return c
```

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

```
1 r := r mod⁺ q
2 r_0 := r mod± α
3 if r − r_0 = q − 1 then
4     r_1 := 0
5     r_0 := r_0 − 1
6 else
7     r_1 := (r − r_0)/α
8 return (r_1, r_0)
```

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

```
1 m := (q − 1)/α
2 (r_1, r_0) := Decompose_q(r, α)
3 if h = 1 and r_0 > 0 then
4     return
      (r_1 + 1) mod⁺ m
5 if h = 1 and r_0 ≤ 0 then
6     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± 2^d
3 return ((r − r_0)/2^d, r_0)
```

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

```
1 (r_1, r_0) := Decompose_q(r, α)
2 return r_1
```

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

```
1 (r_1, r_0) := Decompose_q(r, α)
2 return r_0
```

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

```
1 r_1 := HighBits_q(r, α)
2 v_1 := HighBits_q(r + z, α)
3 return [[r_1 ≠ 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

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

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

$\mathbf{a} \in R_q := \mathsf{ExpandA}(\zeta)$
$\mu \in \{0,1\}^{512} := \mathrm{H}(tr \parallel M)$
$\kappa := 0$, $(\mathbf{z}, \mathbf{h}) := \perp$
$\rho \in \{0,1\}^{512} := \mathrm{H}(K \parallel \mu)$ (or $\rho \leftarrow \{0,1\}^{512}$ for randomized signing)
while $(\mathbf{z}, \mathbf{h}) = \perp$ **do**
    $\mathbf{y} \in \tilde{S}_{\gamma_1} := \mathsf{ExpandMask}(\rho, \kappa)$
    $\mathbf{w} := \mathbf{a}\mathbf{y}$
    $\mathbf{w}_1 := \mathsf{HighBits}_q(\mathbf{w}, 2\gamma_2)$
    $\tilde{c} \in \{0,1\}^{256} := \mathrm{H}(\mu \parallel \mathbf{w}_1)$
    $\mathbf{c} \in B_\tau := \mathsf{SampleInBall}_{p,\tau}(\tilde{c})$
    $\mathbf{z} := \mathbf{y} + \mathbf{c}\mathbf{s}_1$
    $\mathbf{r}_0 := \mathsf{LowBits}_q(\mathbf{w} - \mathbf{c}\mathbf{s}_2, 2\gamma_2)$
    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** $[\![\#\text{ of 1's in } \mathbf{h} \text{ is} \leq \omega]\!]$

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 tr, then uses a secret key K 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 $\mathbf{a}$.

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 CRYSTALS-Dilithium [18, 39] to our case: $l = k = 1$ and $R = \mathbb{Z}[X]/(X^p - X - 1)$. We follow the proof in [18, 39] 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 and lemma.

Definition 2.1 (RLWE$_{q,D}$ Distribution). Let q be a positive ineteger. For a probability distribution $D : R_q \to \{0,1\}$, choose a random $\mathbf{a} \leftarrow R_q$ and a vector $\mathbf{s}_1, \mathbf{s}_2 \leftarrow D$, and output $(\mathbf{a}, \mathbf{as}_1 + \mathbf{s}_2)$.

Definition 2.2 (Decision RLWE Problem.) Given a pair $(\mathbf{a}, \mathbf{t})$ decode 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

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

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

Definition 2.3 (RSIS Problem.) The advantage of the adversary $\mathcal{A}$ to solve RSIS problem over the ring R_q is

$$\mathrm{Adv}_{\gamma}^{\mathrm{RSIS}}(\mathcal{A}) := Pr[0 < \|\vec{\mathbf{y}}\|_\infty \leq \gamma \wedge \begin{bmatrix}1 \ \mathbf{a}_1 \ \mathbf{a}_2\end{bmatrix} \cdot \vec{\mathbf{y}} = 0 \,|\, \mathbf{a}_1, \mathbf{a}_2 \leftarrow R_q; \vec{\mathbf{y}} \leftarrow \mathcal{A}(\mathbf{a}_1, \mathbf{a}_2)].$$

