

Distributed Key Generation for NTRU

Patrick Hough^{*1} , Jérôme Nguyen¹ ,
Caroline Sandsbråten² , and Tjerand Silde² 

¹ Universität der Bundeswehr München

{patrick.hough, jerome.nguyen}@unibw.de

² Norwegian University of Science and Technology

{caroline.sandsbraten, tjerand.silde}@ntnu.no

Abstract. NTRU-based encryption enjoys compact keys and ciphertexts and admits non-interactive distributed decryption, making it an attractive basis for threshold encryption with applications to threshold FHE, threshold signatures, and electronic voting. All known protocols, however, assume a secret key shared by a trusted dealer. The public NTRU key $h = f^{-1}g$ is a nonlinear function of the secret, so distributed key generation (DKG) techniques for LWE-based schemes do not apply, and generic MPC is prohibitively expensive.

We present the first dedicated DKG protocol for NTRU. Each party publishes an NTRU sample, defining a joint public key whose secret key is shared multiplicatively, and a multiplicative-to-additive (MtA) conversion yields the additive sharing required for non-interactive decryption. The protocol runs in few rounds and is actively secure with abort.

At the heart of our DKG lies the MtA conversion, for which we give two efficient lattice-based certified constructions; one from additively homomorphic NTRU encryption and one from homomorphic secret sharing, both of which may be of independent interest. We demonstrate the protocol by building a threshold variant of NTRU-Encrypt, which we prove secure and instantiate with concrete parameters.

Keywords: NTRU · Distributed KeyGen · Threshold Encryption

1 Introduction

Threshold encryption distributes the capability to decrypt among T parties so that no single party ever holds the full secret key. It is a core building block in constructions such as threshold fully homomorphic encryption [BGG⁺18], threshold signatures from threshold homomorphic encryption [GKS24, GHK⁺26], and cryptographic voting systems [HSS24]. It is also the subject of an ongoing NIST standardization effort on threshold schemes.³ Among lattice-based candidates, schemes built on the NTRU assumption are particularly attractive: keys and

^{*} This work was, in part, funded by the SCANDIUM (unibw.de/pacy-en/research/projects/scandium) and EU PiQASO projects.

³ <https://csrc.nist.gov/projects/threshold-cryptography>

ciphertexts are compact [PFH+22, BBC+20], operations are fast [LS19], and, crucially for the threshold setting, NTRU admits a *non-interactive* distributed decryption [HSS24]. Given an additive sharing $f = \sum_i \text{sk}_i$ of the secret key, each party publishes a single noise-drowned partial decryption, and the plaintext is recovered by summing the shares; no further interaction is required.

All such protocols, however, assume that the secret key is *already* secret-shared, implicitly relying on a trusted dealer. For LWE-based schemes, distributed key generation (DKG) is by now well understood: the public key $b = as + e$ is linear in the secret, so linear secret sharing essentially suffices, and a long line of work covers the passive, covert, and active settings [BHP17, AJL+12, DKL+13, RST+22, DDEK+23, GKS24, GHK+26]. The NTRU setting is fundamentally different: the public key $h = f^{-1}g$ is a *nonlinear* function of the secret, requiring a ring inversion and an invertibility check in $R_q = \mathbb{Z}_q[x]/(x^d + 1)$, neither of which is amenable to linear secret-sharing techniques. The only known route is generic MPC, as used by Rotaru et al. [RST+22] for BGV key generation. That protocol is already expensive and requires many rounds of communication for a linear key relation; evaluating an inversion in R_q inside generic MPC would be considerably worse. To the best of our knowledge, no dedicated DKG protocol for NTRU exists in the literature.

Our approach. Our starting point is the observation that NTRU keys *multiply*. If each party P_i samples its own NTRU pair (f_i, g_i) and publishes $h_i = f_i^{-1}g_i$, then $h = h_1h_2 = (f_1f_2)^{-1}(g_1g_2)$ is itself an NTRU public key whose secret key $f = f_1f_2$ is held *multiplicatively* by the two parties. Non-interactive distributed decryption, on the other hand, requires an *additive* sharing of f . We bridge this gap with a multiplicative-to-additive (MtA) protocol, which converts the shares f_1, f_2 into shares a_1, a_2 with $a_1 + a_2 = f_1f_2$. To achieve active security we introduce a *certified* MtA, which additionally guarantees that the input f_i is consistent with the published public key share h_i . This yields a DKG protocol with a small constant number of rounds.

1.1 Our Contribution

We provide the following contributions:

- **A DKG protocol for NTRU** (Section 4). We present a two-party DKG protocol in which each party broadcasts a commitment to its public key share $h_i = f_i^{-1}g_i$, opens it, and runs a certified MtA on f_i to obtain an additive share of $f = f_1f_2$. The protocol uses two broadcast rounds and a single MtA invocation; in practice the broadcast rounds can be run alongside the MtA rounds. It achieves active security with abort against a static, malicious adversary. We prove that the resulting threshold variant of NTRU-Encrypt is Th-IND-CPA-secure, with non-interactive distributed decryption in which each party sends a single message.
- **Two lattice-based MtA protocols** (Section 3). At the heart of our DKG lies the multiplicative-to-additive conversion, for which we formalize a certified

MtA functionality and give two efficient lattice-based constructions that may be of independent interest. The first, Π_{MtA1} , follows the blueprint of MtA from additively homomorphic encryption [GG18, LN18, CGG⁺20, XAX⁺21] instantiated with NTRU-Encrypt itself: it runs in two rounds, requires no interactive setup, and is secure under NTRU and RLWE given suitable zero-knowledge proofs. The second, Π_{MtA2} , builds on techniques from lattice-based homomorphic secret sharing [BKS19]: after a one-time interactive setup, each MtA invocation consists of a *single broadcast* per party, so the online phase is non-interactive.

- **Concrete parameters** (Section 6). We instantiate our protocols at $\lambda = 128$ bits of computational security and $\text{sec} = 40$ bits of statistical security. With noise-flooded decryption, this yields per-party communication under 700 KB, a public key and a ciphertext of 95 KB each (a single R_q element), and a decryption key share of 79 KB; an MPC-rounding decryption variant reduces these to roughly 170 KB, 6 KB and 6 KB, respectively (see Table 2 for more details). Furthermore, in Section 6 we conjecture that, via a conservative estimate, our design would reduce per-party communication by four orders of magnitude over state-of-the-art generic MPC techniques as presented in [RST⁺22], even if one were able to apply their techniques to NTRU.

We present our protocols in the two-party setting, matching the standard presentation of MtA-based protocols; scaling to more parties is left as future work.

1.2 Related Works

Distributed key generation for lattice-based encryption. For LWE-type schemes, DKG protocols exist with passive security in the full-threshold setting [BHP17, AJL⁺12], covert security via cut-and-choose [DKL⁺13], and active security from generic MPC in the full-threshold and honest-majority settings [RST⁺22, DDEK⁺23]. Arbitrary thresholds are achieved by Chowdhury et al. [CSS⁺22] assuming a trusted setup, and in [GKS24, GHK⁺26] using lattice-based Shamir secret sharing. None of these techniques applies to the nonlinear key relation of NTRU.

Distributed decryption. Threshold and distributed decryption for Ring/Module-LWE schemes has been studied extensively, from the early work of Bendlin and Damgård [BD10] to recent constructions based on noise drowning and Rényi-divergence arguments [ABGS23, HSS24, BS23b, BC25, MS24]. Our threshold scheme adopts the non-interactive distributed decryption of [HSS24]; our contribution is generating the required additive key sharing without a dealer.

NTRU and MtA. While most standardized lattice schemes are built from Module-LWE and Module-SIS [LS15, SAB⁺22, LDK⁺22, dPPRS23], with Threshold Racecon [DKM⁺24] as a notable threshold signature, the NTRU problem [HPS98, SS11, SXY18] and its module variants [FPS22, BJRW22] underlie compact signatures [CPS⁺20, BBJ⁺22] and pseudorandom functions [CS19]. MtA protocols are

a central tool in threshold ECDSA [DKLs24], instantiated classically from Paillier encryption or variants of oblivious transfer [GG18, LN18, CGG⁺20, XAX⁺21]; these operate over $\mathbb{Z}_q$, and we are not aware of a prior MtA tailored to the lattice setting.

2 Background

Notation. For a set S , distribution χ , and algorithm Alg , we let “ $\xleftarrow{\$} S$ ” denote uniform sampling from S and let “ $\leftarrow \chi$ ” and “ $\leftarrow \text{Alg}$ ” denote sampling according to χ and output according to a (randomized) algorithm Alg , respectively. The sign “ $:=$ ” denotes general assignment, and “ $=$ ” is used for logical equality only. Unless otherwise stated, logarithms are defined with respect to base 2.

Communication model. Throughout this work we assume 2 parties $\mathcal{P}_1, \mathcal{P}_2$ and consider static corruptions where the set of corrupted parties is fixed from the outset. Moreover, we work in the synchronous communication model in which parties have access to pairwise authenticated and private channels as well as a public broadcast channel. Finally, in the malicious setting, we consider a *rushing* adversary, meaning that in each round the corrupted parties will receive messages sent by all honest parties before sending their own messages.

The Ring R . Consider the ring $R := \mathbb{Z}[x]/(x^d + 1)$ for d a power of 2 and $R_q := \mathbb{Z}_q[x]/(x^d + 1)$ for q a prime. We denote elements of $\mathbb{Z}_q$, R and R_q by lower-case letters. Addition and multiplication of elements in R_q are implicitly reduced to the same ring throughout the paper.

We will additionally view elements $a \in R_q$ interchangeably as both polynomials and vectors in $\mathbb{Z}_q^d$ (in the coefficient embedding). Throughout the paper, we identify elements of $\mathbb{Z}_q$ with their unique centered representative in $\mathbb{Z} \cap (-q/2, q/2)$, and, coefficient-wise, extend this identification to $\mathbb{Z}_q^d$. For such elements, we can thus consider their ℓ_1 , ℓ_2 , and ℓ_∞ -norms. For an element $a \in R_q$, we may sometimes write $a \bmod p$ to mean that a ’s coefficients are reduced modulo p so that $(a \bmod p) \in R_p$. We will also regularly use the standard relations,

$$\|xy\|_\infty \leq \|x\|_2 \|y\|_2 \quad \text{and} \quad \|xy\|_2 \leq \sqrt{d} \|x\|_2 \|y\|_2. \quad (1)$$

2.1 Gaussians

Recall the standard discrete Gaussian distribution with support $\mathbb{Z}^d$, center 0, and standard deviation σ , defined by its probability mass function

$$D'_\sigma(z) := \frac{\rho_\sigma(z)}{\sum_{y \in \mathbb{Z}^d} \rho_\sigma(y)}, \quad \text{where} \quad \rho_\sigma(z) := \exp\left(\frac{-\|z\|_2^2}{2\sigma^2}\right).$$

In this paper, we will always use *truncated* Gaussian distributions, from which sampling yields d -dimensional integer vectors with bounded infinity and two-norms. We write D_σ to denote the truncated Gaussian distribution corresponding to D'_σ . An algorithm for sampling from a truncated Gaussian can be found in [HSS24].

2.2 Noise Drowning

For two probability distributions χ_0, χ_1 over a space S , we write

$$\Delta(\chi_0, \chi_1) := \frac{1}{2} \sum_{x \in S} |\Pr[\chi_0 = x] - \Pr[\chi_1 = x]|$$

for the statistical distance between χ_0 and χ_1 . We will make use of the following standard “smudging” lemma, that states that adding a sufficiently large noise to a small value “smudges” them out.

Lemma 1 ([AJL⁺12]). *Let $B_1 = B_1(\lambda)$ and $B_2 = B_2(\lambda)$ be positive integers and let $e_1 \in [-B_1, B_1]$ be a fixed integer. Let $e_2 \xleftarrow{\$} [-B_2, B_2]$ be an integer sampled from the uniform distribution. If $B_1/B_2 \leq \epsilon$, then*

$$\Delta(e_1 + e_2, e_2) \leq \epsilon.$$

2.3 Ring-LWE and NTRU

The security of our scheme relies on the well-known computational lattice problems Ring Learning with Errors (RLWE) [Reg05, LPR10], and NTRU [HPS98, LTV12, SXY18].

