Threshold Lattice-Based Zero-Knowledge Proofs

Scott Griffy [*], Victor Youdom Kemmoe [*], Ngoc Khanh Nguyen [†], Tjerand Silde [‡]

[*]Brown University, US
scottgriffy@gmail.com, vyoudomk@cs.brown.edu
[†]King's College London, UK
ngoc_khanh.nguyen@kcl.ac.uk
[‡]Norwegian University of Science and Technology, Norway
tjerand.silde@ntnu.no

Abstract

Lattice-based zero-knowledge proofs are now efficient enough for practical use, but in all known constructions a single prover holds the entire witness and is therefore a single point of failure. Thresholdizing them is understood only for three-round Σ-protocols, which certify shortness only *approximately*. The *exact* statements needed by applications such as anonymous credentials require more rounds and rely on rejection sampling, and neither property survives thresholdization.

We construct the first lattice-based threshold zero-knowledge proof systems for exact relations. The witness is Shamir-shared among $\mathtt{n}$ parties, any $\mathtt{t}$ of them can jointly produce a proof, and the proof has the same form as a single-prover proof, only a factor $\sqrt{\mathtt{t}}$ larger, with verification unchanged. We thresholdize the product proof of Attema, Lyubashevsky, and Seiler (CRYPTO 2020) and the exact proof of Esgin, Nguyen, and Seiler (ASIACRYPT 2020), making both rejection-free using Hint-MLWE and evaluating them over threshold homomorphic encryption. We define threshold commit-and-prove protocols with the corresponding zero-knowledge and simulation-extractability notions, and prove our constructions secure against passive adversaries that statically corrupt at most $\mathtt{t} - 1$ parties.

Of independent interest, we show that the Fiat–Shamir transforms of both proof systems are simulation-extractable in the random oracle model, and that MLWE remains hard when secrets are drawn from the subring fixed by a ring automorphism.

Keywords: Lattices, Threshold cryptography, Zero-knowledge proofs

1 Introduction

A zero-knowledge proof lets a prover convince a verifier that a statement is true while revealing nothing beyond its validity. Such proofs underlie most privacy-preserving primitives, from anonymous credentials [CL03] and verifiable encryption [LN17] to private payments [BCG+14]. With the steady progress in building quantum computers, realizing them from post-quantum assumptions has become pressing, and lattices are the most mature foundation on which to do so. Lattices also support advanced functionality, most notably fully homomorphic encryption [BV11, BGV12], that is not known from any other cryptographic assumption, which makes it natural to prove lattice statements with a lattice-based proof system.

Lattice-based zero-knowledge proofs have come a long way. Early protocols for proving knowledge of a short solution to a linear system followed a combinatorial, Stern-like paradigm and produced proofs of several megabytes [LNSW13, LLNW18]. The algebraic line of work initiated by Bootle, Lyubashevsky, and Seiler [BLS19], refined in a sequence of follow-ups [ALS20, ENS20, LNS20, LNP22, LSS24], has brought this down to a dozen kilobytes and put lattice-based proofs within reach of real applications. In all of these

protocols, however, the proof is produced by a single prover holding the entire witness, who is therefore a single point of failure.

Threshold cryptography [Des88, DF90] exists to remove such weaknesses: the secret is split into $\mathtt{n}$ shares [Sha79], and the scheme retains its security as long as no more than $\mathtt{t}-1$ parties are corrupted. The practical importance of the paradigm is reflected in the ongoing NIST standardization effort for threshold cryptography,[1] whose call lists zero-knowledge proof systems among the primitives to be thresholdized. One motivating application is multi-holder anonymous credentials (MHAC) [FLL25], where a credential is stored across several of a user's devices so that presenting it requires all devices to interact, and no single compromised device can be used to impersonate the user. For attribute-based credentials [CG08], where relatively complex predicates are proven on a committed value, this quickly becomes unwieldy without an efficient threshold proof system.

Thresholdizing lattice-based zero-knowledge proofs nevertheless remains open. The gap is not merely theoretical: the LaZer library [LNO+26], which packages the state of the art in lattice-based proofs [LSS24, LNP22], is itself planned for submission to the NIST call above, yet no threshold construction for these proof systems is known. The difficulty is specific to the lattice setting. Efficient thresholdization is well understood for three-round Σ-protocols, and threshold lattice signatures built from them are by now practical [ASY22, GKS24, DKM+24, CATZ24, PKN+25]. But a Σ-protocol only certifies an *approximate* solution: it establishes that *some* related vector is short, not that the witness itself has the prescribed form. The applications above need *exact* statements, for instance that every coefficient of the witness lies in $\{0, \pm 1\}$. Every exact lattice proof system uses more than three rounds, and relies on rejection sampling, where the prover restarts whenever its response would leak information about the witness. Neither property survives thresholdization for free.

1.1 Our Contributions

We construct the first lattice-based threshold zero-knowledge proof systems for exact relations. In more detail:

- We give threshold versions of the product proof of Attema, Lyubashevsky, and Seiler (ALS) [ALS20], which proves multiplicative relations between committed ring elements, and of the exact proof of Esgin, Nguyen, and Seiler (ENS) [ENS20], which proves that a committed vector has coefficients in $\{0, \pm 1\}$ (Sections 5 and 6). The witness is Shamir-shared among $\mathtt{n}$ parties, so that any $\mathtt{t}$ of them can jointly produce a proof while any $\mathtt{t}-1$ of them learn nothing about it. Both are rejection-free, which we obtain by widening the masking vectors and appealing to Hint-MLWE [KLSS23] rather than aborting.

- The resulting proofs have the same shape as single-prover ALS and ENS proofs and are only a factor of $\sqrt{\mathtt{t}}$ larger, since each party's mask must hide its own share. Verification is unchanged, and the verifier need not know that several parties were involved.

- We introduce definitions of threshold commit-and-prove protocols, in which the parties first commit to their shares of a witness and later prove statements about it, together with the notions of weak $(\mathtt{t}-1)$-zero-knowledge and threshold simulation-extractability that such protocols should satisfy (Section 4). As we explain there, threshold simulation-extractability does not follow from combining $(\mathtt{t}-1)$-zero-knowledge of the threshold protocol with knowledge soundness of the single-prover proof it is built from, since the latter may be malleable.

- As a building block of independent interest, we prove that the Fiat–Shamir transforms of ALS and ENS are simulation-extractable in the random oracle model, using the predicate special soundness of

[1] https://csrc.nist.gov/projects/threshold-cryptography

Aardal et al. [AAB+24] together with the weak unique response notion of Ganesh et al. [GOP+25] (Sections 5.1 and 6.1).

- Also of independent interest, we show that MLWE remains hard when secrets are drawn from the subring fixed by a ring automorphism (Section 7), which lets us recover the automorphism-based soundness optimizations of ALS and ENS inside the homomorphic layer.

Our protocols are proven secure against passive adversaries. We view this as the right first step given their complexity: the multi-round structure, the homomorphic evaluation, and the simulation of the aggregated response already require substantial machinery, and settling them in the passive setting isolates what is genuinely new. We expect that they can be promoted to active security in the standard way, by having each party commit to its contributions and prove in zero-knowledge that they were computed correctly, at the cost of larger proofs and additional communication. Some care will be needed beyond that: handling rushing adversaries is likely to require further rounds, so that no party can choose its contribution after seeing those of the others. We leave this to future work, and note that the constructions below already provide the core ideas on which such a treatment would rest.

1.2 Technical Overview

We begin with a high-level overview of the framework of Gür, Katz, and Silde [GKS24] (the GKS framework), refined in Olingo [GHK+26], for building threshold signatures from threshold homomorphic encryption, and explain why it does not extend to proof systems for exact relations. We then describe the key ideas underlying our constructions. Several technical details are omitted, since this is intended only to provide intuition.

The GKS framework. Fix a ring R_q and recall the three-message identification scheme of Lyubashevsky [Lyu12], whose secret key is a short $\vec{\mathbf{s}}$ and whose public key is $(\mathbf{A}, \vec{\mathbf{u}} := \mathbf{A}\vec{\mathbf{s}})$. The prover sends $\vec{\mathbf{w}} := \mathbf{A}\vec{\mathbf{y}}$ for a short masking vector $\vec{\mathbf{y}}$, receives a short challenge $\mathbf{c}$, and replies with $\vec{\mathbf{z}} := \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{s}}$; the verifier accepts if $\vec{\mathbf{z}}$ is short and $\mathbf{A}\vec{\mathbf{z}} = \vec{\mathbf{w}} + \mathbf{c}\vec{\mathbf{u}}$. GKS distribute this among $\mathtt{n}$ parties by having each party hold an encryption $\mathsf{ctx}_{\vec{\mathbf{s}}}$ of $\vec{\mathbf{s}}$ under a threshold homomorphic encryption scheme whose *decryption* key is Shamir-shared $\mathtt{t}$-out-of-$\mathtt{n}$. Any set $\mathcal{S}$ of at least $\mathtt{t}$ parties can then sign as follows:

1. For $j \in \mathcal{S}$, party j samples a bounded $\vec{\mathbf{y}}_j$ and sends $\vec{\mathbf{w}}_j := \mathbf{A}\vec{\mathbf{y}}_j$ together with $\mathsf{ctx}_{\vec{\mathbf{y}}_j}$, an encryption of $\vec{\mathbf{y}}_j$.
2. Each party locally computes $\vec{\mathbf{w}} := \sum_{j\in\mathcal{S}} \vec{\mathbf{w}}_j$, hashes it to obtain the challenge $\mathbf{c} := H(\vec{\mathbf{w}})$, forms the encrypted response $\mathsf{ctx}_{\vec{\mathbf{z}}} := \mathbf{c} \cdot \mathsf{ctx}_{\vec{\mathbf{s}}} + \sum_{j\in\mathcal{S}} \mathsf{ctx}_{\vec{\mathbf{y}}_j}$ homomorphically, and sends its decryption share of $\mathsf{ctx}_{\vec{\mathbf{z}}}$.
3. Given the decryption shares, each party recovers $\vec{\mathbf{z}}$ and outputs $(\mathbf{c}, \vec{\mathbf{z}})$.

Two features make this framework efficient. The signing key is never Shamir-shared, only the decryption key is, which sidesteps the difficulty that shares of a *short* secret need not themselves be short; decryption shares, by contrast, are allowed to be large. Further, the random oracle is evaluated on $\vec{\mathbf{w}}$, which is public by construction, so the hash never has to be computed inside the homomorphic evaluation. It is precisely the latter that the generic compiler of Boneh et al. [BGG+18] cannot avoid, and why that compiler yields only heuristically secure instantiations.

From approximate to exact. What this framework produces is an approximate proof of shortness. From two accepting transcripts sharing a first message one extracts a short $\vec{\mathbf{s}}^*$, longer than $\vec{\mathbf{s}}$, together with a slack term $\mathbf{c}^*$ satisfying $\mathbf{A}\vec{\mathbf{s}}^* = \mathbf{c}^*\vec{\mathbf{u}}$. For unforgeability that is enough, since the reduction only needs some short

preimage. For a zero-knowledge proof system, it is not. Relations that are naturally phrased over $\mathbb{Z}$ or $\mathbb{Z}_q$, such as integer arithmetic or bounds on the individual coefficients of $\vec{\mathbf{s}}$, are not directly captured either. Applications such as anonymous credentials [BLNS23] and verifiable random functions [ESLR23] run into both restrictions at once: they call for exact statements, for instance that $\mathsf{NTT}(\mathbf{s})$ has coefficients in $\{0, \pm 1\}$ or that $\|\vec{\mathbf{s}}\|_\infty \leq \gamma$ for a prescribed γ, and an approximate extractor does not certify these.

Why exact proofs do not fit the framework. Every known exact proof system needs more than three rounds [ENS20, LNS20, LNP22, ESLR23], and this is what makes the step from signatures to proofs hard. In the protocol above, once the shares and the masks are fixed, the whole remainder of the interaction is a *single linear function* of one challenge: one homomorphic evaluation and one threshold decryption suffice, and only $\vec{\mathbf{z}}$ ever leaves the encrypted domain. With $\mu > 1$ challenges the prover must, between consecutive challenges, commit to values that are themselves functions of the shared witness and of the earlier masks, and each has to be produced homomorphically and then opened before the next challenge can be derived. Every additional opening is a fresh function of the secret shares becoming public, and must be argued not to leak. The difficulty of this work lies here, in the multi-round bookkeeping, rather than in the Σ-protocol layer.

Our approach. We keep the shape of the GKS framework but move from sharing a signing key to sharing a witness, and we commit to the shares so that the multi-round structure can be carried out over the commitments. Each party commits to its Shamir share of the witness and to its own masking vector using a BDLOP commitment [BDL+18], and publishes encryptions of both. Writing $\mathbf{t}_4$ and $\mathbf{v}$ for the auxiliary values that the product proof requires, a set $\mathcal{S}$ of at least $\mathtt{t}$ parties proceeds as follows:

1. For $j \in \mathcal{S}$, party j samples a mask $\vec{\mathbf{y}}_j$, sends $\vec{\mathbf{w}}_j := \mathbf{B}_0 \vec{\mathbf{y}}_j$, and publishes encryptions of its share of the commitment randomness and of $\vec{\mathbf{y}}_j$.
2. The parties aggregate $\vec{\mathbf{w}} := \sum_{j \in \mathcal{S}} \vec{\mathbf{w}}_j$, compute $\mathsf{ctx}_{\mathbf{t}_4}$ and $\mathsf{ctx}_{\mathbf{v}}$ homomorphically from the encrypted shares, and threshold-decrypt them.
3. Each party computes the challenge $\mathbf{c}$ by hashing the values opened so far, forms $\mathsf{ctx}_{\vec{\mathbf{z}}}$ with $\vec{\mathbf{z}} := \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}}$ homomorphically, and sends its decryption share.
4. Given the decryption shares, each party recovers $\vec{\mathbf{z}}$ and outputs the proof.