Definition 2.4. (SelfTargetRSIS Problem). For the cryptographic hash function H, the advantage of $\mathcal{A}$ to solve SelfTargetRSIS problem $\mathrm{Adv}_{\mathrm{H},\gamma}^{\mathrm{SelfTargetRSIS}}(\mathcal{A})$ is defined as

$$Pr\left[\begin{array}{c} 0 \leq \|\vec{\mathbf{y}}\|_\infty \leq \gamma \wedge \\ \mathrm{H}(\mu\| \begin{bmatrix}1 & \\ & \mathbf{a}_1 \ \mathbf{a}_2\end{bmatrix} \cdot \vec{\mathbf{y}}) = \mathbf{c} \end{array} \middle| \mathbf{a}_1, \mathbf{a}_2 \leftarrow R_q; \left(\vec{\mathbf{y}} := \begin{bmatrix}\mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{c}\end{bmatrix}, \mu\right) \leftarrow \mathcal{A}^{|\mathrm{H}(\cdot)\rangle}(a_1, a_2)\right].$$

We note that there is a classical reduction from RSIS to SelfTargetRSIS [18, 39].

Lemma 1 ([18, 39]). *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 $\mathbf{h} = \mathbf{h}'$*

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 [31], the authors showed that, for existential unforgeability against chosen-message attacks (UF-CMA), existential unforgeability against no-message attacks (UF-NMA) is sufficient if a signature scheme is zero-knowledge and deterministic. Since our scheme is deterministic, we show that our scheme achieves zero-knowledge 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 zero-knowledge. 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 $\|\mathbf{r}_0\|_\infty = \|\mathsf{LowBits}_q(\mathbf{w} - \mathbf{cs}_2, 2\gamma_2)\|_\infty = \|\mathsf{LowBits}_q(\mathbf{az} - \mathbf{ct}, 2\gamma_2)\|_\infty \leq \gamma_2 - \beta$.

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 quantum security, we assume that the SVP oracle costs in time $\approx 2^{0.265b}$. 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 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 $\zeta\sqrt{b}$, where ζ 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 the analysis of CRYSTALS-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

Based on the security estimates for RLWE and RSIS, we choose secure parameter sets for our scheme at the three security levels.

Selection of p and q. Our parameter choice is different from CRYSTALS-Dilithium [18, 39] and NTRU Prime KEM [8, 13].

- In NTRU Prime KEM [8, 13], the smallest p is 653 with $q = 4621$, but our smallest prime p is larger with the corresponding 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. We need the rejection sampling in signing for security: it makes the distribution of a signature independent from the secret key. For the 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. As a result, our p and q are larger than [8, 13], and it seems to be unavoidable.
- The size of q in our scheme is similar to CRYSTALS-Dilithium [18, 39]. While CRYSTALS-Dilithium uses a single prime q for the modulus for all security levels, our q is different at each security level. This is because we need inert modulus q. For each security level, we need to choose different prime p: for each prime p, different prime q inerts.

Actually, there exist enough candidate inert primes for each prime p. Now we explain the method to choose p and q. The expected number of repetitions in the rejection sampling is about $e^{p\beta(1/\gamma_1+1/\gamma_2)}$. Thus, we choose suitable p and q such that the expected number of repetitions is not too large for efficiency. Since we need inert modulus q, we find the candidate prime and modulus p and

q and check whether they satisfy the required security levels. To find the inert modulus prime q, we search q in the certain range. In our experiments in sage, we could 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 chose 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. In Table 1, we list some of inert primes q for a given p.

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

Table 1: Selection of p and q.

Concrete Parameters. 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 $k = 1$, $\zeta' = \max\{2(\gamma_1 - \beta), 4\gamma_2 + 2\}$

Unlike CRYSTALS-Dilithium, we cannot choose single prime q since we require q to be inert which depends on p. Thus, we choose suitable q from the prime p. We choose γ_1 as a power of two and choose 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 $\beta = 2\tau\eta$.

Concrete parameters are in Table 2. Costs are measured in cpu-cycles. LWE cost is calculated by lattice estimator from `https://github.com/malb/lattice-estimator`. For the quantum security, we use simple estimation method that use classical security estimate with BKZ block size b. For this, we assume that solving shortest vector problem in a lattice of dimension b costs $2^{0.292b}$ and $2^{0.265b}$ for classical and quantum attacker, 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).