Definition 1 (Ring Learning with Errors). *Let $q > 2$ be a prime defining the ring R_q with dimension d , and let χ be a distribution over R_q . The advantage of a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ in solving the ring learning with errors problem $\text{RLWE}_{d,q,\chi}$ is defined as*

$$\text{Adv}_{d,q,\chi}^{\text{RLWE}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}_{\text{RLWE}}}(d, q, \chi) \rightarrow 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\mathbb{S}}}(d, q, \chi) \rightarrow 1]|,$$

where the oracles $\mathcal{O}_{\text{RLWE}}$ and $\mathcal{O}_{\mathbb{S}}$ are defined as follows:

- $\mathcal{O}_{\text{RLWE}}$: Samples $a \xleftarrow{\$} R_q$, $s, e \leftarrow \chi$, and outputs $(a, as + e)$;
- $\mathcal{O}_{\mathbb{S}}$: Samples $(a, b) \xleftarrow{\$} R_q \times R_q$, and outputs (a, b) .

Definition 2 (NTRU). *Let $q > 2$ be a prime defining the ring R_q with dimension d . Let D_f, D_g be bounded distributions over R_q . The advantage of a PPT algorithm $\mathcal{A}$ in solving the decisional $\text{NTRU}_{d,q,D_f,D_g}$ problem is defined as*

$$\text{Adv}_{d,q,D_f,D_g}^{\text{NTRU}}(\mathcal{A}) := |\Pr[\mathcal{A}^{\mathcal{O}_{\text{NTRU}}}(d, q, D_f, D_g) \rightarrow 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\mathbb{S}}}(d, q, D_f, D_g) \rightarrow 1]|,$$

where oracles $\mathcal{O}_{\text{NTRU}}$ and $\mathcal{O}_{\mathbb{S}}$ are defined as

- $\mathcal{O}_{\text{NTRU}}$: Samples $f \leftarrow D_f$ and $g \leftarrow D_g$, resamples if $f \notin R_q^\times$, and outputs $f^{-1}g \in R_q$;
- $\mathcal{O}_{\mathbb{S}}$: Samples $h \xleftarrow{\$} R_q$, and then outputs h .

Let $\sigma_{\text{NTRU}} \in \mathbb{R}$ and $D_{\sigma_{\text{NTRU}}}$ be a bounded distribution over R_q . In this work, we will use $D_g := D_{\sigma_{\text{NTRU}}}$. Depending on the context, we will use two different distributions for D_f . For some positive integer p , we define

$$D_f^p := p \cdot D_{\sigma_{\text{NTRU}}} + 1 \quad \text{and} \quad D_f^0 := D_{\sigma_{\text{NTRU}}}.$$

When it is clear from context, we will write $\text{NTRU}_{d,q,\sigma_{\text{NTRU}}}$ to denote both $\text{NTRU}_{d,q,D_f^p,D_g}$ and $\text{NTRU}_{d,q,D_f^0,D_g}$.

We are interested in the probability that $h \leftarrow \mathcal{O}_{\text{NTRU}}$ has an inverse. For certain parameter choices where $x^d + 1$ factors modulo q into $k < d$ distinct irreducible factors, it follows from [LS18] that sufficiently short non-zero elements are invertible. Hence, h has an inverse with overwhelming probability (with probability 1 if $g = 0$ is excluded). In our setting, we choose q such that $x^d + 1$ *completely* splits, i.e. it has $k = d$ distinct linear factors, and we instead use the following lemma.

Lemma 2. *Let q be a prime such that $x^d + 1$ completely splits. Let ϵ_{NTRU} be a bound on the distinguishing advantage of every PPT adversary in the $\text{NTRU}_{d,q,\sigma_{\text{NTRU}}}$ problem. Then*

$$\Pr_{h \leftarrow \mathcal{O}_{\text{NTRU}}} [h \notin R_q^\times] \leq d/q + \epsilon_{\text{NTRU}}. \quad (2)$$

Proof. Since $x^d + 1$ splits into d distinct linear factors, we have $\Pr_{h \leftarrow R_q} [h \notin R_q^\times] \leq 1 - (1 - 1/q)^d \leq d/q$ [Ros02, Proposition 1.7]. Consider the NTRU distinguisher $\mathcal{D}$ that outputs 1 on the input $h \notin R_q^\times$ and 0 otherwise. We have,

$$\begin{aligned} \left| \Pr_{h \leftarrow \mathcal{O}_{\text{NTRU}}} [h \notin R_q^\times] - \Pr_{h \leftarrow R_q} [h \notin R_q^\times] \right| &= \left| \Pr_{h \leftarrow \mathcal{O}_{\text{NTRU}}} [\mathcal{D}(h) \rightarrow 1] - \Pr_{h \leftarrow R_q} [\mathcal{D}(h) \rightarrow 1] \right| \\ &\leq \epsilon_{\text{NTRU}}. \end{aligned}$$

The claim follows. $\square$

2.4 Distributed Key Generation and Threshold Encryption

A public-key encryption scheme is a tuple of polynomial-time algorithms $\Pi = (\text{KeyGen}, \text{Enc}, \text{Dec})$ where:

- $\text{KeyGen}(1^\lambda)$ is a probabilistic algorithm that outputs a public key, secret key pair (pk, sk) .
- $\text{Enc}(\text{pk}, m)$ is a probabilistic algorithm that takes as input a public key pk and a plaintext m , and outputs a ciphertext c .
- $\text{Dec}(\text{sk}, c)$ is a deterministic algorithm that takes as input the secret key sk and a ciphertext c , and outputs a plaintext m .

We use the standard definitions of correctness and IND-CPA security, which we omit here; see Goldwasser and Micali [GM84]. We say that Π is a T -out-of- T threshold encryption scheme if it is extended with the following:

- DKG is an interactive protocol run by T parties. All parties output a decryption key share $\mathbf{sk}_i$ and a common public key $\mathbf{pk}$ or a failure symbol $\perp$.
- $\text{DDec}(\mathbf{sk}_i, c)$ is a probabilistic algorithm that takes as input a key share $\mathbf{sk}_i$ and a ciphertext, and outputs a partial decryption $\mathbf{ds}_i$.
- $\text{Comb}(\{\mathbf{ds}_i\})$ is a deterministic polynomial-time algorithm that takes as input the T partial decryptions $\mathbf{ds}_i$, and outputs a plaintext m .

Definition 3. We say that a T -out-of- T threshold encryption scheme Π is threshold indistinguishable if every PPT adversary $\mathcal{A}$ wins the experiment of Figure 1, written $\text{Exp}_{\Pi, \mathcal{A}}^{\text{Th-IND-CPA}}(1^\lambda)$, with probability at most $1/2 + \text{negl}(\lambda)$. We set

$$\text{Adv}_{\Pi}^{\text{Th-IND-CPA}}(\mathcal{A}) := |\Pr[b = b'] - 1/2|.$$

$\text{Exp}_{\Pi, \mathcal{A}}^{\text{Th-IND-CPA}}(1^\lambda)$
1. $\mathcal{A}$ outputs a set of corrupted parties $\mathcal{C}$ of size at most $T - 1$. Let $\mathcal{H} := [T] \setminus \mathcal{C}$. 2. $\mathcal{A}$ (controlling parties in $\mathcal{C}$) runs DKG with the parties in $\mathcal{H}$. If the protocol aborts, return a random bit. Else, this sets a key pair $(\mathbf{pk}, \mathbf{sk})$. 3. At any time after key generation, the adversary may make a threshold decryption query $(\mathcal{U}, (m, r) \in \mathcal{M} \times \mathcal{R})$: – Compute $c = \text{Enc}(\mathbf{pk}, m; r)$. – Let $\mathcal{H}_{\mathcal{U}} := \mathcal{H} \cap \mathcal{U}$. For $i \in \mathcal{H}_{\mathcal{U}}$, compute $\mathbf{ds}_i \leftarrow \text{DDec}(\mathbf{sk}_i, c)$. – Return the ciphertext and the decryption shares $\{\mathbf{ds}_i\}_{i \in \mathcal{H}_{\mathcal{U}}}$ to $\mathcal{A}$. 4. At some point, $\mathcal{A}$ outputs messages m_0 and m_1. Sample $b \xleftarrow{\\$} \{0, 1\}$ and return $\text{Enc}(\mathbf{pk}, m_b)$ to $\mathcal{A}$. 5. Eventually $\mathcal{A}$ returns b'. The adversary wins the game if $b = b'$.

Fig. 1. The indistinguishability experiment $\text{Exp}_{\Pi, \mathcal{A}}^{\text{Th-IND-CPA}}(1^\lambda)$.

2.5 Rounding Secret Shares

We will use the following two lemmas introduced in [BKS19] that quantify the behaviour of additive shares under reduction and lifting between moduli.

Lemma 3 (adapted from [BKS19] Lemma 1). Let $t, Q \in \mathbb{N}$ be moduli values with $Q/t \geq \lambda^{\omega(1)}$. Let $\tau_1, \tau_2 \in R_Q$ with $\tau_1 \xleftarrow{\$} R_Q$ and

$$\tau_1 + \tau_2 = m + te \bmod Q$$

for some $m \in R_t, e \in R$ with $Q/(t \cdot \|e\|_\infty) \geq \lambda^{\omega(1)}$. Then,

$$(\tau_1 \bmod t) + (\tau_2 \bmod t) = m \bmod t$$

holds with probability at least $1 - d \cdot (\|e\|_\infty + 1) \cdot t/Q \geq 1 - \lambda^{-\omega(1)}$ over the randomness of the shares τ_1, τ_2 .

Lemma 4 ([BKS19] Lemma 2). *Let $t \in \mathbb{N}$ be integers with $t \geq \lambda^{\omega(1)}$. Let $m \in R$ and $z_1, z_2 \in R_t$ with $z_1 \xleftarrow{\$} R_t$ and*

$$z_1 + z_2 = m \bmod t.$$

Then, with probability at least $1 - d(\|m\|_\infty + 1)/t \geq 1 - \lambda^{-\omega(1)}$ over the randomness of the shares z_1, z_2 , we have

$$z_1 + z_2 = m \text{ over } R.$$

2.6 NTRU Encryption

We describe in Figure 2 an encryption scheme $\text{NTRU-Encrypt} = (\text{KeyGen}_{\text{NTRU}}, \text{Enc}_{\text{NTRU}}, \text{Dec}_{\text{NTRU}})$ based on the NTRU and RLWE problems [HPS98].

Setup. Let $p \ll q$ be co-prime integers, defining the rings R_p and R_q for a fixed dimension d . Messages lie in R_p . Let $\sigma_{\text{NTRU}}, \sigma_{\text{LWE}} \in \mathbb{R}$. Define the setup parameters

$$\text{sp}_{\text{NTRU-Encrypt}} := (d, p, q, \sigma_{\text{NTRU}}, \sigma_{\text{LWE}}).$$

Key Generation $\text{KeyGen}_{\text{NTRU}}(\text{sp}_{\text{NTRU-Encrypt}})$.

Given input $\text{sp}_{\text{NTRU-Encrypt}} = (d, p, q, \sigma_{\text{NTRU}}, \sigma_{\text{LWE}})$:

1. Sample short terms $f, g \leftarrow D_{\sigma_{\text{NTRU}}}$.
2. If $f \notin R_p^\times$ or $g \notin R_q^\times$, resample.
3. Compute $h := f^{-1}g \in R_q$.
4. Return the secret key $\text{sk} := f$ and public key $\text{pk} := h$.

Encryption $\text{Enc}_{\text{NTRU}}(\text{pk}, m)$. Given public key $\text{pk} = h$ and message $m \in R_p$:

1. Sample randomness $s \leftarrow D_{\sigma_{\text{LWE}}}$ and noise term $e \leftarrow D_{\sigma_{\text{LWE}}}$.
2. Compute ciphertext $c := p(hs + e) + m \in R_q$.
3. Return ciphertext c .

Decryption $\text{Dec}_{\text{NTRU}}(\text{sk}, c)$. Given secret key $\text{sk} = f$ and ciphertext c :

1. Compute $m' := fc \in R_q$.
 2. Return message $m = (f^{-1} \cdot m' \bmod p) \in R_p$.
-

Fig. 2. The encryption scheme NTRU-Encrypt adapted from [HPS98, SS11, HSS24].

Lemma 5 ([SS11, Theorem 3.3]). *The NTRU-Encrypt scheme is IND-CPA-secure under the $\text{NTRU}_{d,q,\sigma_{\text{NTRU}}}$ and $\text{RLWE}_{d,q,D_{\sigma_{\text{LWE}}}}$ assumptions. In particular, for every PPT adversary $\mathcal{A}$ attacking the IND-CPA security of NTRU-Encrypt , there exist PPT distinguishers $\mathcal{B}_{\text{NTRU}}$ and $\mathcal{B}_{\text{RLWE}}$ against NTRU and RLWE respectively, such that*

$$\text{Adv}_{\text{NTRU-Encrypt}}^{\text{IND-CPA}}(\mathcal{A}) \leq \text{Adv}_{d,q,\sigma_{\text{NTRU}}}^{\text{NTRU}}(\mathcal{B}_{\text{NTRU}}) + \text{Adv}_{d,q,D_{\sigma_{\text{LWE}}}}^{\text{RLWE}}(\mathcal{B}_{\text{RLWE}}).$$

3 Lattice-Based MtA

We propose two lattice-based multiplicative-to-additive (MtA) protocols. We note that our ultimate goal is to convert *small* multiplicative shares m_1 and m_2 to *uniform* additive shares a_1 and a_2 .