The key insight is that because Shamir sharing and the BDLOP commitment are both linear, the aggregate commitment is the sum of the individual ones, so the parties can run the multi-round protocol over the aggregate without ever reconstructing a share, and the only values that leave the encrypted domain are ones the verifier is meant to see in any case. We stress two consequences. First, the homomorphic evaluation is only ever *quadratic* in the encrypted data, so a leveled scheme such as BGV [BGV12] suffices and no bootstrapping is needed. Second, as in GKS, every challenge is derived by hashing already-decrypted values, so no hash is ever evaluated homomorphically. See Section 5 and Section 6 for details.

Avoiding rejection sampling. The state of the art in exact lattice proofs is the framework of Lyubashevsky, Nguyen, and Plançon [LNP22], packaged in the LaZer library [LNO+26]. We cannot use it, and the obstruction is rejection sampling. With a single prover, one that samples a skewed $\vec{\mathbf{z}}$ aborts and restarts, and since the abort probability is independent of the witness nothing leaks. In the threshold setting this fails twice over. The rejection predicate is a non-linear function of the aggregate $\vec{\mathbf{z}}$, so evaluating it under encryption would reintroduce exactly the circuit-in-FHE cost our framework tries to avoid. If instead each party rejects locally, the acceptance probability would be dramatically reduced, and the pattern of which parties aborted is itself a function of their shares, which an adversary controlling $\mathtt{t} - 1$ parties can exploit. We therefore

need a proof system whose masking can be made rejection-free, by widening the masks and appealing to Hint-MLWE [KLSS23] rather than aborting. This is why we build on ENS, with ALS as a subprotocol, instead of on LNP. Neither ALS nor ENS was designed with thresholdization in mind, so working out the details for both is a substantial part of this paper.

New definitions: threshold commit-and-prove. Thresholdizing an interactive proof forces the security notions to be restated, and the obvious statements are not the right ones. One might hope that $(\mathtt{t}-1)$-zero-knowledge together with ordinary knowledge soundness of the underlying proof would suffice. They do not, because the underlying proof may be malleable, and an adversary that has seen simulated proofs may be able to maul one into a proof for a statement it cannot open. We therefore define threshold commit-and-prove protocols together with weak $(\mathtt{t}-1)$-zero-knowledge and threshold simulation-extractability (Section 4), where the extractor must recover a witness even after the adversary has interacted with honest provers. Establishing these needs one further ingredient. The challenge spaces of ALS and ENS consist of ring elements whose *differences* must be invertible in enough CRT slots, which is not what ordinary $(k_1,\dots,k_\mu)$-special soundness assumes. We combine the predicate special soundness of Aardal et al. [AAB+24] with the weak unique response notion of Ganesh et al. [GOP+25] to obtain simulation-extractability in the random oracle model (ROM).

Simulating the aggregated response. Both ALS and ENS produce a vector $\vec{\mathbf{z}} = \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}}$ whose distribution must be close to a prescribed B-bounded distribution, where $\vec{\mathbf{r}}$ is the commitment randomness and $\vec{\mathbf{y}}$ the mask, and in the threshold setting both are determined by the contributions of all participants. The natural simulation strategy, sampling $\vec{\mathbf{z}}$ from its intended distribution and handing it to every participant, is distinguishable. Writing $\vec{\mathbf{z}} = \vec{\mathbf{z}}_{\mathcal{H}} + \vec{\mathbf{z}}_{\mathcal{V}}$ for the honest and dishonest contributions, an independently sampled $\vec{\mathbf{z}}$ makes $\vec{\mathbf{z}} - \vec{\mathbf{z}}_{\mathcal{V}}$ differ from $\vec{\mathbf{z}}_{\mathcal{H}}$: if the intended distribution is a discrete Gaussian of width $\mathfrak{s} = \sqrt{\mathfrak{s}_{\mathcal{H}}^2 + \mathfrak{s}_{\mathcal{V}}^2}$, then $\vec{\mathbf{z}} - \vec{\mathbf{z}}_{\mathcal{V}}$ has width $\sqrt{\mathfrak{s}^2 + \mathfrak{s}_{\mathcal{V}}^2} > \mathfrak{s}_{\mathcal{H}}$. We instead sample $\vec{\mathbf{z}}_{\mathcal{H}}$ from its intended distribution and add the $\vec{\mathbf{z}}_{\mathcal{V}}$ that we extract from the dishonest participants, taking care that every value derived from $\vec{\mathbf{z}}$ is simulated consistently. Threshold Raccoon [DKM+24] handles this implicitly for the case where $\vec{\mathbf{r}}$ is fixed and shared, and BRaccoon [HMM+26] makes it explicit; we extend the technique to the case where $\vec{\mathbf{y}}$ and $\vec{\mathbf{r}}$ are both determined by all contributions.

Additional technical ingredients. Three further ingredients deserve mention. *Automorphisms.* ALS and ENS both amplify soundness using ring automorphisms, but these interact badly with the homomorphic layer, since applying an automorphism to a ciphertext requires the secret key to be stable under it. We first present automorphism-free variants, and then show that MLWE remains hard when secrets are drawn from the subring fixed by an automorphism (Section 7), which lets us recover the optimization. *A threshold-friendly ENS.* Standard ENS produces $\mathbf{h}$ with $\mathsf{ct}(\mathbf{h}) = 0$; we instead arrange $\mathsf{ct}(\mathbf{h}) = \langle \vec{u}, \vec{v} \rangle$, a public value derivable from the challenge, so that this step becomes linear in the shares and hence thresholdizable, without affecting zero-knowledge or knowledge soundness. *Extraction against a larger bound.* Because each party's mask must hide its own share against an adversary holding $\mathtt{t}-1$ shares, the aggregate $\vec{\mathbf{z}}$ is wider than in the single-prover setting, and extraction has to be argued against that larger bound, so MSIS must remain hard at the corresponding norm.

Efficiency. Our protocols are communication-heavy. Proving runs over several rounds, one for each challenge of the underlying multi-round proof, each requiring a homomorphic evaluation followed by a threshold decryption, and the communication is dominated by the ciphertexts and decryption shares exchanged in each of them. Total communication per party is therefore substantially larger than in the threshold signature

schemes discussed above. What the parties ultimately output, however, is a proof of the same shape as a single-prover ALS or ENS proof, only a factor of $\sqrt{\mathfrak{t}}$ larger. Verification is unchanged, and the verifier neither knows nor cares that the proof was produced by several parties. We regard this as the right trade-off for the setting we target: proof size and verification cost are what an application such as multi-holder anonymous credentials pays on every presentation, whereas the interaction is paid once.

1.3 Related Work

At the core of most lattice-based zero-knowledge proof systems lies the task of proving knowledge of a low-norm vector $\vec{\mathbf{s}}$ satisfying a linear equation

$$\mathbf{A}\vec{\mathbf{s}} = \vec{\mathbf{u}} \pmod{q} \tag{1}$$

where $\mathbf{A}$ is a public matrix and $\vec{\mathbf{u}}$ a public vector over a cyclotomic ring $R := \mathbb{Z}[X]/(X^d + 1)$. Statements of this shape underlie a broad range of applications, from digital signatures [Lyu12] to verifiable encryption [LN17].

Exact lattice proofs. A line of work [ENS20, LNS20, LNP22, ESLR23] lifts the approximation and the restriction to linear relations in the single-prover setting, proving exact shortness and inner-product relations over $\mathbb{Z}$ or $\mathbb{Z}_q$. The techniques are varied: exploiting fully-splitting rings so that exact statements can be proven slot-wise in the CRT decomposition of R_q [ENS20], approximate range proofs based on modular Johnson–Lindenstrauss projections [LNS21, BS23], and reducing the statement at hand to proving that a single ring element has constant coefficient zero [ENS20, LNP22]. The price is interaction: all of them require at least five rounds, although the Fiat–Shamir transform [FS87] makes them non-interactive in the ROM.

Threshold lattice signatures. In the threshold setting, where $\vec{\mathbf{s}}$ is secret-shared among the participants, Equation 1 has been studied extensively in the context of digital signatures, yielding threshold signature schemes under a variety of security notions [ASY22, GKS24, DKM+24, CATZ24, PKN+25]. Underlying all of them is the three-round Σ-protocol recalled above, whose transcripts have the form $(\vec{\mathbf{w}} = \mathbf{A}\vec{\mathbf{y}},\ \mathbf{c},\ \vec{\mathbf{z}} = \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{s}})$. We build on the threshold homomorphic encryption route of Gür, Katz, and Silde [GKS24] and its refinement in Olingo [GHK+26]. All of these certify approximate statements only, and none extends past three rounds.

Generic thresholdization. Boneh et al. [BGG+18] give a universal thresholdizer: encrypt the witness under a threshold fully homomorphic encryption scheme, evaluate the scheme homomorphically, and threshold-decrypt the result. The compiler is generic but non-black-box, since it reaches inside the circuit description of the scheme it is applied to, and for Fiat–Shamir-compiled protocols it yields only a heuristically secure instantiation, because the random oracle must be replaced by a concrete hash function inside that circuit.

The discrete-logarithm setting. A series of works [OB22, GGJ+23, GGJ+25] achieves efficient thresholdization of $(2\mu + 1)$-round protocols with $\mu > 1$. Their techniques do not transfer to lattices: over a prime-order group there is no notion of a short element, and hence none of the bookkeeping on the norms of shared values that dominates the lattice setting.

2 Preliminaries

Notations. A function $\mathtt{f} : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is said to be *negligible* if $\mathtt{f}(x) = x^{-\omega(1)}$. We use $\mathsf{negl}(\cdot)$ to denote an unspecified negligible function and $\lambda \in \mathbb{N}$ to denote the security parameter. For any $a \in \mathbb{Z}_{\geq 1}$, $[a] := \{1, \ldots, a\}$. For a finite set $\mathcal{S}$, we use $|\mathcal{S}|$ to denote its cardinality, $x \leftarrow\!\!\$\ \mathcal{S}$ to denote sampling x uniformly at random from $\mathcal{S}$, and $U(\mathcal{S})$ to denote the uniform distribution over $\mathcal{S}$. For a distribution $\mathcal{D}$, its *collision probability* is $\mathsf{Coll}(\mathcal{D}) = \sum_{x \in \mathrm{Supp}(\mathcal{D})} \Pr[X = x]^2$. For any $q \in \mathbb{N}$ and ring $R \supseteq \mathbb{Z}$, $\mathbb{Z}_q$ denotes the ring of integers modulo q and $R_q = R/qR$. We use **bold** symbols for objects over R or R_q and non-bold symbols for objects over $\mathbb{Z}$ or $\mathbb{Z}_q$; e.g. $\mathbf{a} \in R$ vs. $a \in \mathbb{Z}$, $\vec{\mathbf{a}} \in R^n$ vs. $\vec{a} \in \mathbb{Z}^n$, and $\mathbf{A} \in R^{n \times m}$ vs. $A \in \mathbb{Z}^{n \times m}$. For $\mathbf{a} \in R$ (resp. R_q), $\mathsf{ct}(\mathbf{a}) \in \mathbb{Z}$ (resp. $\mathbb{Z}_q$) denotes its constant coefficient and $\overline{\mathsf{ct}}(\mathbf{a}) \in R$ (resp. R_q) its non-constant part, so that $\mathbf{a} = \mathsf{ct}(\mathbf{a}) + \overline{\mathsf{ct}}(\mathbf{a})$. By default, all vectors are column vectors and $\|\cdot\|$ denotes the ℓ_2-norm. For any $n \times m$ matrix A, $s_1(A) = \sup_{\vec{x} \neq \vec{0}} \frac{\|A\vec{x}\|}{\|\vec{x}\|}$ denotes its second operator norm and $\|A\|_\infty = \max_{i,j} \|A_{i,j}\|_\infty$. We use $\odot$ to denote the element-wise product between vectors of same dimensions, i.e., $[a_1, \ldots, a_n]^\top \odot [b_1, \ldots, b_n]^\top = [a_1 b_1, \ldots, a_n b_n]^\top$.

For interactive Turing machines M, N and an interactive protocol Alg, we write $(\mathsf{out}_{\mathsf{M}}; \mathsf{tr}) \leftarrow \langle \mathsf{M}(x) \leftrightharpoons \mathsf{N}(y) \rangle_{\mathsf{Alg}}$ for the execution of Alg between M on input x and N on input y, where $\mathsf{out}_{\mathsf{M}}$ is M's output and tr the resulting transcript. At times we abuse this notation and let the first component contain the outputs of either or both parties. We simply use $\langle \mathsf{M}(x) \leftrightharpoons \mathsf{N}(y) \rangle$ for an unspecified protocol.

We rely on the first n columns of a matrix $\mathbf{A} \leftarrow\!\!\$\ R_q^{n \times m}$ to be in $\mathrm{GL}_n(R_q)$. Hence, we establish Lemma 2.1, proven in Appendix A.4, which simply extends a known fact about the rank of uniformly random matrices over finite fields.

Lemma 2.1. *Let $q \in \mathbb{Z}$ be a sufficiently large unramified prime with inertia degree $r < d$ in R. Let $k, n, m \in \mathbb{Z}$ such that $0 \leq k < n \leq m$. For a matrix $\mathbf{A} \leftarrow\!\!\$\ R_q^{n \times m}$, let E_1 be the event that the top left $n - k$ by $n - k$ submatrix of $\mathbf{A}$ is in $\mathrm{GL}_{n-k}(R_q)$. Let E_2 be the event that the first n columns of $\mathbf{A}$ form a matrix in $\mathrm{GL}_n(R_q)$ conditioned on E_1. Then $\Pr[\mathsf{E}_1] \approx \Pr[\mathsf{E}_2] = 1 - O(\frac{d}{r} q^{-r})$.*

Coefficient embedding. For any polynomial $\mathbf{f}(X) = \sum_{i=0}^{d-1} a_i X^i \in R$, $\vec{f} = \mathsf{Coeffs}(\mathbf{f}) = (a_0, \ldots, a_{d-1}) \in \mathbb{Z}^d$. The ℓ_∞ and ℓ_p norms of f are defined as follows: $\|\mathbf{f}\|_\infty = \left\|\vec{f}\right\|_\infty = \max_{i \in \{0,\ldots,d-1\}} |a_i|$, $\|\mathbf{f}\|_p = \left\|\vec{f}\right\|_p = \left(\sum_{i=0}^{d-1} |a_i|^p\right)^{1/p}$. In addition, for any vector $\vec{\mathbf{f}} \in R^m$, $\left\|\vec{\mathbf{f}}\right\|_\infty = \max_{i \in [m]} \|\mathbf{f}_i\|_\infty$ and $\left\|\vec{\mathbf{f}}\right\|_p = \left(\sum_{i \in [m]} \|\mathbf{f}_i\|_p^p\right)^{1/p}$. Unless otherwise specified, we will primarily work with the coefficient embedding. Hence, for any $\vec{\mathbf{f}} \in R^m$, $\left\|\vec{\mathbf{f}}\right\|$ denotes the ℓ_2-norm of $\vec{\mathbf{f}}$.