Parameter/Seurity Level	I	III	V
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. [$\approx e^{p\beta(1/\gamma_1+1/\gamma_2)}$]	6.6	5.7	5.5
Public key size	1564	1997	2663
Secret key size	2266	3312	4402
Signature size	2458	3605	5055
Cost to SIS (BKZ β)	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 β)	147.7 (413)	211.5 (641)	291.3 (924)
Quantum cost to LWE	123.0	182.0	255.6

Table 2: Concrete Parameters for NCC-Sign.

Parameter/Security Level	I^c	III^c	V^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. [$\approx e^{p\beta(1/\gamma_1+1/\gamma_2)}$]	2.5	3.02	3.95
Public key size	1984	2443	3091
Secret key size	2800	3914	4940
Signature size	3186	4251	5385
Cost to SIS (BKZ β)	155.5 (484)	218.1 (697)	289.7 (941)
Quantum cost to SIS	135.3	192.0	256.8
Cost to LWE (BKZ β)	167.3 (483)	229.3 (704)	298.1 (949)
Quantum cost to LWE	141.1	198.4	262.0

Table 3: Conservative Parameters for NCC-Sign.

Conservative Parameters. Recently, researchers in MATZOV published a report which improves the dual lattice attack [35]. The dual lattice attack is considered to be less efficient than the primal lattice attack, previously. In [35], they improved the attack and showed that their attack was better than the primal attack for some LWE parameters. Considering future cryptanalysis, we provide more conservative parameter sets. For the cost of LWE, we use the lattice estimator [4] which includes the corrected sieving cost of [35] and will be updated with the dual attack model of [35] shortly. In the mean time, using the lattice estimator with conservative parameter sets seems to be good enough. For reference, we list the estimated security of LWE for some p and q's, where coefficients of secret and error are sampled uniformly from the set $\{-2, -1, 0, 1, 2\}$, as is chosen in our signature scheme. For the AES-like security, one needs $143, 207, 272$ for AES-128/192/256. Now, we provide conservative parameter sets in Table 3 to thwart attacks in the foreseeable future with larger q to lower the rejection probability.

3.4 Cost Analysis of Known Attacks

Relying on the 'LWE estimator' of Albrecht et al. [4], we provide cost analysis of our scheme against the primal and dual lattice attackson all cost models for lattice reductions.

Cost Models. We use the default option (MATZOV) in the lattice estimator for the cost estimation, but there exist other cost models. Although we are convinced to use the default option, we provide cost estimates of the RLWE problem on other cost models for reference in Table 4.

- 'bdd' means that solving a bounded distance decoding problem in the lattice is the best attack strategy. Bounded distance decoding problem can be easily converted to a unique shortest vector problem by the embedding approach and
- 'usvp' means that solving unique shortest vector problem is the best estimated strategy.
- 'bkw' means that Blum-Kalai-Wasserman [10] which needs quite many samples for the attack to succeed. More details can be found in [4].

Some cost models are simple and can be described in the following for the logarithmic cost of BKZ-β of dimension-d lattice:

- ABFKSW20: $0.125\beta \log_2 \beta - 0.547\beta + 10.4 + \log_2 64 + \log_2 8d$,
- ABLR21: $0.125\beta \log_2 \beta - 0.654\beta + 25.84 + \log_2 64 + \log_2 8d$,
- ADPS16: 0.292β,
- BDGL16: $0.292\beta + 16.4 + \log_2 8d$,
- CheNgu12: $0.270\beta \log \beta - 1.019\beta + 16.103 + \log_2 100 + \log_2 8d$,
- LaaMosPol14: $0.265\beta + 16.4 + \log_2 8d$.