Our first scheme takes inspiration from a line of works [CGG⁺20, LN18, XAX⁺21, GG18] making use of an additive homomorphic encryption scheme to construct a two-round, interactive MtA.

Our second scheme uses ideas from recent lattice-based homomorphic secret sharing (HSS) [BKS19]. HSS allows for local evaluation of secret shares. That is, homomorphic evaluation is distributed between two parties who do not interact with each other. We give an MtA based on techniques from HSS requiring only a single broadcast from each party. In contrast to our first design however, this scheme requires an initial interactive key generation procedure.

Preparation. In the rest of this section, we will use the following notation. Let $t \ll Q$ be co-prime integers and let d be a power of two, defining the rings R_t and R_Q . Let $D_{\sigma_{\text{NTRU}}}$ and $D_{\sigma_{\text{LWE}}}$ be discrete Gaussian distributions over R_Q with standard deviations σ_{NTRU} and σ_{LWE} respectively. We write B_{NTRU} and B_{LWE} (respectively $B_{\text{NTRU}}^{(\infty)}$ and $B_{\text{LWE}}^{(\infty)}$) to denote bounds on the two-norm (respectively the infinity norm) of elements sampled from $D_{\sigma_{\text{NTRU}}}$ and $D_{\sigma_{\text{LWE}}}$. We assume that the multiplicative shares (m_1, m_2) input to the MtA satisfy

$$\|m_1\|_2, \|m_2\|_2 \leq B_m.$$

Let NIZK be a non-interactive zero-knowledge argument of knowledge (NIZKAoK). We use standard definitions of completeness, knowledge soundness, and zero-knowledge. Let sp_{NIZK} denote its parameters. We make the relations proven with NIZK explicit in Section 5.

3.1 The Multiplicative-to-Additive Ideal Functionality

The ideal functionality $\mathcal{F}_{\text{MtA}}$ is defined for two parties $\mathcal{P}_1$ and $\mathcal{P}_2$ and presented in Figure 3. It is parameterized by a modulus t and a ring dimension d . Parties $\mathcal{P}_1$ and $\mathcal{P}_2$ participate in the Setup phase once, and run the Conversion phase as many times as they wish. On input $m_1, m_2 \in R_t$ from $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, $\mathcal{F}_{\text{MtA}}$ outputs $a_1, a_2 \in R_t$ such that

$$a_1 + a_2 = m_1 m_2 \in R_t.$$

Moreover, a_1 is sampled uniformly from R_t , and therefore both output shares are individually uniform over R_t .

We also introduce the notion of certified MtA protocol. In addition to the normal functionality, we require the certified protocol to output a public string $\text{cert} \in \{0, 1\}^*$. The certification string proves the validity of the input m_i to the protocol with respect to some relation and serves as an argument of knowledge of

$\mathcal{F}_{\text{MtA}}$
Setup: On receiving (setup) from $\mathcal{P}_1$ and $\mathcal{P}_2$: – Store and send (setup-complete) to $\mathcal{P}_1$ and $\mathcal{P}_2$. Conversion: On receiving (input, sid, $m_1 \in R_t$) from $\mathcal{P}_1$ and (input, sid, $m_2 \in R_t$) from $\mathcal{P}_2$, if (setup-complete) exists: – Sample $a_1 \xleftarrow{\\$} R_t$ and compute $a_2 := m_1 m_2 - a_1 \in R_t.$ – Send (output-1, sid, a_1) to $\mathcal{P}_1$. – Send (output-2, sid, a_2) to $\mathcal{P}_2$.

Fig. 3. Multiplicative-to-additive functionality $\mathcal{F}_{\text{MtA}}$.

the output a_i . Looking ahead, in Section 4, we will use the certification to ensure that the NTRU secret $f_i \in R_q$ that is used as input to the MtA corresponds to the NTRU partial public key $h_i \in R_q$.

Definition 4 (Certified MtA protocol). *We say that an MtA protocol Π_{MtA} is certified for a relation $\mathcal{R}_{\text{cert}}$, if*

1. Π_{MtA} securely computes $\mathcal{F}_{\text{MtA}}$ with abort in the presence of static malicious adversaries,
2. Each party has an additional input $y_i \in \mathcal{Y}$ from some set $\mathcal{Y}$ for $i \in \{1, 2\}$. For an execution of the protocol with input $((m_1, y_1), (m_2, y_2))$ and output (a_1, a_2) , all parties output a string $\text{cert} \in \{0, 1\}^*$ that contains zero-knowledge arguments of knowledge that $\mathcal{R}_{\text{cert}}(m_i, y_i) = 1$ and that $\mathcal{P}_i$ knows a_i .

By abuse of notation, we write $\text{Vf}(\text{cert}, \{y_1, y_2\})$ to denote using the verification algorithm on all proofs contained in cert and outputting 1 if and only if all proofs verify.

3.2 Our First Construction: Π_{MtA1}

Preparation. Let $\text{sec} = \text{sec}(\lambda)$ be a statistical security parameter. Let χ_{sec} be the uniform distribution over $\{x \in R_Q : \|x\|_{\infty} \leq b_{\text{MtA}}\}$, for some integer $b_{\text{MtA}} = b_{\text{MtA}}(\text{sec})$. Define the setup parameters:

$$\text{sp} := (d, t, Q, \sigma_{\text{NTRU}}, \sigma_{\text{LWE}}, \text{sec}, \text{sp}_{\text{NIZK}}).$$

Here we use the NTRU-Encrypt encryption scheme where plaintext elements live in R_t and ciphertext operations are executed in R_Q . This notation is chosen to allow for a third modulus level, denoted by p , in Section 4. We present our NTRU-Encrypt-based certified MtA protocol Π_{MtA1} in Figure 4.

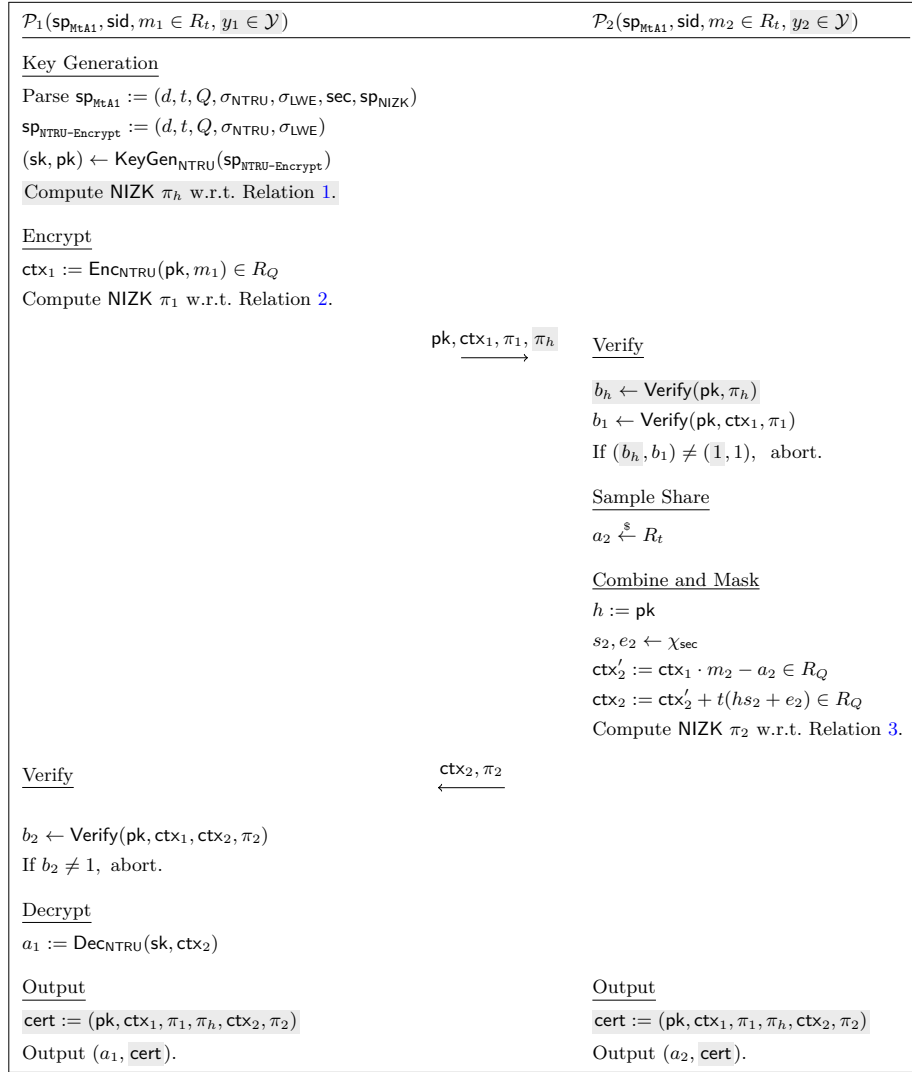


Fig. 4. The multiplicative-to-additive protocol Π_{MtA1} , built from additively homomorphic NTRU-Encrypt. The gray background indicates the parts that are only relevant for the certified MtA.

Lemma 6 (Correctness of Π_{MtA1}). *Let $\text{sp} = (d, t, Q, \sigma_{\text{NTRU}}, \sigma_{\text{LWE}}, \text{sec}, \text{sp}_{\text{NIZK}})$ be as defined above. Let $B_{\text{sec}} := \sqrt{d}b_{\text{MtA}}$. Then the Π_{MtA1} protocol correctly computes $\mathcal{F}_{\text{MtA}}$ if the proof system NIZK is complete and the following condition is satisfied:*

$$2t(\sqrt{d}B_{\text{NTRU}}B_{\text{LWE}}B_m + B_{\text{NTRU}}B_{\text{sec}}) + B_{\text{NTRU}}\sqrt{d}(B_m^2 + t/2) < Q/2. \quad (3)$$

Proof. Since NIZK is complete, all honestly generated proofs verify with overwhelming probability. It remains to show that ctx_2 decrypts correctly. It suffices to show that

$$\|f \cdot \text{ctx}_2\|_\infty < Q/2,$$

In the honest execution, ctx_2 is an encryption of $m_1 m_2 - a_2$. Writing the NTRU secret key as f and the public key as $h = f^{-1}g$, we have

$$\begin{aligned} f \cdot \text{ctx}_2 &= t(g(s_1 m_2 + s_2) + f \cdot (e_1 m_2 + e_2)) + f \cdot (m_1 m_2 - a_2) \\ &= t(gs_1 m_2 + gs_2 + fe_1 m_2 + fe_2) + f \cdot (m_1 m_2 - a_2). \end{aligned}$$

Here s_1, e_1 are sampled from $D_{\sigma_{\text{LWE}}}$, and s_2, e_2 are sampled from χ_{sec} . Since $a_2 \in R_t$ is represented coefficient-wise in the centered interval, we have $\|a_2\|_\infty \leq t/2$ and hence $\|a_2\|_2 \leq \sqrt{d}t/2$. Similarly, $\|s_2\|_2, \|e_2\|_2 \leq B_{\text{sec}}$.

Applying the triangle inequality together with Equation (1) gives the upper bound appearing on the left-hand side of Equation (3). By Equation (3), this is strictly smaller than $Q/2$. Therefore reduction modulo t gives

$$f \cdot (m_1 m_2 - a_2),$$

and ctx_2 decrypts correctly. Thus $\mathcal{P}_1$ obtains $a_1 := m_1 m_2 - a_2$, while $\mathcal{P}_2$ holds a_2 , and the two shares sum to $m_1 m_2$ in R_t . $\square$

Theorem 1 (Security of Π_{MtA1}). *Assume NTRU-Encrypt is an IND-CPA encryption scheme and NIZK is zero-knowledge and has knowledge soundness. Let $b_{\text{MtA}} := 2^{\text{sec}} dB_{\text{LWE}} B_m$.*

Then, Π_{MtA1} securely computes $\mathcal{F}_{\text{MtA}}$ with abort in the presence of static, malicious adversaries. For every PPT adversary $\mathcal{A}$ corrupting one party, there exists a PPT simulator $\mathcal{S}$ such that

$$\{\text{REAL}_{\Pi_{\text{MtA1}}, \mathcal{A}}\} \equiv_c \{\text{IDEAL}_{\mathcal{F}_{\text{MtA}}, \mathcal{S}}\}.$$

We prove Theorem 1 in two parts, corresponding to the two scenarios in which the adversary corrupts $\mathcal{P}_1$ or $\mathcal{P}_2$. For $i \in \{1, 2\}$, let $\text{REAL}_i, \text{IDEAL}_i$, be the view of $\mathcal{P}_i$ in the real and simulated execution respectively.