2.1 Number Theoretic Transform

Let q be an unramified prime of inertia degree $r < d$ in R. Let $g = d/r$ and suppose $\zeta \in \mathbb{Z}_q$ is a $2g$-primitive root of unity. Then $R_q \cong \prod_{i \in \mathbb{Z}_{2g}^\times} \mathbb{Z}_q[X]/(X^r - \zeta^i) \cong (\mathbb{F}_{q^r})^g$. The Number Theoretic Transform $\mathsf{NTT} : R_q \to (\mathbb{F}_{q^r})^g$ is a ring isomorphism defined by $\mathsf{NTT}(\mathbf{f}) = (\mathbf{f} \bmod (X^r - \zeta^i))_{i \in \mathbb{Z}_{2g}^\times}$ for all $\mathbf{f} \in R_q$. We use NTT^{-1} to denote its inverse. We refer the reader to the work of Esgin, Nguyen, Seiler [ENS20, Section 2.3] and Seiler [Sei18] for further details.

Lemma 2.2 ([ENS20])**.** *Let $\mathbf{f} = \sum_{i=0}^{d-1} f_i X^i \in R_q$. Then the sum of the NTT components of $\mathbf{f}$, scaled by $1/g$, is equal to a degree $r - 1$ polynomial whose coefficients are the first r coefficients of $\mathbf{f}$. Specifically, $\frac{1}{g} \sum_{i \in \mathbb{Z}_{2g}^\times} (\mathbf{f} \bmod (X^r - \zeta^i)) = \sum_{j=0}^{r-1} f_j X^j$.*

2.2 Lattice Background

An m-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. In this work, we will mainly work with full-rank lattices, i.e., for $\Lambda \subset \mathbb{R}^m$, there exists a set of linearly independent vectors $B = \{\vec{b}_1, \ldots, \vec{b}_m\}$ such that $\Lambda = B\mathbb{Z}^m$, and its dual, which is also a lattice, is defined by the set $\Lambda^* = \{\vec{x} \in \mathbb{R}^m : \forall \vec{y} \in \Lambda, \langle \vec{x}, \vec{y} \rangle \in \mathbb{Z}\}$.

For any vector $\vec{c} \in \mathbb{R}^n$ and positive definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, let $\rho_{\sqrt{\mathbf{\Sigma}},\vec{c}}(\vec{x}) = \exp(-\pi(\vec{x}-\vec{c})^\top \mathbf{\Sigma}^{-1}(\vec{x}-\vec{c}))$ for all $\vec{x} \in \mathbb{R}^n$ be the Gaussian function centered at $\mathbf{c}$ with covariance matrix $\mathbf{\Sigma}$. We use $D_{\sqrt{\mathbf{\Sigma}}}$ to denote the continuous Gaussian distribution $D_{\sqrt{\mathbf{\Sigma}}}(\vec{x}) = (\det \mathbf{\Sigma})^{-1/2} \cdot \rho_{\sqrt{\mathbf{\Sigma}},\vec{0}}(\vec{x})$. If $\mathbf{\Sigma} = \mathfrak{s}^2\mathbf{I}$ with $\mathfrak{s} > 0$, then we write $\rho_{\sqrt{\mathbf{\Sigma}},\vec{c}}(\vec{x}) = \rho_{\mathfrak{s},\vec{c}}(\vec{x})$ and similarly, whenever $\mathbf{\Sigma} = \mathfrak{s}^2\mathbf{I}$, we use $D_{\mathfrak{s}}$. For any $\alpha > 0$, $\Psi_{\leq\alpha} = \{D_{\mathsf{diag}(\vec{r})} : \vec{r} \in (\mathbb{R}_{>0})^d, s_1(\mathsf{diag}(\vec{r})) \leq \alpha\}$.

For any discrete set $\mathcal{X}$ admitting an efficient embedding ϕ to a subset of $\mathbb{R}^n$, the discrete Gaussian distribution over $\mathcal{X}$ centered at $\vec{c} \in \mathbb{R}^n$ with standard deviation $\mathfrak{s} > 0$ is defined as $D_{\mathcal{X},\mathfrak{s},\vec{c}}(x) = \rho_{\mathfrak{s},\vec{c}}(\phi(x))/\rho_{\mathfrak{s},\vec{c}}(\phi(\mathcal{X}))$ for all $x \in \mathcal{X}$. If $\vec{c} = \vec{0}$, we use $D_{\mathcal{X},\mathfrak{s}} = D_{\mathcal{X},\mathfrak{s},\vec{c}}$. In this work, if $\mathcal{X} = R$, then ϕ is the coefficient embedding. For an n-dimensional lattice Λ and a positive real $\varepsilon > 0$, the smoothing parameter [MR04], $\eta_\varepsilon(\Lambda)$, is the smallest real value $\mathfrak{s} > 0$ such that $\rho_{1/\mathfrak{s},\vec{0}}(\Lambda^*) \leq 1 + \varepsilon$.

Lemma 2.3 (Discrete Gaussian Entropy [Lyu12])**.** *For any $0 < \varepsilon \leq 2md e^{-9\pi}$ and $\mathfrak{s} \geq \eta_\varepsilon(R^m)$, we have that: $\max_{\mathbf{x} \in R^m} D_{R^m,\mathfrak{s}}(\mathbf{x}) \leq 2^{-md}$.*

Lemma 2.4 (Discrete Gaussian Convolution [GMPW20, BTT22])**.** *For any $\lambda \in \mathbb{N}$, let $d = d(\lambda)$ and $n = n(\lambda)$ be lattice parameters. Let $\Lambda \subset R^n$ be a full-rank nd-dimensional (module) lattice, $\varepsilon = \mathsf{negl}(\lambda)$, $\mathbf{z}_1, \ldots, \mathbf{z}_m \in R$ be arbitrary elements, $\Lambda + \vec{\mathbf{c}}_1, \ldots, \Lambda + \vec{\mathbf{c}}_m$ be arbitrary cosets of Λ, with $\vec{\mathbf{c}}_i \in K^n_{\mathbb{R}}$, $\mathfrak{s}_1, \ldots, \mathfrak{s}_m \in \mathbb{R}$ such that $\mathfrak{s}_i \geq \sqrt{2} \cdot \eta_\varepsilon(\Lambda) \cdot \max_i \|\mathbf{z}_i\|$. Let $\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_m \in R^n$ such that $\vec{\mathbf{y}}_i \sim D_{\Lambda+\vec{\mathbf{c}}_i,\mathfrak{s}_i}$ and $\vec{\mathbf{y}} = \sum_{i=1}^m \mathbf{z}_i\vec{\mathbf{y}}_i$. Then, the distribution of $\vec{\mathbf{y}}$ is statistically close to $D_{\mathcal{I}\Lambda+\vec{\mathbf{c}},\mathfrak{s}}$, where $\mathcal{I}$ is the ideal generated by the $\mathbf{z}_i$'s, $\vec{\mathbf{c}} = \sum_{i=1}^m \mathbf{z}_i\vec{\mathbf{c}}_i$, and $\mathfrak{s} = \sqrt{\sum_{i=1}^m \|\mathbf{z}_i\|^2\mathfrak{s}_i^2}$. In particular, if $\mathcal{I} = (1)$, then $\vec{\mathbf{y}}$ is statistically close to $D_{\Lambda+\vec{\mathbf{c}},\mathfrak{s}}$.*

Lemma 2.5 (Lattice Tail Bounds)**.** *Let Λ be a full-rank n-dimensional lattice. For any $\mathbf{c} \in \mathbb{R}^n$, $\varepsilon > 0$ and sufficiently large standard deviation $\mathfrak{s} \geq \eta_\varepsilon(\Lambda)$, we have $\Pr_{\mathbf{x} \sim D_{\Lambda,\mathfrak{s},\mathbf{c}}}[\|\mathbf{x} - \mathbf{c}\|_2 > \mathfrak{s}\sqrt{n}] \leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 2^{-n}$ ([MR04, Lemma 4.4]).*

2.3 Lattice Assumptions

Definition 2.1 (MSIS [LS15], adapted)**.** *Let n, m, q be positive integers and $0 < \beta < q$. Then, given $\mathbf{A} \hookleftarrow\$ R_q^{n \times m}$, the Module Short Integer Solution ($\mathsf{MSIS}_{n,m,q,\beta}$) problem asks an efficient adversary $\mathcal{A}$ to find $\vec{\mathbf{z}} \in R^m$ such that $\mathbf{A}\vec{\mathbf{z}} = \vec{\mathbf{0}}$ and $0 < \|\vec{\mathbf{z}}\| \leq \beta$. $\mathcal{A}$ is said to have advantage ϵ_{MSIS} in solving $\mathsf{MSIS}_{n,m,q,\beta}$ if: $\Pr[0 < \|\vec{\mathbf{z}}\| \leq \beta \wedge \mathbf{A}\vec{\mathbf{z}} = \vec{\mathbf{0}} | \mathbf{A} \hookleftarrow\$ R_q^{n \times m}; \vec{\mathbf{z}} \leftarrow \mathcal{A}(\mathbf{A})] \geq \epsilon_{\mathsf{MSIS}}$.*

Definition 2.2 (MLWE [LS15], adapted)**.** *Let n, m, q be positive integers and let $\mathcal{D}$ be a distribution over R. The Module Learning With Errors, denoted $\mathsf{MLWE}_{n,m,q,\mathcal{D}}$, asks a ppt adversary $\mathcal{A}$ to distinguish $(\mathbf{A}, \mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}} \bmod R)$ from $(\mathbf{A}, \vec{\mathbf{u}})$ for any $\mathbf{A} \hookleftarrow\$ R_q^{m \times n}$, $\vec{\mathbf{s}} \leftarrow \mathcal{D}^n$, $\vec{\mathbf{e}} \leftarrow \mathcal{D}^m$, and $\vec{\mathbf{u}} \hookleftarrow\$ R_q^m$. We mainly consider the case where $\chi = D_{R,\mathfrak{s}}$, for some standard deviation $\mathfrak{s} > 0$, and for this case, we simply use $\mathsf{MLWE}_{n,m,q,\mathfrak{s}}$.*

In Appendix A, we consider a version of MLWE where $\vec{\mathbf{e}}$ is sampled from a distribution defined over the torus $\mathbb{R}/\mathbb{Z}$, and we denote this version by $\widehat{\mathsf{MLWE}}$.

Definition 2.3 (Hint-MLWE [KLSS23])**.** *Let $n, m, \ell \in \mathbb{Z}_{>0}$. Let χ, υ be distributions over R^{m+n} and $\mathcal{C}$ be a distribution over R. The Hint-MLWE problem denoted by $\mathsf{Hint\text{-}MLWE}^{\ell,\upsilon,\mathcal{C}}_{n,m,q,\chi}$ states that for any ppt*

adversary $\mathcal{A}$ the following tuple are indistinguishable: $(\mathbf{A}, [\mathbf{I}_m \mid \mathbf{A}]\vec{\mathbf{r}}, (\mathbf{c}_i, \vec{\mathbf{z}}_i)_{i\in[\ell]}) \overset{c}{\approx} (\mathbf{A}, \vec{\mathbf{u}}, (\mathbf{c}_i, \vec{\mathbf{z}}_i)_{i\in[\ell]})$, *where* $\mathbf{A} \leftarrow\!\$\ R_q^{m\times n}, \vec{\mathbf{r}} \leftarrow \chi$, $\vec{\mathbf{u}} \leftarrow\!\$\ R_q^m$, *and for all* $i \in [\ell]$, $\vec{\mathbf{y}}_i \leftarrow \upsilon, \mathbf{c}_i \leftarrow \mathcal{C}, \vec{\mathbf{z}}_i = \mathbf{c}_i \cdot \vec{\mathbf{r}} + \vec{\mathbf{y}}_i$. *If* χ *and* υ *correspond to spherical discrete Gaussian distributions over* R^{m+n} *with standard deviations* $\mathfrak{s}_1$ *and* $\mathfrak{s}_2$, *respectively, we simply use* $\mathsf{Hint\text{-}MLWE}^{\ell,\mathfrak{s}_2,\mathcal{C}}_{n,m,q,\mathfrak{s}_1}$.

We also consider instances of the $\mathsf{Hint\text{-}MLWE}^{\ell,\upsilon,\mathcal{C}}_{n,m,q,\chi}$ assumption with k secrets under the same random matrix. In this setting, an adversary is tasked with distinguishing a tuple $(\mathbf{A}, [\mathbf{I}_m \mid \mathbf{A}]\mathbf{R}, (\mathbf{c}_{i,j}, \vec{\mathbf{z}}_{i,j})_{i\in[k],j\in[\ell]})$ from $(\mathbf{A}, \mathbf{U}, (\mathbf{c}_{i,j}, \vec{\mathbf{z}}_{i,j})_{i\in[k],j\in[\ell]})$ where $\mathbf{R} = [\vec{\mathbf{r}}_1 \mid \cdots \mid \vec{\mathbf{r}}_k]$ with $\vec{\mathbf{r}}_i \leftarrow \chi$, $\mathbf{U} \leftarrow\!\$\ R_q^{m\times k}$ and $\vec{\mathbf{z}}_{i,j} \leftarrow \mathbf{c}_{i,j}\vec{\mathbf{r}}_i + \upsilon$. We denote this as the k-$\mathsf{Hint\text{-}MLWE}^{\ell,\upsilon,\mathcal{C}}_{n,m,q,\chi}$ assumption. By a standard hybrid argument, there exists an efficient reduction from $\mathsf{Hint\text{-}MLWE}^{\ell,\upsilon,\mathcal{C}}_{n,m,q,\chi}$ to k-$\mathsf{Hint\text{-}MLWE}^{\ell,\upsilon,\mathcal{C}}_{n,m,q,\chi}$ that loses a factor of k in the advantage.