Other cost models are more complex and can be found in the homepage of the estimator[1]. Kyber cost model uses dimension for free technique and gate

[1] https://github.com/malb/lattice-estimator/blob/main/estimator/reduction.py

Cost model	I (128)	III (192)	V (256)	I^c (128)	III^c (192)	V^c (256)
ABFKSW20	259.6 (usvp)	363.5 (bkw)	478.8 (bkw)	307.1 (bkw)	403.5 (bkw)	500.1 (bkw)
ABLR21	229.9 (usvp)	363.5 (bkw)	478.8 (bkw)	274.2 (usvp)	403.5 (bkw)	500.1 (bkw)
ADPS16	123.2 (usvp)	190.1 (usvp)	273.3 (usvp)	143.7 (usvp)	208.5 (usvp)	280.3 (usvp)
BDGL16	150.7 (bdd)	217.5 (bdd)	300.8 (bdd)	171.3 (bdd)	236.1 (bdd)	308.0 (bdd)
CheNgu12	270.8 (bkw)	363.5 (bkw)	478.8 (bkw)	307.1 (bkw)	403.5 (bkw)	500.1 (bkw)
Kyber	154.4 (bdd)	218.7 (bdd)	299.2 (bdd)	174.2 (bdd)	236.6 (bdd)	306.1 (bdd)
MATZOV	147.7 (bdd)	211.5 (bdd)	291.3 (bdd)	167.3 (bdd)	229.3 (bdd)	298.1 (bdd)
GJ21	154.4 (bdd)	218.7 (bdd)	299.2 (bdd)	174.2 (bdd)	236.6 (bdd)	306.1 (bdd)
LaaMosPol14	139.3 (bdd)	200.0 (bdd)	275.5 (bdd)	158.1 (bdd)	216.9 (bdd)	282.0 (bdd)

Table 4: RLWE Cost on Known Cost Models Estimated by Lattice Estimator

metric. GJ21 cost model follows [26] which runs a sieve on the first β_0 vectors of the basis after BKZ-β reduction to produce many short vectors. Note that β_0 is chosen such that BKZ-β reduction and the sieve run in approximately the same time. MATZOV follows [35] and uses improved enumeration in list decoding.

Comparison with CRYSTALS-Dilithium. For comparison, we also provide cost estimates against the MLWE problem of CRYSTALS-Dilithium parameters [18, 39] in Table 5. At the security level I, our costs for the concrete parameter are comparable to those of CRYSTALS-Dilithium, but at the other security levels, our costs are higher than those of CRYSTALS-Dilithium. Obviously, in the conservative parameters, our costs are higher than those of CRYSTALS-Dilithium at all the security levels.

Cost model/Security Level	2 (I)	3(III)	5(V)
ABFKSW20	261.0 (usvp)	363.4 (bkw)	454.7 (bkw)
ABLR21	231.1 (usvp)	363.0 (usvp)	454.7 (bkw)
ADPS16	123.8 (usvp)	182.5 (usvp)	252.0 (usvp)
BDGL16	151.2 (bdd)	209.7 (bdd)	279.6 (bdd)
CheNgu12	270.9 (bkw)	363.4 (bkw)	454.7 (bkw)
Kyber	154.8 (bdd)	211.1 (bdd)	278.7 (bdd)
MATZOV	148.1 (bdd)	204.0 (bdd)	271.0 (bdd)
GJ21	154.8 (bdd)	211.1 (bdd)	278.7 (bdd)
LaaMosPol14	139.7 (bdd)	192.8 (bdd)	256.3 (bdd)

Table 5: MLWE Cost of on Known Cost Models Estimated by Lattice Estimator

3.5 Cyclotomic Trinomial Counterpart

Our scheme supports a cyclotomic trinomial for better performance. For it, we use the cyclotomic trinomial, $\phi(X) = X^n - X^{n/2} + 1$, and power-of-two modulus $q = 2^{23}$ instead of $X^p - X + 1$ and prime modulus, respectively. We use the degree of the polynomial of the form $2^a 3^b$ for flexible choices of parameters. Possible degrees of the polynomial of the form $2^a 3^b$ between 512 and 2000 are 512, 576, 648, 729, 768, 864, 972, 1024, 1152, 1296, 1458, 1536, 1728, and 1944. We use 1024, 1458, and 1944. The concrete parameter sets based on security analysis similar to the non-cyclotomic case are presented in Table 6.