Proof of Theorem 1: Part 1. If $\mathcal{A}$ corrupts $\mathcal{P}_1$, then $\mathcal{S}$ simulates $\mathcal{P}_2$ as follows.

1. $\mathcal{S}$ receives $(h, \text{ctx}_1, \pi_1, \pi_h)$ from $\mathcal{A}$. If the proofs verify, $\mathcal{S}$ extracts m_1 using the extractor of NIZK. Otherwise it aborts.
2. $\mathcal{S}$ queries $\mathcal{F}_{\text{MtA}}$ with input $(\text{input}, \text{sid}, m_1)$ on behalf of $\mathcal{P}_1$ and receives a_1 as the output for $\mathcal{P}_1$.
3. $\mathcal{S}$ samples $s, e \leftarrow \chi_{\text{sec}}$, sets

$$\text{ctx}_2 := t(hs + e) + a_1,$$

and uses the zero-knowledge simulator to obtain π_2 .

4. $\mathcal{S}$ sends (ctx_2, π_2) to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs.

There are two differences between the simulation and the real execution. First, in the real execution we have

$$\text{ctx}_2 = \text{ctx}_1 m_2 - a_2 + t(hs_2 + e_2).$$

By the knowledge soundness of NIZK, the ciphertext ctx_1 is an encryption of the extracted value m_1 . Hence, in the real execution,

$$\text{ctx}_2 = t(h(s_1 m_2 + s_2) + e_1 m_2 + e_2) + m_1 m_2 - a_2.$$

By the definition of $\mathcal{F}_{\text{MtA}}$, we have

$$a_1 = m_1 m_2 - a_2.$$

Moreover, from Equation (1), we have

$$\|s_1 m_2\|_\infty, \|e_1 m_2\|_\infty \leq B_{\text{LWE}} B_m.$$

Therefore, it follows from Lemma 1 that,

$$\Delta((s_1 m_2 + s_2, e_1 m_2 + e_2), (s, e)) \leq 2^{-\text{sec}+1}.$$

Second, in the ideal execution π_2 is simulated. It follows from the zero-knowledge property of NIZK that this is indistinguishable from the real π_2 . Thus, $\mathcal{S}$ successfully simulates the view for the corrupted party $\mathcal{P}_1$. For every PPT distinguisher $\mathcal{D}_1$, its advantage is at most:

$$\text{Adv}_{\text{REAL}_1, \text{IDEAL}_1}^{\text{dist}}(\mathcal{D}_1) \leq \text{Adv}_{\text{NIZK}}^{\text{zk}}(\mathcal{B}_{\text{zk}}) + \text{Adv}_{\text{NIZK}}^{\text{ks}}(\mathcal{B}_{\text{ks}}) + 2^{-\text{sec}+1}.$$

□

Proof of Theorem 1: Part 2. Now consider the case that $\mathcal{A}$ corrupts $\mathcal{P}_2$ and $\mathcal{S}$ simulates $\mathcal{P}_1$.

1. $\mathcal{S}$ runs the key generation phase to obtain (h, π_h) . It then samples s_1, e_1 as in the protocol and sets

$$\text{ctx}_1 := t(hs_1 + e_1).$$

Finally, it uses the zero-knowledge simulator to obtain π_1 . It then sends $(h, \text{ctx}_1, \pi_1, \pi_h)$ to $\mathcal{A}$.

2. $\mathcal{S}$ receives (ctx_2, π_2) from $\mathcal{A}$. If π_2 verifies, $\mathcal{S}$ extracts m_2 from it. Otherwise it aborts.
3. $\mathcal{S}$ queries $\mathcal{F}_{\text{MtA}}$ with input $(\text{input}, \text{sid}, m_2)$ on behalf of $\mathcal{P}_2$ and receives a_2 .
4. $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs and halts.

In the real execution, ctx_1 encrypts m_1 , whereas in the simulation it encrypts 0. By the IND-CPA security of NTRU-Encrypt, these ciphertexts are computationally indistinguishable. Moreover, by the zero-knowledge property of NIZK, the simulated proof π_1 is indistinguishable from a real proof.

Second, by the knowledge soundness of NIZK, the ciphertext sent by the corrupted $\mathcal{P}_2$ is well formed with respect to the extracted value m_2 . Therefore, $\mathcal{S}$ queries the ideal functionality with the correct input. This completes the proof. For every PPT distinguisher $\mathcal{D}_2$, its advantage is at most:

$$\text{Adv}_{\text{REAL}_2, \text{IDEAL}_2}^{\text{dist}}(\mathcal{D}_2) \leq \text{Adv}_{\text{NTRU-Encrypt}}^{\text{IND-CPA}}(\mathcal{B}_{\text{enc}}) + \text{Adv}_{\text{NIZK}}^{\text{zk}}(\mathcal{B}_{\text{zk}}) + \text{Adv}_{\text{NIZK}}^{\text{ks}}(\mathcal{B}_{\text{ks}}).$$

□

Corollary 1. Π_{MtA1} is certified for $\mathcal{R}_{\text{cert}}$.

We sketch the proof. It follows immediately from the knowledge soundness of NIZK that $\mathcal{R}_{\text{cert}}(m_i, y_i) = 1$ for $i \in \{1, 2\}$ and that there exists an extractor such that $a_2 \leftarrow \text{extract}(\text{ctx}_2, \pi_2)$. To extract a_1 , extract $f \leftarrow \text{extract}(\text{pk}, \pi_h)$. Then compute a_1 by decrypting ctx_2 .

3.3 Our Second Construction: Π_{MtA2}

Preparation. Let $\text{PRF} : \{0, 1\}^* \rightarrow R_t^3$ be a pseudorandom function. Define the setup parameters:

$$\text{sp}_{\text{MtA2}} := (d, t, Q, \sigma_{\text{LWE}}, \text{PRF}, \text{sp}_{\text{NIZK}}).$$

Our protocol Π_{MtA2} is based on an RLWE distributed key generation protocol which we describe in Figure 5. We present Π_{MtA2} in Figure 6.

Distributed Key Generation $\text{DKG}_{\text{RLWE}}(\text{sp}_{\text{Th-RLWE}})$.

Given input $\text{sp}_{\text{Th-RLWE}} = (d, t, Q, \sigma_{\text{LWE}}, \text{PRF}, \text{sp}_{\text{NIZK}})$:

1. Sample a common uniform element $a \xleftarrow{\$} R_Q$ and a PRF key K .
 2. Each party $\mathcal{P}_i$ samples short terms $s_i, e_i \leftarrow D_{\sigma_{\text{LWE}}}$ and computes $b_i := as_i + te_i \in R_Q$.
 3. Each party $\mathcal{P}_i$ computes NIZK proof $\pi_{\text{pk}}^{(i)}$ with respect to Relation 6.
 4. Party $\mathcal{P}_i$ broadcasts $(b_i, \pi_{\text{pk}}^{(i)})$ and outputs $\text{dk}_i := (s_i, K)$.
 5. Return the public key $\text{pk} := (a, b := b_1 + b_2 \in R_Q)$.
-

Fig. 5. Distributed key generation DKG_{RLWE} used for Π_{MtA2} . The gray background indicates parts that are only relevant for the certified MtA.

Theorem 2 (Correctness Π_{MtA2}). *Let $s = s_1 + s_2$ and assume that the proof system NIZK is complete. Let ϵ_{PRF} be a bound on the advantage of any PPT adversary against PRF. Define*

$$B_0 := 2B_{\text{LWE}}, \quad B_1 := 4B_{\text{LWE}}^2 + B_{\text{LWE}}, \quad B_2 := 4\sqrt{d}B_{\text{LWE}}^2B_m + B_{\text{LWE}}B_m.$$

Let $\delta_0, \delta_1, \delta_2 \in (0, 1)$ satisfy

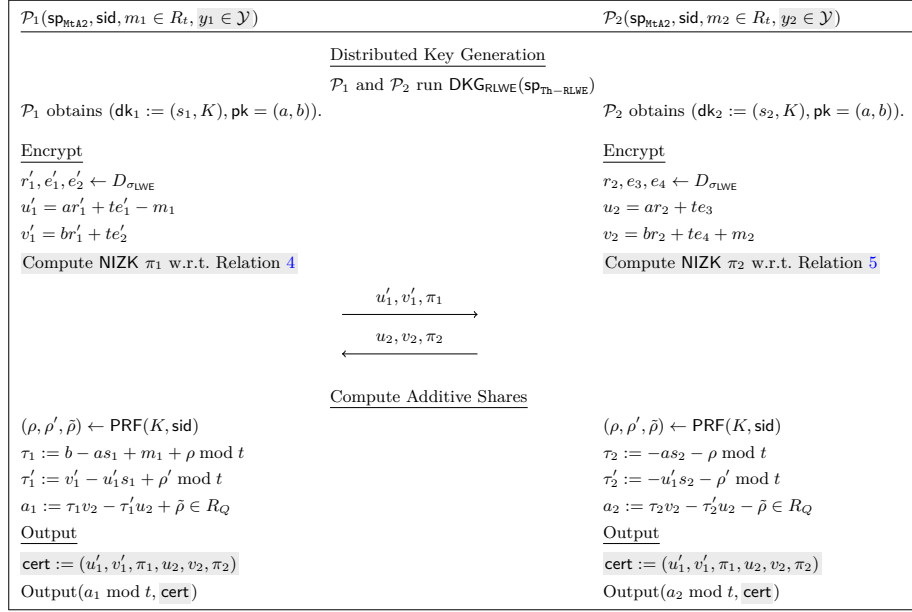


Fig. 6. Multiplicative-to-additive protocol Π_{MtA2} from RLWE secret-share rounding. The gray background indicates parts that are only relevant for the certified MtA.

$$1 - d(B_i + 1) \frac{t}{Q} \geq \delta_i \quad \text{for } i \in \{0, 1, 2\}.$$

Then Π_{MtA2} outputs additive shares $a_1, a_2 \in R_t$ such that

$$a_1 + a_2 = m_1 m_2 \in R_t \tag{4}$$

with probability at least $\delta_0 \delta_1 \delta_2 - \epsilon_{\text{PRF}}$.

Proof of Theorem 2. Since NIZK is complete, all honestly generated proofs verify with overwhelming probability. As in Boyle et al. [BKS19, Lemma 5], we prove correctness via a hybrid argument. Let ϵ_0 be the probability that Equation (4) holds at the end of protocol Π_{MtA2} . Define Π'_{MtA2} to be exactly the same protocol as Π_{MtA2} , with PRF replaced by a true random function. Let ϵ_1 be the success probability for that protocol. We have:

$$|\epsilon_0 - \epsilon_1| \leq \epsilon_{\text{PRF}}. \tag{5}$$

We now show that the success probability in Π'_{MtA2} is at least $\delta_0 \delta_1 \delta_2$. Recall that $s = s_1 + s_2$, $b = as + te$ with $e = e_1 + e_2$, and

$$u_2 = ar_2 + te_3, \quad v_2 = br_2 + te_4 + m_2.$$

We have,

$$\begin{aligned}\tau_1 + \tau_2 &= te + m_1 \bmod Q, \\ \text{and } \tau'_1 + \tau'_2 &= m_1 s + t(er'_1 + e'_2 - e'_1 s) \bmod Q.\end{aligned}$$

It follows from Equation (1) that,

$$\|er'_1 + e'_2 - e'_1 s\|_\infty \leq 4B_{\text{LWE}}^2 + B_{\text{LWE}}.$$

Since ρ, ρ' are sampled from the uniform distribution, it follows from Lemma 3 that, with probability at least $\delta_0 \delta_1$,

$$(\tau_1 \bmod t) + (\tau_2 \bmod t) = m_1 \bmod t, \quad (\tau'_1 \bmod t) + (\tau'_2 \bmod t) = m_1 s \bmod t. \quad (6)$$

When used below, we view $\tau_1, \tau_2, \tau'_1, \tau'_2 \in R_t$ as their canonical lifts to R_Q . Hence, assuming Equation (6) holds,

$$\begin{aligned}a_1 + a_2 &= (\tau_1 + \tau_2)v_2 - (\tau'_1 + \tau'_2)u_2 \bmod Q \\ &= m_1 m_2 + t(er_2 + e_4 - e_3 s)m_1 \bmod Q.\end{aligned}$$

By Equation (1),

$$\|(er_2 + e_4 - e_3 s)m_1\|_\infty \leq 4\sqrt{d}B_{\text{LWE}}^2 B_m + B_{\text{LWE}} B_m.$$

Again, $\tilde{\rho}$ is sampled from the uniform distribution. Following Lemma 3 shows that, with probability at least $\delta_0 \delta_1 \delta_2$,

$$(a_1 \bmod t) + (a_2 \bmod t) = m_1 m_2 \in R_t.$$

□

Theorem 3 (Security of Π_{MtA2}). *The protocol Π_{MtA2} securely computes $\mathcal{F}_{\text{MtA}}$ with abort in the presence of static, malicious adversaries under the RLWE assumption. For every PPT adversary $\mathcal{A}$ corrupting one party, there exists a PPT simulator $\mathcal{S}$ such that*

$$\{\text{REAL}_{\Pi_{\text{MtA2}}, \mathcal{A}}\} \equiv_c \{\text{IDEAL}_{\mathcal{F}_{\text{MtA}}, \mathcal{S}}\}.$$

We prove Theorem 3 in two parts, corresponding to the two scenarios in which the adversary corrupts $\mathcal{P}_1$ and $\mathcal{P}_2$. For $i \in \{1, 2\}$, let $\text{REAL}_i, \text{IDEAL}_i$, be the view of $\mathcal{P}_i$ in the real and simulated execution respectively.