2.4 Challenge Distribution

We recall the challenge distribution that was introduced by Attema, Lyubashevsky, and Seiler [ALS20]. Let $C = \{0, \pm 1\}^d \subset R_q$ be the set of ternary polynomials. Let $\mathcal{X}_p$ be the distribution defined by $\mathcal{X}_p(0) = p$ and $\mathcal{X}_p(\pm 1) = (1-p)/2$, with $p \in [0,1]$, and define $\mathcal{C} = \mathcal{X}_p^d$. Thus, $\mathcal{C}$ is a probability distribution on C. Suppose $X^d + 1 \equiv \prod_{j=1}^{h/\ell}(X^{\ell d/h} - \zeta_j) \bmod q$, where $\zeta_j \in \mathbb{Z}_q^\times, 0 < \ell \le h \le d$ such that $\ell | h | d$, and each factor $(X^{\ell d/h} - \zeta_j)$ is not necessarily irreducible over $\mathbb{Z}_q$. In this work, we consider the case $p = 1/3$, i.e., $\mathcal{C}$ is the uniform distribution over $\{0, \pm 1\}^d$. Attema, Lyubashevsky, and Seiler showed that, for appropriate parameters, the maximum probability of $\mathbf{c} \bmod (X^{\ell d/h} - \zeta_j)$, where $\mathbf{c} \leftarrow \mathcal{C}$, is negligible.

Lemma 2.6 ([ALS20], [ENS20], adapted)**.** *For any* $\mathbf{c} \leftarrow \mathcal{C}$, *the coefficients of* $\mathbf{c} \bmod (X^{\ell d/h} - \zeta_j)$ *are independently identically distributed, and*

$$\max_{\mathbf{x}} \Pr\left[\mathbf{c} \bmod (X^{\ell d/h} - \zeta_j) \equiv \mathbf{x}\right] \le \left(\frac{1}{q} + \frac{2h/\ell}{q} \sum_{f \in \mathbb{Z}_q^\times / \langle \zeta_j \rangle} \prod_{i=0}^{h/\ell - 1} \left|\frac{1}{3} + \frac{2}{3}\cos(2\pi f \zeta_j^i / q)\right|\right)^{\ell d/h}$$

where the maximum is taken over polynomials in R_q *of degree less than* $\ell d/h$.

2.5 Shamir Secret Sharing

We use Shamir secret sharing [Sha79] over $\mathbb{Z}_q$. To share a secret p_0 with threshold parameters $(\mathtt{n}, \mathtt{t})$, sample $p(X) = p_0 + \sum_{k=1}^{\mathtt{t}-1} p_k X^k$, where $p_k \leftarrow\!\$\ \mathbb{Z}_q$. Then, for $j \in [\mathtt{n}]$, the j-th share is derived by evaluating $y_j = p(j)$. Given any subset $\{y_i\}_{i\in\mathcal{S}}$ of those shares, with $\mathcal{S} \subseteq [\mathtt{n}]$ and $|\mathcal{S}| \ge \mathtt{t}$, p_0 can be recovered via Lagrange interpolation by computing $p_0 = \sum_{i\in\mathcal{S}} \lambda_i y_i$, where $\lambda_i = \prod_{j\in\mathcal{S}\setminus\{i\}} -j/(i-j)$ is the i-th Lagrange basis polynomial associated with $\mathcal{S}$, evaluated at 0. However, any subset of shares of size at most $\mathtt{t}-1$ perfectly hides p_0. In the cryptographic literature, the λ_is are also known as Lagrange coefficients. In this paper, we use $[\![v]\!]_i$ to denote the i-th share of v, and we abstract the secret sharing by an algorithm $\Pi_{\mathsf{SS}}.\mathsf{Share}$ with the following syntax: $\{[\![v]\!]_i\}_{i\in[\mathtt{n}]} \leftarrow \Pi_{\mathsf{SS}}.\mathsf{Share}(v, (\mathtt{n}, \mathtt{t}))$.

2.6 Threshold Homomorphic Encryption

A threshold homomorphic encryption (tHE) scheme [MW16, BGG+18] is a tuple of ppt algorithms, $\Pi_{\mathsf{tHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{TDec}, \mathsf{Comb})$. The KGen and Enc algorithms operate similarly to a public key encryption scheme where KGen outputs a public key pk, and Enc takes in pk along with a message m to produce a ciphertext ctx. tHE is generalization of a homomorphic encryption scheme [BGV12] meaning it is constructed with respect to a family of functions $\mathcal{F}$, such that for any $f \in \mathcal{F}$, $\mathsf{ctx} = \mathsf{Enc}(\mathsf{pk}, m)$, $\mathsf{ctx}' = \mathsf{Enc}(\mathsf{pk}, f(m))$ can be computed from ctx using the function, Eval. tHE diverges from HE in that KGen also outputs

$\mathtt{n}$ shares of a secret key, $\{\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathtt{n}\}$ such that each sk_i can be used to produce a decryption share, $\mathsf{ds}_i = \mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx})$. If $\mathsf{ctx} = \mathsf{Enc}(\mathsf{pk}, m)$, and decryption shares are produced from $\mathtt{t}$ of the $\mathtt{n}$ keys, then Comb can be used to combine the shares to recover m. We will use $\vec{\mathsf{ctx}}_{\vec{m}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{m})$ to denote the vector obtained by individually encrypting each component of the message vector, $\vec{m}$.

In addition to being *correct* (described above), a tHE scheme should be *compact*, meaning that the size of a ciphertext depends only on the security parameter. Further, a tHE scheme has *semantic security* if any ppt adversary, given a ciphertext of a message along with $\mathtt{t} - 1$ shares of the secret key, learns nothing about the message. For this work, semantic security is not enough, since the adversary learns decryptions of ciphertexts with inputs generated by honest parties. Thus, we need to be able to simulate the decryption shares of honest parties given only the adversarial inputs and the result of the function to be evaluated (notably, the simulation must be independent of the honest party's inputs to the evaluated function). To capture this requirement, we use a definition of *simulation security* similar to [JRS17].

We review the formal definitions of a tHE scheme along with the informally mentioned security properties in Appendix C. We review a construction of threshold homomorphic encryption in Appendix G.1.

2.7 BDLOP Commitment

Suppose $q \in \mathbb{Z}$ is an unramified prime of inertia degree $r > 0$ in R, and let $g = d/r$. Then, $X^d + 1 \equiv \prod_{i=1}^{g} \Phi_i(X) \bmod q$, where $\Phi_i(X) \in \mathbb{Z}_q[X]$ is irreducible of degree r. Given a distribution χ over R and matrices $\mathbf{B}_0 \leftarrow\!\$\ R_q^{n \times m}$ and $\mathbf{B}_1 \leftarrow\!\$\ R_q^{k \times m}$, the BDLOP commitment [BDL+18] allows to commit to any messages $\vec{\mathbf{m}} \in R_q^k$ as follows: (1) sample $\vec{\mathbf{r}} \leftarrow \chi^m$, and (2) compute $\vec{\mathbf{t}} := \begin{bmatrix} \vec{\mathbf{t}}_0 \\ \vec{\mathbf{t}}_1 \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \end{bmatrix} \vec{\mathbf{r}} + \begin{bmatrix} \mathbf{0}^n \\ \vec{\mathbf{m}} \end{bmatrix}$. This commitment is binding under the MSIS assumption and hiding under the MLWE assumption (see [BDL+18], [ENS20]). In addition, given a challenge distribution $\mathcal{C} \subseteq R_q$ such that for any $\mathbf{c} \leftarrow \mathcal{C}$, $\|\mathbf{c}\|_1 \leq T$, and a distribution υ over R such that for any $\vec{\mathbf{y}} \leftarrow \upsilon^m$ we have $\|\vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}}\|_2 \leq \beta$, the commitment admits a 3-round HVZK (relaxed) proof of knowledge of a randomness-message pair. In brief, an honest proof is a triple $\pi := (\vec{\mathbf{w}} = \mathbf{B}_0\vec{\mathbf{y}}, \mathbf{c}, \vec{\mathbf{z}} = \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}})$. The witness extraction process devised by Baum et al. [BDL+18] requires $\bar{\mathcal{C}} := \{\mathbf{a} - \mathbf{b} : \mathbf{a}, \mathbf{b} \leftarrow \mathcal{C}\} \subseteq R_q^\times$, which is easily achieved when r is large [LS18, Corollary 1.2]. However, to handle cases where r is small, e.g., $r = 1$ (which will be our setting in Section 6), Attema, Lyubashevsky, and Seiler [ALS20, Section 4] devised an extraction process that allows to recover a weak opening (see Definition 2.4) and showed that commitment is still binding with respect to weak openings under the MSIS assumption.

Definition 2.4 (Weak Opening [ALS20], adapted)**.** *A weak opening for a* BDLOP *commitment* $(\vec{\mathbf{t}}_0, \vec{\mathbf{t}}_1) := \vec{\mathbf{t}} \in R_q^{n+k}$ *is a tuple* $(\bar{\mathbf{c}}_1, \ldots, \bar{\mathbf{c}}_g, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$*, where* $\bar{\mathbf{c}}_i \in R_q$*,* $\vec{\mathbf{r}}^* \in R_q^m$*, and* $\vec{\mathbf{m}}^* \in R_q^k$*, such that* $\mathbf{B}_0\vec{\mathbf{r}}^* = \vec{\mathbf{t}}_0$*,* $\mathbf{B}_1\vec{\mathbf{r}}^* + \vec{\mathbf{m}}^* = \vec{\mathbf{t}}_1$*, and for all* $i \in [g]$*,* $\|\bar{\mathbf{c}}_i\vec{\mathbf{r}}^*\|_2 \leq 2\beta$*,* $\|\bar{\mathbf{c}}_i\|_1 \leq 2T$ *and* $\bar{\mathbf{c}}_i \not\equiv \mathbf{0} \bmod \Phi_i(X)$.

Lemma 2.7 ([ALS20])**.** *Under the hardness of the* $\mathsf{MSIS}_{n,m,q,8T\beta}$ *assumption, the* BDLOP *commitment is binding with respect to weak openings.*

Extracting a Weak Opening from BDLOP. Assume that $\bar{\mathcal{C}} \subset R_q^\times$, i.e., for any $\bar{\mathbf{c}} \in \bar{\mathcal{C}}$ we have $\bar{\mathbf{c}} \not\equiv \mathbf{0} \bmod \Phi_i(X)$ for all $i \in [g]$. Given a commitment $(\vec{\mathbf{t}}_0, \vec{\mathbf{t}}_1)$ with two proofs of opening $\pi_1 := (\vec{\mathbf{w}}, \mathbf{c}_1, \vec{\mathbf{z}}_1)$, $\pi_2 := (\vec{\mathbf{w}}, \mathbf{c}_2, \vec{\mathbf{z}}_2)$ sharing the first message and having distinct challenges, a weak opening can be extracted as follows: (1) since $\mathbf{B}_0\vec{\mathbf{z}}_j = \vec{\mathbf{w}} + \mathbf{c}_j\vec{\mathbf{t}}_0$, by setting $\vec{\mathbf{z}}_1^* = \vec{\mathbf{z}}_1 - \vec{\mathbf{z}}_2$ and $\bar{\mathbf{c}}_1 = \mathbf{c}_1 - \mathbf{c}_2$, we have $\mathbf{B}_0(\bar{\mathbf{c}}_1^{-1}\vec{\mathbf{z}}_1^*) = \vec{\mathbf{t}}_0$, (2) therefore, letting $\vec{\mathbf{r}}^* = \bar{\mathbf{c}}_1^{-1}\vec{\mathbf{z}}_1^*$, it follows that $(\bar{\mathbf{c}}_1, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^* = \vec{\mathbf{t}}_1 - \mathbf{B}_1\vec{\mathbf{r}}^*)$ is a candidate weak opening. Now, to ensure that $(\vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$ is unique, we can use a third proof $\pi_3 := (\vec{\mathbf{w}}, \mathbf{c}_3, \vec{\mathbf{z}}_3)$ whose challenge is distinct from the first two challenges and compute $\mathbf{B}_0(\bar{\mathbf{c}}_2^{-1}\vec{\mathbf{z}}_2^*)$, where $\vec{\mathbf{z}}_2^* = \vec{\mathbf{z}}_1 - \vec{\mathbf{z}}_3$ and $\bar{\mathbf{c}}_2 = \mathbf{c}_1 - \mathbf{c}_3$.

If $\mathbf{B}_0(\bar{\mathbf{c}}_1^{-1}\vec{\mathbf{z}}_1^*) \neq \mathbf{B}_0(\bar{\mathbf{c}}_2^{-1}\vec{\mathbf{z}}_2^*)$, we obtain an MSIS solution. In addition, given $\vec{\mathbf{r}}^*$, and π_1, π_2, π_3, we can extract $\vec{\mathbf{y}}^*$ such that for each transcript π_j, we have $\vec{\mathbf{z}}_j = \vec{\mathbf{y}}^* + \mathbf{c}_j\vec{\mathbf{r}}^*$. This follows by simply setting $\vec{\mathbf{y}}^* = \vec{\mathbf{z}}_1 - \mathbf{c}_1\vec{\mathbf{r}}^* = \vec{\mathbf{z}}_2 - \mathbf{c}_2\vec{\mathbf{r}}^* = \vec{\mathbf{z}}_3 - \mathbf{c}_3\vec{\mathbf{r}}^*$. Note that if any of those equalities fails to hold, then we can find an MSIS solution. We refer the reader to the work of Attema, Lyubashevsky, and Seiler [ALS20, Section 4] for further details.

3 Proof Systems

We recall some notions on public-coin interactive proofs, as well as non-interactive proofs in the ROM. In this work, we assume that provers run in polynomial time.