Parameter/Security Level	I	III	V
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	2
β	100	116	128
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	4.15
Public key size	1440	2037	2462
Secret key size	2400	3377	4713
signature size	2529	3678	5135
Cost to SIS (BKZ β)	130.9 (411)	203.6 (658)	260.9 (853)
Quantum cost to SIS	114.4	180.1	232.0
Cost to LWE by estimator (BKZ β)	148.1 (414)	216.1 (657)	296.4 (943)
Quantum cost to LWE	123.3	186.2	260.4

Table 6: Parameters for NCC-Sign using Cyclotomic Trinomials.

4 Implementation Details

We describe implementation details of our scheme. 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 same bit packing method in [18, 39].

4.1 Optimizations of Hashing to a Ball

We chose the challenge polynomial $\mathbf{c} \in \mathcal{R}$ having τ non-zero coefficients. For the optimization, we could choose $\mathbf{c} \in R = \mathbb{Z}[X]/(X^p - X - 1)$ differently, namely, choose two (or more) separate polynomials.

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 $|t_j| \le \eta$ for all j. For $i = 1$, it can be seen that $|t_j| \le \eta$ for all j except that $|t_1| \le 2\eta$. For $i = 2$, it can also be seen that $|t_j| \le \eta$ for all j but $j = 1, 2$ where $|t_1|, |t_2| \le 2\eta$. Similarly, for $\mathbf{t} = \mathbf{s} \cdot X^i$, it can be seen that $|t_j| \le \eta$ for all j except $j = 1, 2, ..., i$. Thus, for $i < p_2$, $|t_j| \le \eta$ for $j \ge p_2$ and $|t_j| \le 2\eta$ 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 $|t_j| \le \tau_2\eta$ for $j \ge p_2$, and $|t_j| \le 2\tau_2\eta$ 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 $|u_j| \le (2\tau_1 + \tau_2)\eta$ for $j \ge p_2$, and $|u_j| \le 2(\tau_1 + \tau_2)\eta$ 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 $j \ge p_2$ 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 7, we can see that this optimization offers speed-up ranging from 9% to 24%, depending on the parameter sets.

Parameter	p	τ	κ	p_1, p_2	τ_1, τ_2	Exp.reps. (new)	Speed-up
I	1021	25	190	510,511	104,76	5.44	1.21
III	1429	29	228	714,715	120,88	4.76	1.19
V	1913	32	259	956,957	128,96	4.42	1.24
I^c	1201	32	241	600,601	132,98	2.27	1.09
III^c	1607	32	254	803,804	132,98	2.7	1.11
V^c	2039	32	265	1019,1020	132,98	3.43	1.15

Table 7: Optimization effects for Our Parameter Sets.

4.2 Polynomial Multiplications

Algorithm 11: Toom-Cook Algorithm [17], [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 $y = \{0, \pm 1, \pm 0.5, 2, \infty\}$.

// 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 [17], [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 $y = \{0, \pm1, \pm0.5, 2, \infty\}$. 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 $y = \{0, \pm1, \pm0.5, 2, \infty\}$.

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 [17], [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`
$A(y) = A_3y^3 + A_2y^2 + A_1y + A_0$ // $y = x^{64}$
$B(y) = B_3y^3 + B_2y^2 + B_1y + B_0$ // $y = x^{64}$
// $A(y)B(y) = (A_3B_3)y^6 + (A_3B_2 + A_2B_3)y^5 + (A_3B_1 + A_2B_2 + A_1B_3)y^4 + (A_3B_0 + A_2B_1 + A_1B_2 + A_0B_3)y^3 + (A_2B_0 + A_1B_1 + A_0B_2)y^2 + (A_1B_0 + A_0B_1)y + (A_0B_0)$
$w_1 = A_3B_3$
$w_3 = A_2B_2$
$w_5 = A_1B_1$
$w_7 = A_0B_0$
$w_2 = (A_3 + A_2)(B_3 + B_2) - w_1 - w_3$
$w_6 = (A_1 + A_0)(B_1 + B_0) - w_5 - w_7$
$w_8 = (A_3 + A_1)(B_3 + B_1)$
$w_9 = (A_2 + A_0)(B_2 + B_0)$
$w_4 = (A_3 + A_2 + A_1 + A_0)(B_3 + B_2 + B_1 + B_0)$
$w_5 = w_5 + w_9 - w_7 - w_3$
$w_3 = w_3 + w_8 - w_1 - w_5$
$w_4 = w_4 - w_8 - w_9 - w_2 - w_6$
return $C(y) = w_1y^6 + w_2y^5 + w_3y^4 + w_4y^3 + w_5y^2 + w_6y + w_7$