Proof of Theorem 3: Part 1. The simulator $\mathcal{S}$ receives system parameters from $\mathcal{A}$ in the Setup phase. If $\mathcal{A}$ corrupts $\mathcal{P}_1$, then $\mathcal{S}$ simulates $\mathcal{P}_2$ as follows.

1. $\mathcal{S}$ runs DKG_{RLWE} with $\mathcal{A}$. It broadcasts the public key share $b_2 \xleftarrow{\$} R_Q$. Then, DKG_{RLWE} outputs the public key pk .
2. $\mathcal{S}$ receives (u'_1, v'_1, π_1) from $\mathcal{A}$. If π_1 verifies, $\mathcal{S}$ extracts the witness using the extractor of NIZK and obtains m_1 . Otherwise it aborts.

3. $\mathcal{S}$ queries $\mathcal{F}_{\text{MtA}}$ with input $(\text{input}, \text{sid}, m_1)$ on behalf of $\mathcal{P}_1$ and receives a_1 .
4. $\mathcal{S}$ samples $u_2, v_2 \xleftarrow{\$} R_Q$ and uses the zero-knowledge simulator to obtain π_2 .
5. $\mathcal{S}$ sends $((u_2, v_2), \pi_2)$ to $\mathcal{A}$ and outputs the same value as $\mathcal{A}$ outputs.

It follows from the RLWE assumption that $b_2 \xleftarrow{\$} R_Q$ sent by $\mathcal{S}$ is indistinguishable from the one sent in the real execution. Similarly for the tuple $((u_2, v_2), \pi_2)$, under the RLWE assumption and the zero-knowledge property of NIZK.

Moreover, by the knowledge soundness of NIZK, u'_1, v'_1 is well-formed with respect to the extracted m_1 , and $\mathcal{S}$ queries $\mathcal{F}_{\text{MtA}}$ with the correct value. Combining the above, we obtain that the distinguishing advantage in this case is bounded by

$$\text{Adv}_{\text{REAL}_1, \text{IDEAL}_1}^{\text{dist}}(\mathcal{D}_1) \leq 2\text{Adv}_{d, Q, \sigma_{\text{LWE}}}^{\text{RLWE}}(\mathcal{B}_{\text{RLWE}}) + \text{Adv}_{\text{NIZK}}^{\text{zk}}(\mathcal{B}_{\text{zk}}) + \text{Adv}_{\text{NIZK}}^{\text{ks}}(\mathcal{B}_{\text{ks}}).$$

□

Proof of Theorem 3: Part 2. Now consider the case that $\mathcal{A}$ corrupts $\mathcal{P}_2$ and $\mathcal{S}$ simulates $\mathcal{P}_1$.

1. $\mathcal{S}$ runs DKG_{RLWE} with $\mathcal{A}$. It broadcasts the public key share $b_1 \xleftarrow{\$} R_Q$. Then, DKG_{RLWE} outputs the public key pk .
2. $\mathcal{S}$ samples $u'_1, v'_1 \xleftarrow{\$} R_Q$ and uses the zero-knowledge simulator to obtain π_1 . $\mathcal{S}$ sends (u'_1, v'_1, π_1) to $\mathcal{A}$.
3. $\mathcal{S}$ receives (u_2, v_2, π_2) from $\mathcal{A}$. If π_2 verifies, $\mathcal{S}$ extracts m_2 using the extractor of NIZK. Otherwise it aborts.
4. $\mathcal{S}$ queries $\mathcal{F}_{\text{MtA}}$ with input $(\text{input}, \text{sid}, m_2)$ on behalf of $\mathcal{P}_2$ and receives a_2 .
5. $\mathcal{S}$ outputs the same value as $\mathcal{A}$ outputs.

As in the first part, it follows from the RLWE assumption, the zero-knowledge property and the knowledge soundness of NIZK that for any PPT distinguisher $\mathcal{D}_2$, the distinguishing advantage is bounded by:

$$\text{Adv}_{\text{REAL}_2, \text{IDEAL}_2}^{\text{dist}}(\mathcal{D}_2) \leq 2\text{Adv}_{d, Q, \sigma_{\text{LWE}}}^{\text{RLWE}}(\mathcal{B}_{\text{RLWE}}) + \text{Adv}_{\text{NIZK}}^{\text{zk}}(\mathcal{B}_{\text{zk}}) + \text{Adv}_{\text{NIZK}}^{\text{ks}}(\mathcal{B}_{\text{ks}}).$$

□

Corollary 2. *The protocol Π_{MtA2} is certified for $\mathcal{R}_{\text{cert}}$.*

We sketch the proof. It follows immediately from the knowledge soundness of NIZK that $\mathcal{R}_{\text{cert}}(m_i, y_i) = 1$ for $i \in \{1, 2\}$. To extract a_1 , extract $s_1 \leftarrow \text{extract}(\pi_{\text{pk}}^{(1)})$. Then, compute τ_1 and τ'_1 using $u'_1, v'_1 \in \text{cert}$ and s_1 as in the protocol. Finally, compute a_1 from τ_1, τ'_1 and $(v_2, u_2) \in \text{cert}$. The share a_2 can be extracted in the same way.

4 Our DKG for NTRU

In this section, we present a two-party DKG protocol for NTRU keys. We then use it to give an Th-IND-CPA-secure threshold version of the NTRU-Encrypt scheme.

4.1 MtA-based DKG

At a high level, each party publishes an NTRU sample $h_i = f_i^{-1}g_i \bmod q$. The public key is $h = h_1h_2$. The parties then run a certified MtA protocol on input f_i to obtain additive shares a_i of $f = f_1f_2$. In order to save on communication cost, we take advantage of the fact that $\|f\|_\infty \ll q$. We run the MtA protocol over a smaller modulus $t < q$, and then mod-switch the output a_i up to q .

Preparation. Let t be an integer and let $p \ll q$ be primes, such that $x^d + 1$ is completely split modulo q . Let $\mathcal{R}_{\text{DKG}} : R_q^2 \rightarrow \{0, 1\}$ be a relation such that $\mathcal{R}_{\text{DKG}}(a, b) = 1$ if and only if:

$$a \in R_q^\times \wedge \|a \cdot b\|_2 \leq B_{\text{NTRU}} \wedge a \equiv 1 \bmod p \wedge \|a\|_2 \leq pB_{\text{NTRU}} + 1.$$

Let Π_{MtA} be a certified MtA over R_t for relation $\mathcal{R}_{\text{DKG}}$. For $i \in \{1, 2\}$, let $j := 3 - i$. The two-party protocol DKG_{NTRU} is executed by each party $\mathcal{P}_i$ as follows:

1. Sample $f'_i, g_i \leftarrow D_{\sigma_{\text{NTRU}}}$ and set $f_i := pf'_i + 1$. If $f_i \notin R_q^\times$, resample. Sample $r_i \xleftarrow{\$} R_t$. Compute $h_i := f_i^{-1}g_i$, $\delta_i := H(i, h_i, r_i)$ and broadcast δ_i .
2. Broadcast (h_i, r_i) . Upon receiving (h_j, r_j) , check that $\delta_j = H(j, h_j, r_j)$ and $h_j \in R_q^\times$. Abort if either check fails.
3. Let $[f_i]_t := f_i \bmod t$. Run the Π_{MtA} protocol with $\mathcal{P}_j$ on input $([f_i]_t, h_i)$, obtaining (a_i, cert) or aborting. If $\text{Vf}(\text{cert}, \{h_1, h_2\}) \neq 1$, abort.
4. Compute $h = h_1h_2$. Set the public-key $\text{pk} := h$ and the secret-key share $\text{sk}_i := a_i + (-1)^{i-1}(r_1 + r_2) \bmod t$.

The protocol implicitly defines $f := f_1f_2$ and $g := g_1g_2$, where the combined secret key is $\text{sk} := f$. Note that we choose parameters such that $\|f_i\|_\infty < t/2$. Hence, by abuse of notation, we regard $[f_i]_t$ as an element of R_q for certification. In the next section, we will use the following bounds to prove correctness and security of the Th-NTRU scheme:

Lemma 7. *Let f and g be as defined above. Then $\|g\|_2 \leq B_g$, $\|f\|_2 \leq B_f$ and $\|f\|_\infty \leq B_f^{(\infty)}$, where*

$$\begin{aligned} B_f^{(\infty)} &:= p^2 B_{\text{NTRU}}^2 + 2p B_{\text{NTRU}}^{(\infty)} + 1 \\ B_g &:= \sqrt{d} B_{\text{NTRU}}^2 \quad \text{and} \quad B_f := \sqrt{d} p^2 B_{\text{NTRU}}^2 + 2p B_{\text{NTRU}} + 1 \end{aligned} \tag{7}$$

The protocol Π_{MtA} provides additive shares of f over R_t . However, our NTRU keys live over R_q . Since the shared value f is small, we use the modulus switching technique from [BKS19] to show that given the right parameters, the shares over the modulus t are in fact shares over the whole ring.

Lemma 8. *Let δ_{MtA} be the correctness probability of Π_{MtA} , and assume that the parameters are such that Equation (9) holds. Then DKG_{NTRU} completes with probability at least*

$$(1 - 2(d/q + \epsilon_{\text{NTRU}})),$$

where ϵ_{NTRU} is a bound on the advantage of any PPT adversary against the NTRU problem. Moreover, conditioned on completion, let $f, \text{sk}_1, \text{sk}_2$ be as defined above with $\|f_1\|_\infty, \|f_2\|_\infty < t/2$. Then, $\text{sk}_1 + \text{sk}_2 = f \bmod q$ with probability at least

$$\delta_{\text{DKG}} \geq (1 - d(B_f^{(\infty)} + 1)/t)\delta_{\text{MtA}}, \quad (8)$$

over the randomness of Π_{MtA} and r_1, r_2 .

Proof. We choose q such that $x^d + 1$ completely splits. By Lemma 2, Step 2 aborts with probability at most $2(d/q + \epsilon_{\text{NTRU}})$. Conditioned on the event that Π_{MtA} is correct, we have that a_1, a_2 satisfy:

$$a_1 + a_2 = [f_1]_t [f_2]_t \bmod t.$$

Recall that $f = f_1 f_2 \bmod q$. Since $\|f_1\|_\infty, \|f_2\|_\infty < t/2$ we have $[f_1]_t [f_2]_t = f_1 f_2$ over R . It follows that

$$\text{sk}_1 + \text{sk}_2 = f \bmod t.$$

Recall that the key shares sk_i are uniform over R_t . It follows from Lemma 4 that $\text{sk}_1 + \text{sk}_2 = f$ over R with probability at least $(1 - d(B_f^{(\infty)} + 1)/t)$, over the randomness of r_1, r_2 . In particular, $\text{sk}_1 + \text{sk}_2 = f \bmod q$. $\square$

4.2 Threshold NTRU-Encrypt

We present a threshold version of the NTRU-Encrypt encryption scheme in Figure 7, based on DKG_{NTRU} from the previous section and the threshold decryption from [HSS24].