A public-coin $(2\mu+1)$-round proof $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for an NP relation $\mathcal{R}$ is an interactive protocol between P and V where they interact in a round-robin fashion such that P sends the first and last messages over $\mu+1$ interactions. It is said to be Honest-Verifier Zero-Knowledge (HVZK) if there exists a ppt simulator Sim that can simulate the view of an honest V without knowledge of P's inputs. Its Fiat–Shamir transform is said to be adaptively knowledge-sound with knowledge error $\kappa(\lambda, Q)$, where $\lambda, Q \in \mathbb{N}$, if there exists an expected polynomial-time extractor Ext that, given black-box rewindable access to any ppt prover making at most Q random-oracle queries and producing a valid proof on a statement $\mathbb{x}$, fails to extract a valid witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$ with probability at most $\kappa(\lambda, Q)$. We refer the reader to Appendix B for formal definitions.

Definition 3.1 (Weak Simulation Extractability [FKMV12])**.** *A NIZK $\Pi^{\mathcal{R}}_{\mathsf{NIZK}} = (\mathsf{Setup}, \mathsf{Prove}^H, \mathsf{Verify}^H)$ for an NP relation $\mathcal{R}$ in the ROM is weakly simulation-extractable if there exists a* ppt *simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ and an expected polynomial-time extractor Ext such that, for any* ppt *adversary $\mathcal{A}$, there exists a negligible function negl such that, for any $\lambda \in \mathbb{N}$, the extractor Ext, given black-box rewindable access to $\mathcal{A}$, satisfies the following:*

$$\Pr\left[\begin{array}{r}(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Sim}_1(1^\lambda); (\mathbb{x}^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{Sim}_2^H(\mathsf{pp},\mathsf{st},\cdot)}(1^\lambda, \mathsf{pp});\\ \mathbb{w} \leftarrow \mathsf{Ext}^{\mathcal{A}}(\mathsf{pp}, \mathsf{st}, \mathbb{x}^*, \pi^*, Q_{\mathsf{Sim}}); (\mathbb{x}^*, \pi^*) \notin Q_{\mathsf{Sim}} :\\ \mathsf{Verify}(\mathsf{crs}, \mathbb{x}^*, \pi^*) = 1 \wedge (\mathbb{x}^*, \mathbb{w}) \notin \mathcal{R}\end{array}\right] \leq \mathsf{negl}(\lambda)$$

where Sim_2 only takes (true or false) statements as input and Q_{Sim} represents the set of statement-proof pairs produced by Sim_2 according to $\mathcal{A}$'s queries.

Definition 3.2 (Tree of Transcripts ([AFK22]))**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a $(2\mu+1)$-round interactive proof for a relation $\mathcal{R}$, and let $k_1, \ldots, k_\mu \in \mathbb{N}$. A $(k_1, \ldots, k_\mu)$-tree of transcripts for Π on statement $\mathbb{x}$ is a set of $K = \prod_{i=1}^{\mu} k_i$ transcripts arranged in the following tree structure: (1) the nodes in the tree correspond to P's messages and the edges correspond to V's challenges, (2) each node at depth i has k_i children corresponding to k_i distinct challenges, and (3) each transcript is in one-to-one correspondence with a path from the root to a leaf node.*

In addition, let $\mathscr{T}_{\mu+1}$ denote the set of all possible accepting transcripts for Π, and $\mathscr{T}_m$, with $m \in [\mu]$, denote the set of all possible $(1, \ldots, 1, k_m, \ldots, k_\mu)$-tree of transcripts such that for any $t \in \mathscr{T}_m$, each transcript $\mathsf{tr} \in t$ is valid.

Definition 3.3 (Predicate System [AAB+24], adapted)**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin $(2\mu+1)$-round interactive proof for a relation $\mathcal{R}$ with challenge spaces $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$. Suppose Π admits a $\vec{k}$-tree of transcripts, where $\vec{k} \in \mathbb{N}^\mu$. A predicate system $\mathfrak{C}$ for Π with respect to the $\vec{k}$-tree structure is a collection of poly-time predicates such that, at depth $m \in [\mu]$, there are k_m challenge predicates $\mathfrak{C}^{\mathsf{chal}}_{m,1}, \ldots, \mathfrak{C}^{\mathsf{chal}}_{m,k_m}$, with $\mathfrak{C}^{\mathsf{chal}}_{m,i} : \mathcal{C}^i_m \to \{0,1\}$, and one commitment predicate, which is composed of two predicates: $\mathfrak{C}^{\mathsf{prop}}_m : \mathscr{T}^{(k_m-1)}_{m+1} \to \{0,1\}$ and $\mathfrak{C}^{\mathsf{bind}}_m : \mathscr{T}^{(k_m-1)}_{m+1} \times \mathscr{T}_{m+1} \to \{0,1\}$.*

Furthermore, let $\mathfrak{C}_m \subseteq \mathfrak{C}$ denote the collection of all predicates from depth μ up to depth m. Clearly, $\mathfrak{C} = \mathfrak{C}_1$. A partial tree of transcripts $t = (t_1, \ldots, t_{k_m}) \in \mathscr{T}_m$, with $t_i \in \mathscr{T}_{m+1}$, is said to satisfy $\mathfrak{C}_m$, denoted $\mathfrak{C}_m(t) = 1$, if and only if

$$\left(\bigwedge_{i\in[k_m]} (\mathfrak{C}_{m+1}(t_i) = 1) \wedge \left(\mathfrak{C}^{\mathsf{chal}}_{m,i}(c_m^{(t_1)}, \ldots, c_m^{(t_i)}) = 1\right)\right) \wedge (\mathfrak{C}^{\mathsf{prop}}_m(t_1, \ldots, t_{k_m-1}) = 1) \wedge \left(\mathfrak{C}^{\mathsf{bind}}_m(t) = 1\right)$$

where $c_m^{(t_i)}$ represents the unique challenge at depth m for t_i.

Intuitively, a predicate system is used to enforce properties that are needed to ensure that an extractor can extract a valid witness from a tree of transcripts.

Definition 3.4 (Challenge Predicate's Failure Density [AAB+24])**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin $(2\mu+1)$-round interactive proof for a relation $\mathcal{R}$ with challenge spaces $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$. Suppose Π admits a $\vec{k}$-tree of transcripts, with $\vec{k} \in \mathbb{N}^\mu$, and a predicate system $\mathfrak{C}$. The failure density of a challenge predicate $\mathfrak{C}^{\mathsf{chal}}_{m,i} \in \mathfrak{C}$, with $m \in [\mu]$ and $i \in [k_m]$, denoted $p^{\mathsf{chal}}_{m,i}$, is defined as follows:*

$$p^{\mathsf{chal}}_{m,i} = \sup_{c_1,\ldots,c_{i-1}} \frac{\left|\{c_i \in \mathcal{C}_m \setminus \{c_1, \ldots, c_{\ell-1}\} : \mathfrak{C}^{\mathsf{chal}}_{m,i}(c_1, \ldots, c_i) = 0\}\right|}{|\mathcal{C}_m|}$$

where the supremum is taken over all tuples $(c_1, \ldots, c_{i-1}) \in \mathcal{C}_m^{(i-1)}$ of distinct challenges satisfying $\mathfrak{C}^{\mathsf{chal}}_{m,i-1}(c_1, \ldots, c_{i-1}) = 1$.

Definition 3.5 (Commitment Predicate's Failure Density [AAB+24])**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin $(2\mu+1)$-round interactive proof for a relation $\mathcal{R}$ [2] with challenge spaces $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$. Suppose Π admits a $\vec{k}$-tree of transcripts, with $\vec{k} \in \mathbb{N}^\mu$, and a predicate system $\mathfrak{C}$. The failure density of a commitment predicate $(\mathfrak{C}^{\mathsf{prop}}_m, \mathfrak{C}^{\mathsf{bind}}_m) \in \mathfrak{C}$, with $m \in [\mu]$ and $i \in [k_m]$, denoted p^{com}_m, is defined as follows:*

$$p^{\mathsf{com}}_m = \sup_{t_1,\ldots,t_{k_m-1}} \frac{\left|\left\{ c_m^{(t)} \in \mathcal{C}_m \setminus \begin{Bmatrix} c_m^{(t_1)}, \ldots, \\ c_m^{(t_{k_m-1})} \end{Bmatrix} : \begin{array}{l} t \in \mathscr{T}_{m+1}, (t_1, \ldots, t_{k_m-1}, t) \in \mathscr{T}_m, \\ \mathfrak{C}^{\mathsf{chal}}_{m,k_m}(c_m^{(t_1)}, \ldots, c_m^{(t_{k_m-1})}, c_m^{(t)}) = 1, \\ \mathfrak{C}^{\mathsf{bind}}_m(t_1, \ldots, t_{k_m-1}, t) = 1, \\ \mathfrak{C}^{\mathsf{prop}}_m(t_1, \ldots, t_{k_m-1}) = 0, \mathfrak{C}_{m+1}(t) = 1, \end{array} \right\}\right|}{|\mathcal{C}_m|}$$

where the supremum is taken over all $(t_1, \ldots, t_{k_m-1}) \in \mathscr{T}_{m+1}^{(k_m-1)}$ such that the $c_m^{(t_j)}$ are distinct, and for any $i \in [k_m - 1]$, $\mathfrak{C}^{\mathsf{chal}}_{m,i}(c_m^{(t_1)}, \ldots, c_m^{(t_i)}) = 1$ and $\mathfrak{C}_{m+1}(t_i) = 1$.

Intuitively, the failure density of a commitment predicate bounds the fraction of challenges that allow an adversary to produce a valid proof with a committed witness that fails to satisfy a property checked by the proof.

Definition 3.6 (Predicate Special Soundness [AAB+24])**.** *Let $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ be a public-coin $(2\mu+1)$-round interactive proof for a relation $\mathcal{R}$. Π is said to be $(\vec{k}, \mathfrak{C})$-predicate-special-sound, with $\vec{k} \in \mathbb{N}^\mu$ and $\mathfrak{C}$ a predicate system, if there exists a poly-time algorithm that, given any statement $\mathbb{x}$ and $\vec{k}$-tree of transcripts $t_{\mathbb{x}}$ such that $\mathfrak{C}(t_{\mathbb{x}}) = 1$, outputs a witness $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$.*

[2] In [AAB+24] the NP relation $\mathcal{R}$ is lifted to a relation $\mathcal{R}' = \mathcal{R} \cup \mathcal{R}_{\mathsf{Com}}$, where $\mathcal{R}_{\mathsf{Com}}$ is defined by the binding property of the commitment. In our case, we omitted $\mathcal{R}_{\mathsf{Com}}$ to simplify the exposition and because it is always associated with a hardness assumption, such as the hardness of MSIS.

Theorem 3.1 ([AAB+24, Theorem 5.1] adapted)**.** *Let* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *be a public-coin* $(2\mu+1)$*-round interactive proof for a relation* $\mathcal{R}$ *with challenge spaces* $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$*. Suppose* Π *is* $(\vec{k}, \mathfrak{C})$*-predicate-special-sound. Then the adaptive Fiat–Shamir transformation of* Π *in the ROM is adaptively knowledge-sound with knowledge error at most* $2(Q+1)\sum_{i=1}^{\mu} \max\left((k_i-1)/|\mathcal{C}_i|, p_i^{\mathsf{com}} + \sum_{j=1}^{k_i} p_{i,j}^{\mathsf{chal}}\right)$*, where* Q *is the number of random oracle queries made by* P*. Furthermore, the extractor makes* $K + Q(K-1)$ *queries to* P *in expectation, where* $K = \prod_{i=1}^{\mu} k_i$*.*

Definition 3.7 (Weak Unique Response [GOP+25], adapted)**.** *Let* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *be a public-coin* $(2\mu+1)$*-round interactive proof for a relation* $\mathcal{R}$*. Suppose* Π *is HVZK. The Fiat–Shamir transformation of* Π *is said to have weak unique response (WUR) with respect to the canonical simulator* $\mathsf{Sim} = (\mathsf{SimSetup}, \mathsf{SimProve}, \mathsf{SimRO})$ *if, for any* ppt *adversary* $\mathcal{A}$ *with oracle access to* SimRO*, there exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{r} \mathsf{crs} \leftarrow \mathsf{SimSetup}(1^\lambda); (\mathbb{x}, \mathsf{st}_{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathsf{SimRO}}(1^\lambda, \mathsf{crs}); \\ \tilde{\pi} \leftarrow \mathsf{SimProve}(\mathsf{crs}, \mathbb{x}); \pi \leftarrow \mathcal{A}^{\mathsf{SimRO}}(\mathsf{st}_{\mathcal{A}}, \tilde{\pi}); \\ \text{Parse } (\tilde{a}_1, \ldots, \tilde{c}_\mu, \tilde{a}_{\mu+1}) := \tilde{\pi},\ (a_1, \ldots, c_\mu, a_{\mu+1}) := \pi : \\ (\mathsf{V}^{\mathsf{SimRO}}(\mathsf{crs}, \mathbb{x}, \pi) = 1) \wedge \\ \left(\exists i \in [\mu] : ((\tilde{a}_1, \ldots, \tilde{a}_i, \tilde{c}_i) = (a_1, \ldots, a_i, c_i)) \wedge (\tilde{a}_{i+1} \neq a_{i+1})\right) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

Ganesh et al. [GOP+25] showed that the Fiat–Shamir transformation of any public-coin interactive proof satisfying WUR and special soundness is weakly simulation-extractable in the ROM. The following lemma extends their result [GOP+25, Lemma 3.2] to the setting of predicate special soundness, and the proof follows by replacing $\vec{k}$-special soundness with predicate special soundness in a black-box way.