Algorithm 13: Signed Montgomery Reduction ($\beta = 2^{32}$) [38]

Require: $0 < q < \frac{\beta}{2}$ odd, $-\frac{\beta}{2}q \le a = a_1\beta + a_0 < \frac{\beta}{2}q$ where $0 \le a_0 < \beta$
Ensure : $r' \equiv \beta^{-1}a \pmod q$, $-q < r' < q$
$m \leftarrow a_0q^{-1} \bmod {}^{\pm}\beta$
$t_1 \leftarrow \lfloor \frac{mq}{\beta} \rfloor$
$r' \leftarrow a_1 - t_1$

4.3 Modular Reductions

Our scheme performs polynomial multiplications over the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^p - X - 1)$. Using Montgomery reduction [36], 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 $[\frac{-q+1}{2}, \frac{q-1}{2}]$.

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 CYSTALS-Dilithium [18, 39], 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.

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: Type I Modulus.

4.4 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 our conservative parameters is faster than that of the concrete parameters. The conservative parameters use q's with bigger size, which lead the smaller number of expected repetitions in the rejection sampling. Our concrete parameters and conservative parameters can be considered as optimized for key/signature sizes and performance (in signing), respectively.

Our reference implementation uses SampleInBall algorithm in CRYSTALS-Dilithium [18, 39]. The new optimized SampleInBall algorithm and special forms of q will be used in our optimized implementation using AVX2.

Algorithm/Security Level	I	II	III
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 for Concrete Parameters at Three Security Levels

Algorithm/Security Level	I^c	III^c	V^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 for Conservative Parameters at Three Security Levels

5 Conclusion

In order to remove the structures that were the causes of the previous attacks, our scheme is the first lattice-based signature scheme using a prime-degree large Galois group inert modulus with $\phi(X) = X^p - X + 1$. We follow the design paradigm of CRYSTALS-Dilithium based on Bai and Galbraith scheme with public key compression. However, some critical distinctions exist between our scheme and CRYSTALS-Dilithium: our scheme is based on RLWE using non-cylcotomic polynomials instead of MLWE using the power-of-2 cylcotomic polynomial. The use of the non-cylcotomic polynomials leads to different selection of parameters and different implementation techniques. We also exploit a new optimized hashing to a ball using two separate polynomials. Consequently, our scheme provides stronger security guarantee than CRYSTALS-Dilithium and comparable key sizes and signature sizes.

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. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015)
5. 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)
6. 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)
7. 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)
8. 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)
9. 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)
10. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM (JACM) **50**(4), 506–519 (2003)
11. 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)
12. 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)
13. 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
14. Campbell, P., Groves, M., Shepherd, D.: Soliloquy: A cautionary tale. In: ETSI 2nd Quantum-Safe Crypto Workshop. vol. 3, pp. 1–9 (2014)
15. 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)
16. 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)

17. 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)
18. 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)
19. Eisenträger, K., Hallgren, S., Lauter, K.: Weak instances of PLWE. In: International Conference on Selected Areas in Cryptography. pp. 183–194. Springer (2014)
20. 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)
21. 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)
22. 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)
23. 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)
24. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Annual International Cryptology Conference. pp. 112–131. Springer (1997)
25. 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)
26. Guo, Q., Johansson, T.: Faster dual lattice attacks for solving LWE with applications to crystals. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 33–62. Springer (2021)
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. 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)
34. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
35. MATZOV: Report on the Security of LWE: Improved Dual Lattice Attack (Apr 2022). https://doi.org/10.5281/zenodo.6412487, https://doi.org/10.5281/zenodo.6412487
36. Montgomery, P.L.: Modular multiplication without trial division. Mathematics of computation **44**(170), 519–521 (1985)
37. 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)
38. Seiler, G.: Faster avx2 optimized ntt multiplication for ring-LWE lattice cryptography. Cryptology ePrint Archive (2018)
39. 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