Let $\text{sec} = \text{sec}(\lambda)$ be a statistical security parameter. Let χ_{sec}^* be the uniform distribution over $\{x \in R_q : \|x\|_\infty \leq b^*\}$ (distinct from the distribution χ_{sec} of Section 3.2, which lives over R_Q), for some integer

$$b^* := 2^{\text{sec}} d(B_{\text{LWE}}(B_g + B_f) + (B_f + 1)\sqrt{d}/2).$$

Lemma 9. *Let the parameters be such that Equation (9) holds. Then Th-NTRU is correct. More precisely, let the keys $\text{pk}, f, \text{sk}_1, \text{sk}_2$ be generated through DKG_{NTRU} . Conditioned on $\text{sk}_1 + \text{sk}_2 = f \bmod q$, for all $m \in R_p$, we have*

$$\Pr \left[\text{Comb}(\{\text{ds}_1, \text{ds}_2\}) = m \mid \begin{array}{l} c = \text{Enc}_{\text{ThNTRU}}(\text{pk}, m) \\ \{\text{ds}_i = \text{DDec}(\text{sk}_i, c)\}_{i \in \{1, 2\}} \end{array} \right] = 1.$$

Proof. Let $\text{pk} = h$ be the output of DKG_{NTRU} and $f = \text{sk}_1 + \text{sk}_2 \bmod q$. A ciphertext $c \leftarrow \text{Enc}_{\text{ThNTRU}}(\text{pk}, m)$ decrypts correctly if $\|fc + p(\tilde{e}_1 + \tilde{e}_2)\|_\infty < q/2$. Recall that $c = p(hs + e) + m$ and $h = f^{-1}g$. We have $fc = p(gs + fe) + fm$ and $\|\tilde{e}_1\|_\infty, \|\tilde{e}_2\|_\infty \leq b^*$. Using the triangle inequality, Equation (1), Lemma 7 and $\|m\|_2 \leq \sqrt{d}p/2$, it follows that c decrypts correctly if the following equation holds:

$$pB_{\text{LWE}}(B_g + B_f) + B_f \sqrt{d}p/2 + 2pb^* < q/2. \quad (9)$$

The claim follows. $\square$

Key Generation $\text{DKG}_{\text{NTRU}}(\text{sp}_{\text{Th-NTRU}})$.

Given input $\text{sp}_{\text{Th-NTRU}} = (d, p, t, q, \sigma_{\text{NTRU}}, \sigma_{\text{LWE}}, \text{sec}, \text{sp}_{\text{MtA}})$:

1. The two parties run the protocol DKG_{NTRU} of Section 4 on input $\text{sp}_{\text{Th-NTRU}}$.
2. Party $\mathcal{P}_i$ gets secret key share sk_i for $i \in \{1, 2\}$.
3. Return the public key $\text{pk} := h$.

Encryption $\text{Enc}_{\text{ThNTRU}}(\text{pk}, m)$. Given public key $\text{pk} = h$ and message $m \in R_p$:

1. Sample randomness $s \leftarrow D_{\sigma_{\text{LWE}}}$ and noise term $e \leftarrow D_{\sigma_{\text{LWE}}}$.
2. Compute ciphertext $c := p(hs + e) + m \in R_q$.
3. Return ciphertext c .

Partial Decryption $\text{DDec}(\text{sk}_i, c)$. Given secret key share sk_i and ciphertext c :

1. Sample $\tilde{e}_i \leftarrow \chi_{\text{sec}}^*$.
2. Compute $\text{ds}_i := \text{sk}_i \cdot c + p\tilde{e}_i \in R_q$.
3. Return partial decryption ds_i .

Combination $\text{Comb}(\{\text{ds}_i\}_{i \in \{1, 2\}})$. Given partial decryption shares $\{\text{ds}_1, \text{ds}_2\}$:

1. Compute $m' = \text{ds}_1 + \text{ds}_2 \bmod q$.
 2. Return message $m = (m' \bmod p) \in R_p$.
-

Fig. 7. The two-party Th-NTRU scheme.

In the next theorem, we will use the following shorthand notation. Let $\epsilon_{\text{cor}} := 1 - \delta_{\text{DKG}}$, with δ_{DKG} as defined in Lemma 8. Let $\epsilon_{\text{ks}}, \epsilon_{\text{zk}}, \epsilon_{\text{MtA}}, \epsilon_{\text{NTRU}}$ and ϵ_{RLWE} denote bounds on the advantage of any PPT adversary against the knowledge soundness and zero-knowledge property of NIZK, the distinguishing advantage against Π_{MtA} , and the hardness of the NTRU and RLWE problems respectively.

Theorem 4. *Let Π_{MtA} be a certified MtA. Then Th-NTRU is Th-IND-CPA secure under the NTRU and RLWE assumptions. More precisely, for any PPT adversary $\mathcal{A}$ that corrupts at most one party, makes Q_{H} queries to the random oracle with output length l bits, and Q_{DDec} partial decryption queries:*

$$\text{Adv}_{\text{Th-NTRU}}^{\text{Th-IND-CPA}}(\mathcal{A}) \leq \epsilon_{\text{ks}} + \epsilon_{\text{cor}} + \frac{Q_{\text{DDec}}}{2^{\text{sec}}} + \epsilon_{\text{zk}} + \epsilon_{\text{NTRU}} + \epsilon_{\text{RLWE}} + \frac{Q_{\text{H}}}{(qt)^d} + \frac{Q_{\text{H}}^2 + 1}{2^l} + \epsilon_{\text{MtA}}.$$

Proof. We consider a sequence of experiments G_i . Let W_i be the event that the adversary $\mathcal{A}$ wins game G_i .

Experiment G_0 : This is the normal threshold indistinguishability game from Figure 1. The adversary $\mathcal{A}$ outputs the index of the corrupted party. Without loss of generality, we assume $\mathcal{A}$ corrupts $\mathcal{P}_2$.

Experiment G_1 : In this game, if $\text{Vf}(\text{cert}, \{h_1, h_2\}) = 1$, the challenger uses the extraction algorithm to obtain a_2 from the proof in cert . It aborts the game if the extraction fails. It follows from the knowledge soundness property that,

$$|\Pr[W_0] - \Pr[W_1]| \leq \epsilon_{\text{ks}}.$$

Experiment G_2 : In this game, after receiving δ_2 , the challenger finds (h_2, r_2) from the random oracle transcript such that $\delta_2 = H(2, h_2, r_2)$. If no such tuple can be found, it aborts. If in Step 2, the tuple (h'_2, r'_2) sent by the adversary is such that $\delta_2 = H(2, h'_2, r'_2)$ but $(h_2, r_2) \neq (h'_2, r'_2)$, the challenger aborts. We have,

$$|\Pr[W_1] - \Pr[W_2]| \leq \frac{Q_H^2 + 1}{2^l}.$$

Experiment G_3 : In this game, we change how the challenger answers a partial decryption query $(m, (s, e))$. First, it computes $\mathbf{sk}_2 = a_2 - r_1 - r_2 \bmod t$. Let $c = p(hs + e) + m$ be the honestly generated encryption of m . The partial decryption share is computed as $\mathbf{ds}'_1 := m - c \cdot \mathbf{sk}_2 + p\hat{e}$, where $\hat{e} \leftarrow \chi_{\text{sec}}^*$. The other partial decryption queries are computed as in the normal game.

Let **bad** be the event that $\mathbf{sk}_1 + \mathbf{sk}_2 \neq f \bmod q$. We have $a_1 + a_2 = [f_1]_t[f_2]_t \bmod t$ with probability at least δ_{MTA} . Furthermore, since $\mathcal{P}_1$ is honest, r_1 is uniform and following the changes in game G_2 , r_2 is independent from r_1 . Therefore $\mathbf{sk}_1$ is uniform as well. It follows from the same arguments as in Lemma 4 that,

$$\Pr[\text{bad}] \leq 1 - \delta_{\text{DKG}}.$$

Recall that in game G_2 , the partial decryption share is computed as $\mathbf{ds}_1 = c \cdot \mathbf{sk}_1 + p\tilde{e}_1$. In game G_3 we have,

$$\begin{aligned} \mathbf{ds}'_1 &= m - c \cdot \mathbf{sk}_2 + p\hat{e} \\ &= m + c \cdot \mathbf{sk}_1 - c \cdot \mathbf{sk} + p\hat{e}, \end{aligned}$$

unless the event **bad** happens. It therefore suffices to bound the statistical distance between $p\tilde{e}_1$ and $m - c \cdot \mathbf{sk} + p\hat{e}$. Recall that $\tilde{e}_1$ and $\hat{e}$ are both sampled from χ_{sec}^* and that $\mathbf{sk} = f = (pf'_1 + 1)(pf'_2 + 1)$. Note that $f = pk + 1$ for some $k \in R_q$. We have $m - c \cdot \mathbf{sk} + p\hat{e} = -p(gs + fe + km - \hat{e})$. From Lemma 9, we obtain the following equation:

$$\|gs + fe + km\|_\infty \leq B_{\text{LWE}}(B_g + B_f) + (B_f + 1)\sqrt{d}/2. \quad (10)$$

where we used $\|m\|_2 \leq \sqrt{d}p/2$. It follows from Lemma 1 that,

$$\Delta((-p(gs + fe + km - \hat{e})), p\tilde{e}_1) \leq 2^{-\text{sec}}$$

for each individual query. Summing over the at most Q_{DDec} partial decryption queries and adding the probability of **bad**, we have

$$|\Pr[W_2] - \Pr[W_3]| \leq Q_{\text{DDec}}2^{-\text{sec}} + \epsilon_{\text{cor}}.$$

Experiment G_4 : In this game, we change the DKG protocol. The challenger simulates the proofs in the certification cert for $\mathcal{P}_1$. It follows from the zero-knowledge property that,

$$|\Pr[W_3] - \Pr[W_4]| \leq \epsilon_{\text{zk}}.$$

Experiment G_5 : In this game, during the DKG, $\mathcal{P}_1$ participates in the Π_{MtA} protocol using the simulation algorithm. In particular, it does not use its input f_1 .

$$|\Pr[W_4] - \Pr[W_5]| \leq \epsilon_{\text{MtA}}.$$

Experiment G_6 : In this game, we change the DKG protocol again. Party $\mathcal{P}_1$ returns $h_1 \xleftarrow{\$} R_q$ instead of $h_1 = f_1^{-1}g_1$. It follows from the NTRU assumption that,

$$|\Pr[W_5] - \Pr[W_6]| \leq \epsilon_{\text{NTRU}}.$$

Experiment G_7 : In this game, the challenger derives h_1 after receiving δ_2 . In the first step of the DKG, $\mathcal{P}_1$ samples a random $\delta_1 \xleftarrow{\$} \{0, 1\}^l$. It then samples $h \xleftarrow{\$} R_q$ and $\mathcal{P}_1$ sets $h_1 = hh_2^{-1}$, where h_2 is the element obtained from the random oracle transcript as in G_2 . The challenger programs H such that $H(1, h_1, r_1) = \delta_1$. If this step fails, it aborts.

If Step 2 does not abort, then $h_2 \in R_q^\times$ and h_2^{-1} exists. Since h is sampled from the uniform distribution, h_2 is a unit and independent from h following changes in game G_2 , we have that h_1 is uniformly distributed over R_q .

Thus, the tuple (h_1, δ_1) is distributed as in game G_6 unless the programming step fails. The probability of that event can be bounded by $Q_H/(qt)^d$ and we have,

$$|\Pr[W_6] - \Pr[W_7]| \leq \frac{Q_H}{(qt)^d}.$$

Experiment G_8 : Finally, we change how the challenge ciphertext ctx_b is computed. The challenger samples $c' \xleftarrow{\$} R_q$ from the uniform distribution and returns $\text{ctx}_b := c' + m_b$. It follows from the RLWE assumption that

$$|\Pr[W_7] - \Pr[W_8]| \leq \epsilon_{\text{RLWE}}.$$

The distribution of the challenge ciphertext in game G_8 is now independent of b . We have, $|\Pr[W_8] - \frac{1}{2}| = 0$, which completes the proof. $\square$

5 ZK Relations

In this section we present the relations required for the zero-knowledge proofs used in Π_{MtA1} , Π_{MtA2} , and DKG_{RLWE} . Throughout, we implicitly identify elements of R_t with their canonical lift to R_Q whenever they appear in ciphertext equations. In a concrete instantiation of NIZK, the prover commits to the relevant witness values and proves that the committed values satisfy the corresponding relation. These commitments are included in the proof transcript and are omitted below for readability.