Lemma 3.1. *Let* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *be a public-coin* $(2\mu+1)$*-round interactive proof for a relation* $\mathcal{R}$ *with challenge spaces* $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$*. Suppose* Π *is HVZK and* $(\vec{k}, \mathfrak{C})$*-predicate-special-sound for some* $\vec{k} \in \mathbb{N}^\mu$ *and predicate system* $\mathfrak{C}$*. If the Fiat–Shamir transformation of* Π *has WUR, then it is weakly simulation-extractable with knowledge error at most* $Q_P \cdot \epsilon_{\mathsf{WUR}} + \kappa(Q_H)$*, where* Q_P *and* Q_H *denote the number of (simulated) proof and random oracle queries made by* P*, respectively,* ϵ_{WUR} *is the probability that* P *breaks the weak unique response property of* Π*'s Fiat–Shamir transform, and* $\kappa(Q_H)$ *is the knowledge error from Theorem 3.1.*

4 Threshold Commit-and-Prove

In this section, we provide a definition for the notion of threshold commit-and-prove, which extends the commit-and-prove paradigm [CLOS02],[EG14] to the threshold setting. Our definition is motivated by a setting where a client holding a witness $\mathbb{w}$ for a statement $\mathbb{x}$ in an NP relation $\mathcal{R}$ secret shares its witness among a set of provers, who can then execute a commit-and-prove protocol on the client's behalf. We only consider static security, where an adversary chooses provers that are corrupt before the beginning of the interaction.

Definition 4.1. *Given a thresholdizable commitment* $\Pi_{\mathsf{Com}} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$*, a* $(\mathtt{t}, \mathtt{n})$*-threshold commit-and-prove system for an NP relation* $\mathcal{R}$ *is a tuple of* ppt *algorithms* $\Pi_{\mathsf{tCP}} = (\mathsf{Setup}, \mathsf{WitnessShare}, \mathsf{Commit}, \mathsf{Prove}, \mathsf{Verify})$ *executed among a client* C *and* $\mathtt{n}$ *provers* $\mathsf{P}_1, \ldots, \mathsf{P}_{\mathtt{n}}$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda, \mathtt{n}, \mathtt{t}) \to (\mathsf{pp}, \{\mathsf{paux}_i\}_{i\in[\mathtt{n}]})$*: This algorithm takes as inputs the security parameter* $\lambda \in \mathbb{N}$ *and the threshold parameters* $\mathtt{t}, \mathtt{n} \in \mathbb{N}$ *such that* $0 < \mathtt{t} \leq \mathtt{n}$*. It generates some system parameters* sysp *and private auxiliary information* $\{\mathsf{paux}_i\}_{i\in[\mathtt{n}]}$*. Then, it executes* $\mathsf{ck} \leftarrow \Pi_{\mathsf{Com}}.\mathsf{Setup}(1^\lambda)$ *and outputs the public parameters* $\mathsf{pp} = \mathsf{sysp} \| \mathsf{ck}$ *and* paux_i *to each prover* P_i*.*

- $\mathsf{WitnessShare}(\mathsf{pp}, \mathbb{x}, \mathbb{w}) \to (\{\mathsf{pwit}_i\}_{i\in[\mathtt{n}]}, \mathsf{aux}_w)$*: This algorithm is executed by a client* C*. It takes as inputs* pp *and a statement-witness pair* $(\mathbb{x}, \mathbb{w})$*. It outputs a "partial" witness* pwit_i *to each prover* P_i *and some public auxiliary information* aux_w*.*

- $\mathsf{Commit}(\mathsf{pp}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w) \to (c, \{\mathsf{pr}_i\}_{i\in\mathcal{S}})$*: This is an interactive algorithm executed by the provers. It takes as inputs* pp*, a set* $\mathcal{S} \subseteq [\mathtt{n}]$ *such that* $|\mathcal{S}| \geq \mathtt{t}$*, a pair of private auxiliary information-partial witness* $(\mathsf{paux}_i, \mathsf{pwit}_i)$ *from each prover* $\mathsf{P}_i \in \mathcal{S}$*, and the auxiliary information* aux_w*. It outputs a commitment* c *and a partial opening* pr_i *to each prover* $\mathsf{P}_i \in \mathcal{S}$*.*

- $\mathsf{Prove}(\mathsf{pp}, \mathbb{x}, c, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w) \to \pi$*: This is an interactive algorithm executed by the provers. It takes as inputs* pp*, a statement* $\mathbb{x}$*, a commitment* c*, a set* $\mathcal{S} \subseteq [\mathtt{n}]$ *such that* $|\mathcal{S}| \geq \mathtt{t}$*, a triple* $(\mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i)$ *from each prover* $\mathsf{P}_i \in \mathcal{S}$*, and the auxiliary information* aux_w*. It outputs a proof* π*.*

- $\mathsf{Verify}(\mathsf{pp}, \mathbb{x}, c, \pi) \to 1/0$*: This deterministic algorithm takes as input* pp*, a statement* $\mathbb{x}$*, a commitment* c*, and a proof* π*. It outputs* 1 *if* π *is valid w.r.t to* $\mathbb{x}$ *and* c*. Otherwise, it outputs* 0*.*

Remark 4.1 (Private Auxiliary Information). The private auxiliary information output by Setup in Definition 4.1 is not necessary to produce a commitment-proof pair, unlike the prover's secret key in the (threshold) designated-prover setting [KNYY19]. Indeed, in our case, one should think of it as seeds to generate correlated randomness or keys for threshold decryption.

Definition 4.2 (Completeness)**.** *For any* $\lambda, \mathtt{t}, \mathtt{n} \in \mathbb{N}$ *such that* $\mathtt{n} \geq \mathtt{t}$*, any* $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$*, and any subset* $\mathcal{S} \subseteq [\mathtt{n}]$ *of honest provers such that* $|\mathcal{S}| \geq \mathtt{t}$*, we have*

$$\Pr\left[\begin{array}{r}(\mathsf{pp}, \{\mathsf{paux}_i\}_{i\in[\mathtt{n}]}) \leftarrow \Pi_{\mathsf{tCP}}.\mathsf{Setup}(1^\lambda, \mathtt{n}, \mathtt{t});\\ (\{\mathsf{pwit}_i\}_{i\in[\mathtt{n}]}, \mathsf{aux}_w) \leftarrow \mathsf{WitnessShare}(\mathsf{pp}, \mathbb{x}, \mathbb{w});\\ (c, \{\mathsf{pr}_i\}_{i\in\mathcal{S}}) \leftarrow \Pi_{\mathsf{tCP}}.\mathsf{Commit}(\mathsf{pp}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w);\\ \pi \leftarrow \mathsf{Prove}(\mathsf{pp}, \mathbb{x}, c, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w) :\\ \mathsf{Verify}(\mathsf{pp}, \mathbb{x}, c, \pi) \neq 1\end{array}\right] \leq \mathsf{negl}(\lambda)$$

Definition 4.3 (Weak $(\mathtt{t}-1)$-Zero-knowledge)**.** *A threshold commit-and-prove scheme* Π_{tCP} *satisfies weak* $(\mathtt{t}-1)$*-zero-knowledge under static corruption if for any* ppt *distinguisher* D *and* ppt *adversary* $\mathcal{A}$ *there exists a* ppt *simulator* $\mathsf{Sim} = (\mathsf{SimSetup}, \mathsf{SimWitShare}, \mathsf{SimCom}, \mathsf{SimProve})$ *and a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*, we have:*

$$\left|\begin{array}{r}\Pr\left[\mathsf{Thrsh\text{-}ZK\text{-}Exp}^{\mathsf{D},\mathcal{A},\mathsf{Sim},\Pi_{\mathsf{tCP}}}(1^\lambda, 0) = 1\right]\\ -\Pr\left[\mathsf{Thrsh\text{-}ZK\text{-}Exp}^{\mathsf{D},\mathcal{A},\mathsf{Sim},\Pi_{\mathsf{tCP}}}(1^\lambda, 1) = 1\right]\end{array}\right| \leq \mathsf{negl}(\lambda)$$

where Thrsh-ZK-Exp *is defined in Fig. 1. Furthermore,* SimProve *and* SimCom *are given black-box access to* $\mathcal{A}$*.* Π_{tCP} *is said to be* semi-honest *weak* $(\mathtt{t}-1)$*-zero-knowledge if* $\mathcal{A}$ *is semi-honest.*

Remark 4.2. Our (weak) $(\mathtt{t}-1)$-ZK definition requires the statement-witness pair provided by the distinguisher to be in the relation. This weakens the $(\mathtt{t}-1)$-ZK notion of Ozdemir and Boneh [OB22], where the pair need not be in the relation but the simulator is given a bit indicating whether it is. Garg et al. [GGJ+25] showed that for some ZK proof systems, care is needed when the pair is not in the relation, since the protocol output might reveal information about honest provers' inputs. This is the case for the schemes we consider (see Sections 5 and 6). However, as hinted by Garg et al. in the context of ZK proof systems, any threshold commit-and-prove scheme satisfying our weak notion can be upgraded to one satisfying a stronger definition

$\mathsf{Thrsh\text{-}ZK\text{-}Exp}^{\mathsf{D},\mathcal{A},\mathsf{Sim},\Pi_{\mathsf{tCP}}}(1^\lambda, b)$

$(\mathtt{n}, \mathtt{t}, \mathsf{st}_{\mathcal{A},1}) \leftarrow \mathcal{A}(1^\lambda)$

if $b = 0$: $(\mathsf{pp}, \{\mathsf{paux}_i\}_{i\in[\mathtt{n}]}) \leftarrow \Pi_{\mathsf{tCP}}.\mathsf{Setup}(1^\lambda, \mathtt{n}, \mathtt{t})$

else if $b = 1$: $(\mathsf{pp}, \{\mathsf{paux}_i\}_{i\in[\mathtt{n}]}, \mathsf{td}) \leftarrow \mathsf{Sim.SimSetup}(1^\lambda, \mathtt{n}, \mathtt{t})$

$(\mathcal{S}, \mathcal{V}, \mathsf{st}_{\mathcal{A},2}) \leftarrow \mathcal{A}(\mathsf{st}_{\mathcal{A},1}, \mathsf{pp})$

if $\mathcal{S} \not\subset [\mathtt{n}] \vee \mathcal{V} \not\subset \mathcal{S} \vee |\mathcal{V}| \geq \mathtt{t}$: **abort**

$(\mathbb{x}, \mathbb{w}, \mathsf{st}_{\mathcal{A},3}) \leftarrow \mathsf{D}(\mathsf{st}_{\mathcal{A},2})$

if $(\mathbb{x}, \mathbb{w}) \notin \mathcal{R}$: **abort**

if $b = 0$:

$(\{\mathsf{pwit}_i\}_{i\in[\mathtt{n}]}, \mathsf{aux}_w) \leftarrow \Pi_{\mathsf{tCP}}.\mathsf{WitnessShare}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$

$\mathsf{in}_{\mathsf{Com}} := (\mathsf{pp}, \mathcal{S}, \mathsf{aux}_w)$

$(c, \{\mathsf{pr}_i\}_{i\in\mathcal{S}}, \mathsf{st}_{\mathcal{A},4}; \mathsf{tr}_{\mathsf{Com}}) \leftarrow \langle \mathcal{A}(\mathsf{st}_{\mathcal{A},3}, \mathsf{in}_{\mathsf{Com}}, \{\mathsf{paux}_i, \mathsf{pwit}_i\}_{i\in\mathcal{V}}) \leftrightharpoons \{\mathsf{P}_i(\mathsf{in}_{\mathsf{Com}}, \mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{S}\setminus\mathcal{V}} \rangle_{\Pi_{\mathsf{tCP}}.\mathsf{Commit}}$

$(\pi; \mathsf{tr}_{\mathsf{Prove}}) \leftarrow \langle \mathcal{A}(\mathsf{st}_{\mathcal{A},4}, c, \{\mathsf{pr}_i\}_{i\in\mathcal{V}}) \leftrightharpoons \{\mathsf{P}_i(\mathsf{in}_{\mathsf{Com}}, \mathsf{paux}_i, \mathsf{pwit}_i, c, \mathsf{pr}_i)\}_{i\in\mathcal{S}\setminus\mathcal{V}} \rangle_{\Pi_{\mathsf{tCP}}.\mathsf{Prove}}$

else if $b = 1$:

$\mathsf{in}_{\mathsf{Sim}} := (\mathsf{pp}, \mathsf{td}, \mathbb{x}, \mathcal{S}, \mathcal{V}); (\{\mathsf{pwit}_i\}_{i\in\mathcal{V}}, \mathsf{aux}_w) \leftarrow \mathsf{Sim.SimWitShare}(\mathsf{in}_{\mathsf{Sim}})$

$\mathsf{in}_{\mathcal{A},\mathsf{Com}} := (\{\mathsf{paux}_i, \mathsf{pwit}_i\}_{i\in\mathcal{V}}, \mathsf{aux}_w)$

$(c, \{\mathsf{pr}_i\}_{i\in\mathcal{V}}, \mathsf{st}_{\mathcal{A},4}; \mathsf{tr}_{\mathsf{Com}}) \leftarrow \mathsf{Sim.SimCom}^{\mathcal{A}(\mathsf{st}_{\mathcal{A},3}, \mathsf{in}_{\mathcal{A},\mathsf{Com}})}(\mathsf{in}_{\mathsf{Sim}}, \mathsf{in}_{\mathcal{A},\mathsf{Com}})$

$\mathsf{in}_{\mathcal{A},\mathsf{Prove}} := (c, \{\mathsf{pr}_i\}_{i\in\mathcal{V}})$

$(\pi; \mathsf{tr}_{\mathsf{Prove}}) \leftarrow \mathsf{Sim.SimProve}^{\mathcal{A}(\mathsf{st}_{\mathcal{A},4}, \mathsf{in}_{\mathcal{A},\mathsf{Prove}})}(\mathsf{in}_{\mathsf{Sim}}, \mathsf{in}_{\mathcal{A},\mathsf{Prove}})$

return $\mathsf{D}(\mathsf{tr}_{\mathsf{Com}}, \mathsf{tr}_{\mathsf{Prove}}, \{\mathsf{paux}_i, \mathsf{pwit}_i\}_{i\in\mathcal{V}}, \mathsf{aux}_w, c, \pi)$

Figure 1: Threshold zero-knowledge experiment

in the style of [OB22]: the provers first jointly check, via a secure function evaluation protocol, whether the statement-witness pair is in the relation, and abort otherwise.