5.1 ZK Relations for Π_{MtA1}

The proof π_h shows that $\mathcal{P}_1$ knows a valid NTRU secret key corresponding to h , and is later used to support extraction of a_1 .

$$\mathcal{R}_{\text{KeyGen}_{\text{NTRU}}} := \left\{ (X, W) \mid \begin{array}{l} X := (h), W := (f, g) : \|f\|_2, \|g\|_2 \leq B_{\text{NTRU}} \wedge \\ f \in R_t^\times \wedge f \in R_Q^\times \wedge hf = g \bmod Q \end{array} \right\} \quad (1)$$

If π_h verifies, then h is well formed with respect to the witness (f, g) satisfying Relation 1. By knowledge soundness of NIZK, the extractor recovers f .

The proof π_1 shows that ctx_1 is a valid encryption of m_1 under the public key h , and in the certified Π_{MtA1} , that $\mathcal{R}_{\text{cert}}(m_1, y_1) = 1$.

$$\mathcal{R}_{\text{MtA1}, \text{ctx}_1} := \left\{ (X, W) \mid \begin{array}{l} X := (h, \text{ctx}_1, y_1), W := (s_1, e_1, m_1) : \\ m_1 \in R_t \wedge \|m_1\|_2 \leq B_m \wedge \|s_1\|_2, \|e_1\|_2 \leq B_{\text{LWE}} \wedge \\ \text{ctx}_1 = t(hs_1 + e_1) + m_1 \in R_Q \wedge \mathcal{R}_{\text{cert}}(m_1, y_1) = 1 \end{array} \right\} \quad (2)$$

If π_1 verifies, then ctx_1 is a well formed encryption of m_1 satisfying Relation 2. By knowledge soundness of NIZK, the extractor recovers m_1 .

The proof π_2 shows that ctx_2 is formed correctly from ctx_1, m_2, a_2 and in the certified Π_{MtA1} that $\mathcal{R}_{\text{cert}}(m_2, y_2) = 1$.

$$\mathcal{R}_{\text{MtA1}, \text{ctx}_2} := \left\{ (X, W) \mid \begin{array}{l} X := (h, \text{ctx}_1, \text{ctx}_2, y_2), W := (m_2, a_2, s_2, e_2) : \\ m_2 \in R_t \wedge \|m_2\|_2 \leq B_m \wedge a_2 \in R_t \wedge \|s_2\|_2 \leq B_{\text{sec}} \wedge \\ \|e_2\|_2 \leq B_{\text{sec}} \wedge \text{ctx}_2 = \text{ctx}_1 \cdot m_2 - a_2 + t(hs_2 + e_2) \wedge \\ \mathcal{R}_{\text{cert}}(m_2, y_2) = 1 \end{array} \right\} \quad (3)$$

If π_2 verifies, then ctx_2 is well formed with respect to Relation 3. By knowledge soundness of NIZK, the extractor recovers m_2 and a_2 .

5.2 ZK Relations for Π_{MtA2}

The proofs π_1 and π_2 prove well formedness of the ciphertexts sent by $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, with respect to $\text{pk} = (a, b)$. In the certified Π_{MtA2} they also prove that $\mathcal{R}_{\text{cert}}(m_i, y_i) = 1$. The proof $\pi_{\text{pk}}^{(i)}$ proves well formedness of $\mathcal{P}_i$'s public key share in DKG_{RLWE} and supports extraction of the decryption key share s_i .

The proof π_1 shows that (u'_1, v'_1) is well formed with respect to a small $m_1 \in R_t$ and short values $r'_1, e'_1, e'_2 \in R_Q$, and in the certified Π_{MtA2} that $\mathcal{R}_{\text{cert}}(m_1, y_1) = 1$.

$$\mathcal{R}_{\text{MtA2}, \mathcal{P}_1, \text{Enc}} := \left\{ (X, W) \mid \begin{array}{l} X := (\text{pk} = (a, b), u'_1, v'_1, y_1), W := (m_1, r'_1, e'_1, e'_2) : \\ m_1 \in R_t \wedge \|m_1\|_2 \leq B_m \wedge \|r'_1\|_2 \leq B_{\text{LWE}} \wedge \\ \|e'_1\|_2 \leq B_{\text{LWE}} \wedge \|e'_2\|_2 \leq B_{\text{LWE}} \wedge \mathcal{R}_{\text{cert}}(m_1, y_1) = 1 \wedge \\ u'_1 = ar'_1 + te'_1 - m_1 \in R_Q \wedge v'_1 = br'_1 + te'_2 \in R_Q \end{array} \right\} \quad (4)$$

If π_1 verifies, then (u'_1, v'_1) is well formed, with respect to the witness (m_1, r'_1, e'_1, e'_2) satisfying Relation 4. By knowledge soundness of NIZK, the extractor recovers m_1 .

The proof π_2 shows that (u_2, v_2) is a well-formed encryption of a small $m_2 \in R_t$ with short values $r_2, e_3, e_4 \in R_Q$, and in the certified Π_{MtA2} , that $\mathcal{R}_{\text{cert}}(m_2, y_2) = 1$.

$$\mathcal{R}_{\text{MtA2}, \mathcal{P}_2, \text{Enc}} := \left\{ (X, W) \mid \begin{array}{l} X := (\text{pk} = (a, b), u_2, v_2, y_2), W := (m_2, r_2, e_3, e_4) : \\ m_2 \in R_t \wedge \|m_2\|_2 \leq B_m \wedge \|r_2\|_2 \leq B_{\text{LWE}} \wedge \\ \|e_3\|_2 \leq B_{\text{LWE}} \wedge \|e_4\|_2 \leq B_{\text{LWE}} \wedge \mathcal{R}_{\text{cert}}(m_2, y_2) = 1 \wedge \\ u_2 = ar_2 + te_3 \in R_Q \wedge v_2 = br_2 + te_4 + m_2 \in R_Q \end{array} \right\} \quad (5)$$

If π_2 verifies, then (u_2, v_2) is a well formed encryption of m_2 , with respect to the witness (m_2, r_2, e_3, e_4) satisfying Relation 5. By the knowledge soundness of NIZK, the extractor recovers m_2 .

The proof $\pi_{\text{pk}}^{(i)}$ proves the well-formedness of $\mathcal{P}_i$'s public key share b_i in DKG_{RLWE} with respect to the short values $s_i, e_i \in R_Q$.

$$\mathcal{R}_{\text{MtA2}, \text{DKG}_{\text{RLWE}}} := \left\{ (X, W) \mid \begin{array}{l} X := (a, b_i), W := (s_i, e_i) : \\ \|s_i\|_2 \leq B_{\text{LWE}} \wedge \|e_i\|_2 \leq B_{\text{LWE}} \wedge b_i = as_i + te_i \in R_Q \end{array} \right\}_{(6)}$$

If $\pi_{\text{pk}}^{(i)}$ verifies, then the public key share b_i is well formed with respect to the witness (s_i, e_i) , satisfying Relation 6. By the knowledge soundness of the NIZK, the extractor recovers s_i .

6 Parameters

We present concrete parameter sets to instantiate **Th-NTRU**. We target at least $\lambda = 128$ bit computational security, $\text{sec} = 40$ bit statistical security, and correctness. Note that the per-run failure rate of the DKG_{NTRU} must remain negligible for security. The computational security estimates are obtained using the lattice estimator [APS15] for the RLWE instances, and its NTRU estimator for the NTRU instances. Note that this analysis takes care to avoid so-called ‘overstretched’ parameter regimes that can reduce NTRU hardness over large moduli [KF17, DvW21, HSS24].

Modulus and noise distributions. Our full construction uses four moduli, p, q, t, Q , where p and q are the plaintext and ciphertext moduli used in **Th-NTRU**, and t and Q are the plaintext and ciphertext moduli of the MtA protocols. We sample RLWE secret and error terms from a truncated discrete Gaussian distribution $D_{\sigma_{\text{LWE}}}$. We denote by σ_{NTRU}^1 the standard deviation of the truncated discrete Gaussian used to sample the NTRU secrets in Π_{MtA1} . Finally, we write σ_{NTRU}^2 to denote the standard deviation of the truncated discrete Gaussian used to sample the NTRU secrets in the key generation protocol DKG_{NTRU} .

Correctness constraints. We choose plaintext modulus $p = 2$. The remaining **Th-NTRU** parameters in Table 1 are chosen such that the computational problems are hard and the correctness conditions derived in Lemmata 8 and 9 are satisfied.

From Theorem 4, we also require $\delta_{\text{DKG}} = 1 - 2^{-\text{sec}}$. This value depends on both t and the correctness probability of the MtA protocol. For Π_{MtA1} , the modulus Q must be large enough so that Lemma 6 is satisfied. At the same time, the $D_{\sigma_{\text{NTRU}}}$ distribution must sample from a Gaussian with sufficiently large standard deviation σ_{NTRU}^1 to avoid parameter regimes where dense sub-lattice attacks become effective. For Π_{MtA2} , the ratio t/Q must be sufficiently small to satisfy Theorem 2 with overwhelming probability.

Table 1. Parameter sets for the **Th-NTRU** scheme instantiated using Π_{MtA1} and Π_{MtA2} . We use two sets based on the noise flooding approach described in Section 4.2 and two sets based on decryption with MPC rounding.

Parameter	Explanation	Noise flooding		MPC	
		Π_{MtA1}	Π_{MtA2}	Π_{MtA1}	Π_{MtA2}
λ	Computational security parameter	128	128	128	128
sec	Statistical security parameter	40	40	40	40
d	Ring dimension	8192	8192	2048	2048
p	Th-NTRU plaintext modulus	2	2	2	2
$\log q$	Th-NTRU ciphertext mod. bits	94	94	26	26
$\log t$	MtA plaintext mod. bits	77	78	N/A	N/A
$\log Q$	MtA ciphertext mod. bits	207	168	168	105
σ_{LWE}	Standard dev. of $D_{\sigma_{\text{LWE}}}$	$\sqrt{2/3}$	$\sqrt{2/3}$	$\sqrt{2/3}$	$\sqrt{2/3}$
σ_{NTRU}^1	Standard dev. of $D_{\sigma_{\text{NTRU}}}$ in Π_{MtA1}	2^{44}	N/A	2^{66}	N/A
σ_{NTRU}^2	Standard dev. of $D_{\sigma_{\text{NTRU}}}$ in DKG_{NTRU}	$2^{4.5}$	$2^{4.5}$	$\sqrt{2/3}$	$\sqrt{2/3}$

Concrete parameter sets. Table 1 shows the resulting parameter sets we give for the two instantiations based on Π_{MtA1} and Π_{MtA2} .

Zyskind et al. introduced a novel method for threshold decryption of RLWE schemes [ZZLP25]. It avoids noise flooding by using an MPC protocol to round away the noise. While this requires some pre-processing, it allows one to choose significantly better parameters. As we expect this method to apply to **Th-NTRU**, we also give an alternate parameter set for **Th-NTRU** without noise flooding. Since q can then be chosen significantly smaller, we forgo the reduction mod t completely, and run the **MtA** with plaintext space q .

Sizes. Table 2 shows the communication cost and key sizes for **Th-NTRU** using Π_{MtA1} and Π_{MtA2} . Again we display two sets; one based on noise flooding and another using MPC-based decryption rounding. We instantiate NIZKs throughout using the LaBRADOR proof system [BS23a], implemented in the LaZer library [LSS24]. We use an estimated, succinct, proof size of 60KB as proposed by [BS23a].

Discussion. The concrete parameter sets show that our approach yields a dealer-free dedicated DKG for NTRU, and that this is usable in a threshold version of the **NTRU-Encrypt** scheme, **Th-NTRU**, with 128-bit computational security verified using the lattice estimator. The parameters illustrate the trade-off between our two certified **MtA** instantiations: Π_{MtA1} avoids an interactive setup but has a sequential online phase, whereas Π_{MtA2} requires an initial DKG_{RLWE} setup to make use of homomorphic secret sharing techniques introduced in [BKS19] to compute the additive NTRU secret shares. Overall, we show that threshold **NTRU-Encrypt** with non-interactive distributed decryption can be instantiated

Table 2. Communication cost and key sizes for the Th-NTRU scheme instantiated using Π_{MtA1} and Π_{MtA2} . We use two sets based on the noise flooding approach described in Section 4.2 and two sets based on decryption with MPC rounding. All sizes are given in kilobytes.

Per-party cost	Noise flooding		MPC	
	Π_{MtA1}	Π_{MtA2}	Π_{MtA1}	Π_{MtA2}
Communication cost	583	695	167	170
Secret key share	78	79	6	6
Public key (global)	95	95	6	6

concretely without generic MPC. To the best of our knowledge this gives the first concrete distributed key generation protocol for NTRU keys of its kind.