Definition 4.4 (Non-adaptive Threshold Simulation Extractability)**.** *A threshold commit-and-prove scheme* Π_{tCP} *for an NP relation* $\mathcal{R}$ *with its associated simulator* Sim *satisfies non-adaptive threshold simulation extractability if there exists an extractor* Ext *running in expected poly-time such that for any* ppt *adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*, any poly-sized subset* $\mathcal{G} \subset \mathcal{L}$*, where* $\mathcal{L}$ *is the language associated with* $\mathcal{R}$*, and with* Ext *given black-box rewindable access to* $\mathcal{A}$*, we have*

$$\Pr[\mathsf{Thrsh\text{-}SimExt\text{-}Exp}^{\mathcal{A},\mathsf{Ext},\mathcal{G},\Pi_{\mathsf{tCP}}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda),$$

where Thrsh-SimExt-Exp *is defined in Fig. 2. In addition,* Π_{tCP} *is said to satisfy semi-honest threshold knowledge soundness if* $\mathcal{A}$ *acts honestly in* $\mathcal{O}_{\mathsf{Prove}}$ *and* $\mathcal{O}_{\mathsf{Com}}$ *and acts before the honest parties in those oracles (*P_i*).*

Remark 4.3 (Non-adaptive Threshold Simulation Extractability Intuition). Intuitively, if a ppt adversary $\mathcal{A}$ receives at most $\mathtt{t} - 1$ shares of a witness $\mathbb{w}$ for a statement $\mathbb{x}$ with $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$ and outputs a valid proof π without interacting with an honest prover, then we should be able to extract $\mathbb{w}'$ from $\mathcal{A}$ such that $(\mathbb{x}, \mathbb{w}') \in \mathcal{R}$. Our definition of threshold simulation extractability thus captures the inability of an adversary controlling at most $\mathtt{t} - 1$ provers to generate a valid commitment-proof pair without the participation of an honest prover, *even after obtaining proofs on that statement by interacting with honest provers*. Following the literature on signatures of knowledge [CL06, GM17], we model this by granting $\mathcal{A}$ oracle access to the simulator. We do not allow queries on any statements, since given a full witness, $\mathcal{A}$ could simulate honest provers' inputs and complete the protocol on its own. The experiment in Fig. 2 therefore restricts queries to a poly-sized subset $\mathcal{G}$

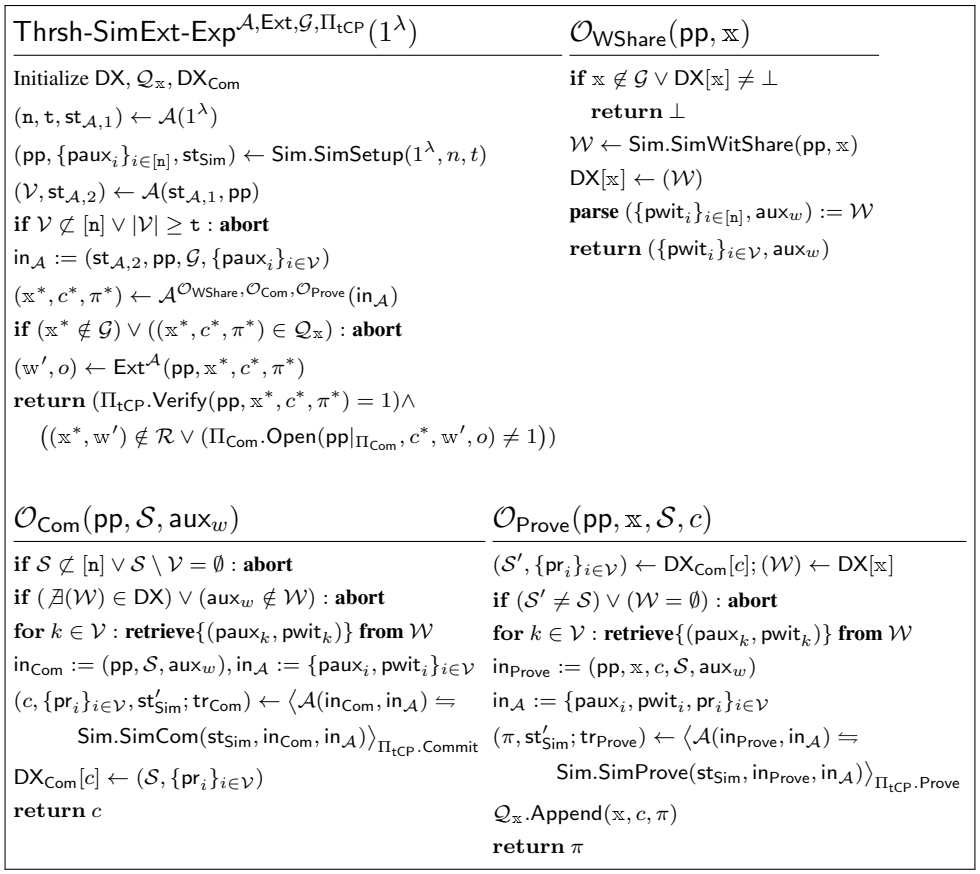

Figure 2: Threshold simulation extractability experiment.

of the language $\mathcal{L}$. Note that when a witness is easy to obtain from $\mathbb{x} \in \mathcal{L}$ alone, as is the case for languages decidable in poly-time, extraction of a witness is the best we can guarantee.

5 Thresholdizing ALS

In this section, we propose a thresholdization of the product proof of Attema, Lyubashevsky, and Seiler [ALS20, Section 5.1]. To simplify our analysis, we modify the product proof to drop the ring automorphism optimization and call the resulting variant "*vanilla*" ALS. In Section 5.1, we show that its Fiat–Shamir transform is simulation extractable, and in Section 5.2 we give its thresholdization.

We recall that ALS allows to issue commit-and-prove-type proofs for the relation $\mathcal{R} := \{(\mathbb{x} := \bot, \mathbb{w} := (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)) : \mathbb{w} \in R_q^3 \wedge \mathbf{m}_1\mathbf{m}_2 = \mathbf{m}_3\}$[3]. Since the statement is empty, we omit it throughout this section.

We assume that X^d+1 factors into h irreducible factors modulo q, i.e., $X^d+1 \equiv \prod_{i=1}^{h}(X^{d/h}-\zeta_i) \bmod q$, where each $\zeta_i \in \mathbb{Z}_q^\times$ is a $2h$-th primitive root of unity.

[3]This can be adapted to vectors of any length, and to require the product of components to equal a public value, as in Section 6.

5.1 Simulation Extractable ALS

Construction 1: vALS

For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be a prime. Let $n = n(\lambda)$, $m = m(\lambda)$ be matrix dimensions. Let $\mathfrak{s}_1 = \mathfrak{s}_1(\lambda)$ and $\mathfrak{s}_2 = \mathfrak{s}_2(\lambda)$ be standard deviations, and $\beta = \beta(\lambda)$ be a norm bound. Let $\mathcal{C}$ be the challenge distribution described in Section 2.4, and $H : R_q^{n+4} \times R_q^n \times R_q \to \mathcal{C}$ be a random oracle. We construct a rejection-free variant of *vanilla* ALS as follows:

- $\mathsf{Setup}(1^\lambda)$: Given $\lambda \in \mathbb{N}$, return $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1) \leftarrow\$ R_q^{n\times m} \times R_q^{4\times m}$.
- $\mathsf{Prove}(\mathsf{pp}, \mathbb{w})$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3) := \mathbb{w}$, proceed as follows:
  - **Commit Phase:**
    1. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_4]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
    2. Sample $\vec{\mathbf{r}} \leftarrow D^m_{R,\mathfrak{s}_1}$, and set $\vec{\mathbf{m}} \leftarrow [\mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3]^\top \in R_q^3$.
    3. Compute $[\vec{\mathbf{t}_0}^\top \mid \mathbf{t}_1 \mid \mathbf{t}_2 \mid \mathbf{t}_3]^\top \leftarrow \begin{bmatrix} \mathbf{B}_0 \\ [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_2 \mid \vec{\mathbf{b}}_3]^\top \end{bmatrix} \vec{\mathbf{r}} + \begin{bmatrix} \mathbf{0}^n \\ \vec{\mathbf{m}} \end{bmatrix}$
    4. Return $\vec{\mathbf{t}} := [\vec{\mathbf{t}_0}^\top \mid \mathbf{t}_1 \mid \mathbf{t}_2 \mid \mathbf{t}_3]^\top$ and $o := (\vec{\mathbf{r}}, 1)$.
  - **Prove Phase:**
    1. Sample $\vec{\mathbf{y}} \leftarrow D^m_{R,\mathfrak{s}_2}$, compute $\vec{\mathbf{w}} \leftarrow \mathbf{B}_0\vec{\mathbf{y}}$, parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_4]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
    2. Compute $\mathbf{t}_4 \leftarrow \langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}} \rangle + \langle \vec{\mathbf{b}}_3, \vec{\mathbf{y}} \rangle - \mathbf{m}_2 \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle - \mathbf{m}_1 \langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}} \rangle$.
    3. Compute $\mathbf{v} \leftarrow \langle \vec{\mathbf{b}}_4, \vec{\mathbf{y}} \rangle + \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle \cdot \langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}} \rangle$.
    4. ▶ Compute $\mathbf{c} \leftarrow H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v})$.
    5. Compute $\vec{\mathbf{z}} \leftarrow \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}}$.
    6. Return $(\vec{\mathbf{t}}, \pi := (\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}))$.
- $\mathsf{Verify}(\mathsf{pp}, (\vec{\mathbf{t}}, \pi))$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, commitment $\vec{\mathbf{t}} \in R_q^{n+3}$ and proof $(\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}) := \pi$, parse $\left[\vec{\mathbf{t}_0}^\top \mid \mathbf{t}_1 \mid \cdots \mid \mathbf{t}_3\right]^\top := \vec{\mathbf{t}}$ and proceed as follows:
  1. Abort and output 0 if $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) \neq \mathbf{c}$ or $\mathbf{B}_0\vec{\mathbf{z}} \neq \vec{\mathbf{w}} + \mathbf{c}\vec{\mathbf{t}}_0$.
  2. Abort and output 0 if $\|\vec{\mathbf{z}}\| > \beta$.
  3. Compute $[\mathbf{f}_1 \mid \cdots \mid \mathbf{f}_4]^\top \leftarrow \mathbf{B}_1\vec{\mathbf{z}} - \mathbf{c}[\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$.
  4. Abort and output 0 if $\mathbf{f}_1\mathbf{f}_2 + \mathbf{c} \cdot \mathbf{f}_3 + \mathbf{f}_4 \neq \mathbf{v}$. Else, output 1.

Theorem 5.1 (vALS Completeness)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let $\mathcal{C}$ be the distribution defined in Section 2.4. Then Construction 1 is complete except with negligible probability.*

Proof. Let $(\vec{\mathbf{t}}, \pi := (\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}))$ be an honest proof generated by executing the Prove algorithm of Construction 1. There exist $\vec{\mathbf{y}} \leftarrow D^m_{R,\mathfrak{s}_2}$ and $\vec{\mathbf{r}} \leftarrow D^m_{R,\mathfrak{s}_1}$ such that $\vec{\mathbf{w}} = \mathbf{B}_0\vec{\mathbf{y}}$ and $\vec{\mathbf{z}} = \vec{\mathbf{y}} + \mathbf{c}\vec{\mathbf{r}}$. We have $\|\vec{\mathbf{z}}\|_2 \leq \|\vec{\mathbf{y}}\|_2 + \|\mathbf{c}\|_1\|\vec{\mathbf{r}}\|_2 \leq \|\vec{\mathbf{y}}\|_2 + d\|\vec{\mathbf{r}}\|_2$. Since $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, Lemma 2.5 implies that $\Pr[\|\vec{\mathbf{z}}\|_2 > \beta] \leq 2^{-md+1}$, which is negligible. The rest of the proof follows by inspection. □

Theorem 5.2 (vALS ZK)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, and $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$. Let $\mathcal{C}$ be the distribution defined in Section 2.4. Then Construction 1 is zero-knowledge against any* ppt *distinguisher making at most Q_H random-oracle and Q_P proof queries, under the hardness of the Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-n-4,n+4,q,\mathfrak{s}_1}$ assumption.*

Proof Sketch. The proof adapts the interactive-setting proof given by Kim et al. [KLSS23, Appendix B] to the

non-interactive setting. Here, we describe the simulator and gives a high level overview of the steps required to show indistinguishability between real and simulated proofs. The full proof is given in Appendix D.

Simulator 1: $\mathsf{Sim}_{\mathsf{vALS}}$

The simulator maintains the random oracle.

- $\mathsf{SimSetup}(1^\lambda)$: Given $\lambda \in \mathbb{N}$, proceed as follows:
  1. Sample $\mathbf{V} \leftarrow\!\$\ \mathrm{GL}_{n+4}(R_q)$ and $\mathbf{A} \leftarrow\!\$\ R_q^{(n+4)\times(m-n-4)}$.
  2. Compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{n+4} \mid \mathbf{A}]$, where $(\mathbf{B}_0, \mathbf{B}_1) \in R_q^{n\times m} \times R_q^{4\times m}$.
  3. Return $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$, and keep $\mathbf{V}$ in memory.
- $\mathsf{SimProve}(\mathsf{pp})$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, proceed as follows:
  1. Sample $\mathbf{c} \leftarrow \mathcal{C}$, $\vec{\mathbf{r}} \leftarrow D^m_{R,\mathfrak{s}_1}$, and $\vec{\mathbf{u}} \leftarrow\!\$\ R_q^{n+4}$. Then, sample $\vec{\mathbf{z}} \leftarrow D^m_{R,\mathfrak{s}_2} + \mathbf{c}\vec{\mathbf{r}}$.
  2. Compute $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top \leftarrow \mathbf{V}\vec{\mathbf{u}}$, and then parse $\left[\vec{\mathbf{t}_0}^\top \mid \mathbf{t}_1 \mid \cdots \mid \mathbf{t}_3\right]^\top := \vec{\mathbf{t}}$.
  3. Compute $\vec{\mathbf{w}} \leftarrow \mathbf{B}_0\vec{\mathbf{z}} - \mathbf{c}\vec{\mathbf{t}_0}$ and $[\mathbf{f}_1 \mid \cdots \mid \mathbf{f}_4]^\top = \mathbf{B}_1\vec{\mathbf{z}} - \mathbf{c}[\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$.
  4. Compute $\mathbf{v} \leftarrow \mathbf{f}_1\mathbf{f}_2 + \mathbf{c}\cdot\mathbf{f}_3 + \mathbf{f}_4$.
  5. ▶ If $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) \neq \perp$, abort. Else, set $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) \leftarrow \mathbf{c}$.
  6. Return $(\vec{\mathbf{t}}, \pi := (\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}))$.
- $\mathsf{SimRO}(\mathsf{qry})$: Given a query $\mathsf{qry} \in R_q^{n+4} \times R_q^n \times R_q$, if $H(\mathsf{qry}) = \perp$, sample $\mathbf{c} \leftarrow \mathcal{C}$, and then, set $H(\mathsf{qry}) \leftarrow \mathbf{c}$. Return $H(\mathsf{qry})$.