A comparison with generic MPC. We wish to provide a comparison with what we believe is the only other existing technique that could *theoretically* perform DKG for NTRU. In [RST⁺22] (see also the correction [RST⁺26]), the authors propose an actively secure DKG for the MPC protocol SPDZ [DPSZ12] based on MASCOT [KOS16] (to perform the oblivious transfer operations). The underlying key generation for SPDZ is essentially that of the BGV encryption scheme. Thus [RST⁺22] essentially proposes a generic MPC framework for computing DKG for BGV (RLWE) keys.

One attraction of this approach is that the distributions of the resulting public key/secret key shares are essentially identical to those in the trusted setup setting. However, as we will discuss, the computation, communication, and number of rounds needed make this approach *very* expensive. Indeed, DKG for SPDZ in [RST⁺22] takes around 40 minutes to complete for their smallest parameter set, even over a LAN with high throughput.

Whilst it is quite hard to estimate how the techniques of [RST⁺22] would apply to NTRU DKG, we expect they would result in an even more expensive solution, both in communication and running time.

The unit cost in [RST⁺22] is the MASCOT per-Beaver triple communication cost. As applied to NTRU, one needs to sample f and g , perform one NTT inversion to compute $h = f^{-1}g$ ($\approx 2d$ Beaver triples + one opening), and a (free) linear assembly. We count $\mathbb{F}_q$ triples for our smallest noise-flooding parameter set ($\log q \approx 94$), and generously assume that an in-MPC sampler for the Gaussian with $\sigma_{\text{NTRU}}^2 = 2^{4.5}$ that this set requires is no more expensive than a centered binomial sampler with $k = 1$, that is, two uniform bits per coefficient, each bit costing one GenBit and hence one multiplication. Under this assumption, sampling f and g requires $4d$ triples and inversion costs another $2d$. Adopting the per-triple formula of [RST⁺22], one might optimistically assume the protocol to require 2.5 million triples, resulting in a per-party communication cost of 7GB. This is a blowup of roughly four orders of magnitude over our worst-case designs with noise flooding.

Whilst we don't provide an implementation, the ring operations and NIZK building blocks are widely used and implemented in other works as well as the NIST post-quantum standards implementations. We therefore expect that the running time of all our protocols would be in the order of seconds (< 10) and dominated by NIZK generation. On the other hand, we expect the running time of the techniques of [RST⁺22] to be significantly higher than this, in the order of minutes or hours.

Future work. Two natural directions for future work are extending our constructions beyond the two-party setting and lifting them to the module setting. Extending $\Pi_{\text{MtA}2}$ to more parties appears infeasible; other compatible MtA constructions, including $\Pi_{\text{MtA}1}$, may generalize more readily. A module generalization of our protocols seems plausible in principle, but its correctness and security analysis would need to be revisited carefully in that setting, which we leave to future work.

References

- ABGS23. Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjandra Silde. Verifiable mix-nets and distributed decryption for voting from lattice-based assumptions. In *ACM CCS 2023: 30th Conference on Computer and Communications Security*, 2023. <https://doi.org/10.1145/3576915.3616683>
- AJL⁺12. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology – EUROCRYPT 2012*, 2012. https://doi.org/10.1007/978-3-642-29011-4_29
- APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.
- BBC⁺20. Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Technical report, National Institute of Standards and Technology, 2020. available at <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>.
- BBJ⁺22. Shi Bai, Austin Beard, Floyd Johnson, Sulani K. B. Vidhanalage, and Tran Ngo. Fiat-Shamir signatures based on module-NTRU. In *ACISP 22: 27th Australasian Conference on Information Security and Privacy*, 2022. https://doi.org/10.1007/978-3-031-22301-3_15
- BC25. Katharina Boudgoust and Anamaria Costache. Improved Rényi arguments for lattice-based threshold encryption. Cryptology ePrint Archive, Report 2025/735, 2025. <https://eprint.iacr.org/2025/735>
- BD10. Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *TCC 2010: 7th Theory of Cryptography Conference*, 2010. https://doi.org/10.1007/978-3-642-11799-2_13

- BGG⁺18. Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *Advances in Cryptology – CRYPTO 2018, Part I*, 2018. https://doi.org/10.1007/978-3-319-96884-1_19
- BHP17. Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In *TCC 2017: 15th Theory of Cryptography Conference, Part I*, 2017. https://doi.org/10.1007/978-3-319-70500-2_22
- BJRW22. Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. Entropic hardness of module-LWE from module-NTRU. In *Progress in Cryptology - INDOCRYPT 2022: 23rd International Conference in Cryptology in India*, 2022. https://doi.org/10.1007/978-3-031-22912-1_4
- BKS19. Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In *Advances in Cryptology – EUROCRYPT 2019, Part II*, 2019. https://doi.org/10.1007/978-3-030-17656-3_1
- BS23a. Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In *Advances in Cryptology – CRYPTO 2023, Part V*, 2023. https://doi.org/10.1007/978-3-031-38554-4_17
- BS23b. Katharina Boudgoust and Peter Scholl. Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In *Advances in Cryptology – ASIACRYPT 2023, Part I*, 2023. https://doi.org/10.1007/978-981-99-8721-4_12
- CGG⁺20. Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In *ACM CCS 2020: 27th Conference on Computer and Communications Security*, 2020. <https://doi.org/10.1145/3372297.3423367>
- CPS⁺20. Chitchanok Chuengsatiansup, Thomas Prest, Damien Stehlé, Alexandre Wallet, and Keita Xagawa. ModFalcon: Compact signatures based on module-NTRU lattices. In *ASIACCS 20: 15th ACM Symposium on Information, Computer and Communications Security*, 2020. <https://doi.org/10.1145/3320269.3384758>
- CS19. Chitchanok Chuengsatiansup and Damien Stehlé. Towards practical GGM-based PRF from (module-)learning-with-rounding. In *SAC 2019: 26th Annual International Workshop on Selected Areas in Cryptography*, 2019. https://doi.org/10.1007/978-3-030-38471-5_28
- CSS⁺22. Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. Efficient threshold FHE for privacy-preserving applications. Cryptology ePrint Archive, Report 2022/1625, 2022. <https://eprint.iacr.org/2022/1625>
- DDEK⁺23. Morten Dahl, Daniel Demmler, Sarah El Kazdadi, Arthur Meyre, Jean-Baptiste Orfila, Dragos Rotaru, Nigel P. Smart, Samuel Tap, and Michael Walter. Noah’s ark: Efficient threshold-fhe using noise flooding. In *Proceedings of the 11th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, WAHC ’23, page 35–46, New York, NY, USA, 2023. Association for Computing Machinery. <https://doi.org/10.1145/3605759.3625259>

- DKL⁺13. Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In *ESORICS 2013: 18th European Symposium on Research in Computer Security*, 2013. https://doi.org/10.1007/978-3-642-40203-6_1
- DKLs24. Jack Doerner, Yashvanth Kondi, Eysa Lee, and abhi shelat. Threshold ECDSA in three rounds. In *2024 IEEE Symposium on Security and Privacy*, 2024. <https://doi.org/10.1109/SP54263.2024.00178>
- DKM⁺24. Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In *Advances in Cryptology – EUROCRYPT 2024, Part II*, 2024. https://doi.org/10.1007/978-3-031-58723-8_8
- dPPRS23. R. del Pino, T. Prest, M. Rossi, and M. Saarinen. High-order masking of lattice signatures in quasilinear time. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1168–1185, Los Alamitos, CA, USA, may 2023. IEEE Computer Society. <https://doi.org/10.1109/SP46215.2023.00160>
- DPSZ12. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multi-party computation from somewhat homomorphic encryption. In *Advances in Cryptology – CRYPTO 2012*, 2012. https://doi.org/10.1007/978-3-642-32009-5_38
- DvW21. Léo Ducas and Wessel P. J. van Woerden. NTRU fatigue: How stretched is overstretched? In *Advances in Cryptology – ASIACRYPT 2021, Part IV*, 2021. https://doi.org/10.1007/978-3-030-92068-5_1
- FPS22. Joël Felderhoff, Alice Pellet-Mary, and Damien Stehlé. On module unique-SVP and NTRU. In *Advances in Cryptology – ASIACRYPT 2022, Part III*, 2022. https://doi.org/10.1007/978-3-031-22969-5_24
- GG18. Rosario Gennaro and Steven Goldfeder. Fast multiparty threshold ECDSA with fast trustless setup. In *ACM CCS 2018: 25th Conference on Computer and Communications Security*, 2018. <https://doi.org/10.1145/3243734.3243859>
- GHK⁺26. Kamil Doruk Gur, Patrick Hough, Jonathan Katz, Caroline Sandsbråten, and Tjerand Silde. Olingo: Threshold lattice signatures with DKG and identifiable abort. In *ACM CCS 2026: 33rd Conference on Computer and Communications Security*, The Hague, The Netherlands, November 15–19, 2026. ACM Press. <https://doi.org/10.1145/3830454.3832660>
- GKS24. Kamil Doruk Gür, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice-based signatures from threshold homomorphic encryption. In *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*, 2024. https://doi.org/10.1007/978-3-031-62746-0_12
- GM84. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2), 1984.
- HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory, Third International Symposium (ANTS-III)*, pages 267–288, 1998. <https://doi.org/10.1007/BFb0054868>
- HSS24. Patrick Hough, Caroline Sandsbråten, and Tjerand Silde. More efficient lattice-based electronic voting from NTRU. *IACR Communications in Cryptology (CiC)*, 1(4), 2024. <https://doi.org/10.62056/a69qudhjdj>

- KF17. Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In *Advances in Cryptology – EUROCRYPT 2017, Part I*, 2017. https://doi.org/10.1007/978-3-319-56620-7_1
- KOS16. Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, 2016. <https://doi.org/10.1145/2976749.2978357>
- LDK⁺22. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- LN18. Yehuda Lindell and Ariel Nof. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In *ACM CCS 2018: 25th Conference on Computer and Communications Security*, 2018. <https://doi.org/10.1145/3243734.3243788>
- LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology – EUROCRYPT 2010*, 2010. https://doi.org/10.1007/978-3-642-13190-5_1
- LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3), 2015. <https://doi.org/10.1007/s10623-014-9938-4>
- LS18. Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *Advances in Cryptology – EUROCRYPT 2018, Part I*, 2018. https://doi.org/10.1007/978-3-319-78381-9_8
- LS19. Vadim Lyubashevsky and Gregor Seiler. NTTU: Truly fast NTRU using NTT. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(3), 2019. <https://doi.org/10.13154/tches.v2019.i3.180-201>
- LSS24. Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In *ACM CCS 2024: 31st Conference on Computer and Communications Security*, 2024. <https://doi.org/10.1145/3658644.3690330>
- LTV12. Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *44th Annual ACM Symposium on Theory of Computing*, 2012. <https://doi.org/10.1145/2213977.2214086>
- MS24. Daniele Micciancio and Adam Suhl. Simulation-secure threshold PKE from LWE with polynomial modulus. *IACR Communications in Cryptology (CiC)*, 1(4), 2024. <https://doi.org/10.62056/a0zogy4e->
- PFH⁺22. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing*, 2005. <https://doi.org/10.1145/1060590.1060603>

- Ros02. Michael Rosen. *Polynomials over Finite Fields*, pages 1–9. Springer New York, New York, NY, 2002. https://doi.org/10.1007/978-1-4757-6046-0_1
- RST⁺22. Dragos Rotaru, Nigel P. Smart, Titouan Tanguy, Frederik Vercauteren, and Tim Wood. Actively secure setup for SPDZ. *Journal of Cryptology*, 35(1), 2022. <https://doi.org/10.1007/s00145-021-09416-w>
- RST⁺26. Dragos Rotaru, Nigel P. Smart, Titouan Tanguy, Frederik Vercauteren, and Tim Wood. Correction to: Actively secure setup for SPDZ. *Journal of Cryptology*, 39(2):22, 2026. <https://doi.org/10.1007/s00145-026-09574-9>
- SAB⁺22. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- SS11. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Advances in Cryptology – EUROCRYPT 2011*, 2011. https://doi.org/10.1007/978-3-642-20465-4_4
- SXY18. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Advances in Cryptology – EUROCRYPT 2018, Part III*, 2018. https://doi.org/10.1007/978-3-319-78372-7_17
- XAX⁺21. Haiyang Xue, Man Ho Au, Xiang Xie, Tsz Hon Yuen, and Handong Cui. Efficient online-friendly two-party ECDSA signature. In *ACM CCS 2021: 28th Conference on Computer and Communications Security*, 2021. <https://doi.org/10.1145/3460120.3484803>
- ZZLP25. Guy Zyskind, Doron Zarchy, Max Leibovich, and Chris Peikert. High-throughput universally composable threshold FHE decryption. In *ACM CCS 2025: 32nd Conference on Computer and Communications Security*, 2025. <https://doi.org/10.1145/3719027.3744884>