First, we observe that $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top$ has high min-entropy (see Lemma 2.3). Therefore, except with negligible probability, we can sample the challenge $\mathbf{c}$ in advance and then program $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v})$. Now that the challenge is independent of the transcript, we can sample the final element $\vec{\mathbf{z}}$ in advance and reverse-engineer $\vec{\mathbf{w}}$ and $\mathbf{v}$ given $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top$. Finally, we use the Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-n-4,n+4,q,\mathfrak{s}_1}$ to argue that $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top$ is random and $\vec{\mathbf{z}}$ does not leak information about the commitment randomness across the adversary's proofs. □

Theorem 5.3 (vALS WUR)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, and $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$. Let $\mathcal{C}$ be the distribution defined in Section 2.4. Let $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$ be the norm bound for the verification. Then Construction 1 has weak unique response under the hardness of $\mathsf{MSIS}_{n,m,q,2\beta}$.*

Proof. Given a simulated proof $(\vec{\mathbf{t}}', (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}', \vec{\mathbf{z}}') := \pi')$, the only way for an adversary to break weak-unique response is to output a different response $\vec{\mathbf{z}}$ that verifies with $(\vec{\mathbf{t}}', (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}'))$. By the verification equation, this implies that $\mathbf{B}_0(\vec{\mathbf{z}}' - \vec{\mathbf{z}}) = \mathbf{0}^n$ and $\vec{\mathbf{z}}' - \vec{\mathbf{z}} \neq \mathbf{0}^m$. This completes the proof. □

Theorem 5.4 (vALS Knowledge Soundness)**.** *For any $\lambda \in \mathbb{N}$, $q = q(\lambda)$ be a prime, $n = n(\lambda)$, $m = m(\lambda)$, and $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_i)$ for any $i \in [h]$ (see Lemma 2.6). Under the hardness of $\mathsf{MSIS}_{n,m,q,8d\beta}$, Construction 1 is adaptively knowledge-sound with knowledge error at most $4(Q+1)(h+1)M$, where Q is the number of random oracle queries made by P. In addition, the extractor makes $2Q+3$ queries to P in expectation.*

Theorem 5.4 is proved in Appendix D.2 by showing that vALS satisfies predicate special soundness via a Schwartz–Zippel argument: if $\mathbf{m}_1\mathbf{m}_2 \neq \mathbf{m}_3$, then Line 4 of verification is a degree-two polynomial.

Corollary 5.4.1 (vALS Simulation-Extractable)**.** *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_i)$ for any $i \in [h]$ (see Lemma 2.6). Under the hardness of $\mathsf{MSIS}_{n,m,q,8d\beta}$, Construction 1 is weakly simulation-extractable.*

Proof. This follows from Theorems 5.3 and 5.4 and Lemma 3.1. □

5.2 Threshold ALS

We omit the Verify algorithm since it is the same as in Construction 1.

Construction 2: Threshold vALS

Let $\Pi_{\mathsf{tHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{TDec}, \mathsf{Comb})$ be a threshold homomorphic encryption scheme supporting the evaluation of polynomials of total degree at most two and $\Pi_{\mathsf{SS}} = (\mathsf{Share}, \mathsf{Reconstruct})$ be Shamir secret sharing scheme. For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be a prime. Let $n = n(\lambda)$, $m = m(\lambda)$ be matrix dimensions. Let $\mathfrak{s}_1 = \mathfrak{s}_1(\lambda)$ and $\mathfrak{s}_2 = \mathfrak{s}_2(\lambda)$ be standard deviations, and $\beta = \beta(\lambda)$ be a norm bound. Let $\mathcal{C}$ be the challenge distribution described in Section 2.4, and $H : R_q^{n+4} \times R_q^n \times R_q \to \mathcal{C}$ be a random oracle. We construct a threshold variant of vALS (Construction 1) as follows:

- $\mathsf{Setup}(1^\lambda, \mathtt{n}, \mathtt{t})$: Given the security parameter $\lambda \in \mathbb{N}$ and the threshold parameters $\mathtt{n}, \mathtt{t} \in \mathbb{N}$, do the following:
  1. Compute $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}) \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t})$.
  2. Sample $\mathbf{B}_0 \leftarrow\!\$\ R_q^{n\times m}$ and $\mathbf{B}_1 \leftarrow\!\$\ R_q^{4\times m}$.
  3. Return $\mathsf{pp} = (\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1)$ and $\mathsf{paux}_i = \mathsf{sk}_i$ to each prover P_i.
- $\mathsf{WitnessShare}(\mathsf{pp}, \mathbb{w})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3) := \mathbb{w}$, do the following:
  1. For $i \in [3]$, compute $\mathsf{ctx}_{\mathbf{m}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{m}_i)$ and $\{[\![\mathbf{m}_i]\!]_j\}_{j\in[\mathtt{n}]} \leftarrow \Pi_{\mathsf{SS}}.\mathsf{Share}(\mathbf{m}_i, (\mathtt{n}, \mathtt{t}))$.
  2. Set $\mathsf{aux}_w = (\mathsf{ctx}_{\mathbf{m}_1}, \mathsf{ctx}_{\mathbf{m}_2}, \mathsf{ctx}_{\mathbf{m}_3})$.
  3. For $i \in [\mathtt{n}]$, set $\mathsf{pwit}_i = ([\![\mathbf{m}_1]\!]_i, [\![\mathbf{m}_2]\!]_i, [\![\mathbf{m}_3]\!]_i)$ and send $(\mathsf{pwit}_i, \mathsf{aux}_w)$ to prover P_i.
- $\mathsf{Commit}(\mathsf{pp}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w)$: This is an interactive protocol executed by provers in $\mathcal{S}$. Each prover $\mathsf{P}_i \in \mathcal{S}$ has inputs $(\mathsf{pp}, \mathcal{S}, \mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{aux}_w)$ and proceeds as follows:
  1. Parse $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $\mathsf{sk}_i := \mathsf{paux}_i$, $([\![\mathbf{m}_1]\!]_i, [\![\mathbf{m}_2]\!]_i, [\![\mathbf{m}_3]\!]_i) := \mathsf{pwit}_i$ and $(\mathsf{ctx}_{\mathbf{m}_1}, \mathsf{ctx}_{\mathbf{m}_2}, \mathsf{ctx}_{\mathbf{m}_3}) := \mathsf{aux}_w$.
  2. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_4]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_j \in R_q^m$.
  3. Sample $\vec{\mathbf{r}}_i \leftarrow D^m_{R,\mathfrak{s}_1}$ and compute $\vec{\mathbf{t}}_{0,i} \leftarrow \mathbf{B}_0\vec{\mathbf{r}}_i$.
  4. Use the set $\mathcal{S}$ to derive the ith Lagrange coefficient λ_i, and then, compute $[\mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]^\top \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_2 \mid \vec{\mathbf{b}}_3]^\top \cdot \vec{\mathbf{r}}_i + \lambda_i \cdot [[\![\mathbf{m}_1]\!]_i \mid [\![\mathbf{m}_2]\!]_i \mid [\![\mathbf{m}_3]\!]_i]^\top$
  5. ▶ Broadcast $(\vec{\mathbf{t}}_{0,i}, \mathbf{t}_{1,i}, \mathbf{t}_{2,i}, \mathbf{t}_{3,i})$ to all other participants in $\mathcal{S}$.
  6. Let $(\vec{\mathbf{t}}_{0,j}, \mathbf{t}_{1,j}, \mathbf{t}_{2,j}, \mathbf{t}_{3,j})$ be the values from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$
  7. For $(\ell + 1) \in [4]$, compute $\mathbf{t}_\ell \leftarrow \sum_{k\in\mathcal{S}} \mathbf{t}_{\ell,k}$.
  8. Set $\mathsf{pr}_i := \vec{\mathbf{r}}_i$ and $\vec{\mathbf{t}} := [\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1 \mid \cdots \mid \mathbf{t}_3]^\top$.
  9. Store pr_i and output $\vec{\mathbf{t}}$.
- $\mathsf{Prove}(\mathsf{pp}, \vec{\mathbf{t}}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w)$: This is an interactive protocol executed by provers in $\mathcal{S}$. Each prover $\mathsf{P}_i \in \mathcal{S}$ has inputs $(\mathsf{pp}, \mathcal{S}, \vec{\mathbf{t}}, \mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i, \mathsf{aux}_w)$ and proceeds as follows:
  1. Parse $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $[\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1 \mid \cdots \mid \mathbf{t}_3]^\top := \vec{\mathbf{t}}$, $\mathsf{sk}_i := \mathsf{paux}_i$, $([\![\mathbf{m}_1]\!]_i, [\![\mathbf{m}_2]\!]_i, [\![\mathbf{m}_3]\!]_i) := \mathsf{pwit}_i$, $\vec{\mathbf{r}}_i := \mathsf{pr}_i$, and $(\mathsf{ctx}_{\mathbf{m}_1}, \mathsf{ctx}_{\mathbf{m}_2}, \mathsf{ctx}_{\mathbf{m}_3}) := \mathsf{aux}_w$.
  2. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_4]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_j \in R_q^m$.
  3. Sample $\vec{\mathbf{y}}_i \leftarrow D^m_{R,\mathfrak{s}_2}$, and then compute $\vec{\mathbf{w}}_i \leftarrow \mathbf{B}_0\vec{\mathbf{y}}_i$.
  4. Compute $\mathsf{ctx}_{\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{\mathbf{y}}_i)$ and $\mathsf{ctx}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{\mathbf{r}}_i)$.
  5. Compute $\vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{B}_1\vec{\mathbf{y}}_i)$; $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_4, \vec{\mathbf{r}}_i\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \langle\vec{\mathbf{b}}_4, \vec{\mathbf{r}}_i\rangle)$.
  6. ▶ Broadcast $(\vec{\mathbf{w}}_i, \mathsf{ctx}_{\vec{\mathbf{y}}_i}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i}, \mathsf{ctx}_{\vec{\mathbf{r}}_i}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4, \vec{\mathbf{r}}_i\rangle})$ to other participants in $\mathcal{S}$.
  7. Upon receiving $(\vec{\mathbf{w}}_j, \mathsf{ctx}_{\vec{\mathbf{y}}_j}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_j}, \mathsf{ctx}_{\vec{\mathbf{r}}_j}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4, \vec{\mathbf{r}}_j\rangle})$ from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$, proceed as follows:
     (a) Compute $\vec{\mathbf{w}} \leftarrow \sum_{k\in\mathcal{S}} \vec{\mathbf{w}}_k$.
     (b) Compute $\mathsf{ctx}_{\vec{\mathbf{y}}} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\vec{\mathbf{y}}_k}$ and $\mathsf{ctx}_{\vec{\mathbf{r}}} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\vec{\mathbf{r}}_k}$.

(c) Compute $\vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}} \leftarrow \sum_{k\in\mathcal{S}} \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_k}$; $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}_k\rangle}$.

8. Parse $\left[\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}\rangle} \mid \cdots \mid \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{y}}\rangle}\right]^\top := \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}}$.
9. Compute $\mathsf{ctx}_{\mathbf{t}_4} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}\rangle} + \mathsf{ctx}_{\langle\vec{\mathbf{b}}_3,\vec{\mathbf{y}}\rangle} - \mathsf{ctx}_{\mathbf{m}_1}\cdot\mathsf{ctx}_{\langle\vec{\mathbf{b}}_2,\vec{\mathbf{y}}\rangle} - \mathsf{ctx}_{\mathbf{m}_2}\cdot\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}\rangle}$.
10. Compute $\mathsf{ds}_{\mathbf{t}_{4,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}\left(\mathsf{ctx}_{\mathbf{t}_4}, \mathsf{sk}_i, \mathcal{S}\right)$.
11. Compute $\mathsf{ctx}_{\mathbf{v}} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{y}}\rangle} + \mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}\rangle}\cdot\mathsf{ctx}_{\langle\vec{\mathbf{b}}_2,\vec{\mathbf{y}}\rangle}$.
12. Compute $\mathsf{ds}_{\mathbf{v}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}\left(\mathsf{ctx}_{\mathbf{v}}, \mathsf{sk}_i, \mathcal{S}\right)$.
13. ▶ Broadcast $(\mathsf{ds}_{\mathbf{t}_{4,i}}, \mathsf{ds}_{\mathbf{v}_i})$ to all other participants in $\mathcal{S}$.
14. Upon receiving $(\mathsf{ds}_{\mathbf{t}_{4,j}}, \mathsf{ds}_{\mathbf{v}_j})$ from all participants $P_j \in \mathcal{S}\setminus\{P_i\}$ proceed as follows:
    (a) Compute $\mathbf{t}_4 \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \{\mathsf{ds}_{\mathbf{t}_{4,k}}\}_{k\in\mathcal{S}}, \mathcal{S})$.
    (b) Compute $\mathbf{v} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \{\mathsf{ds}_{\mathbf{v}_k}\}_{k\in\mathcal{S}}, \mathcal{S})$
15. ▶ Compute $\mathbf{c} \leftarrow H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v})$.
16. Compute $\mathsf{ctx}_{\vec{\mathbf{z}}} \leftarrow \mathsf{ctx}_{\vec{\mathbf{y}}} + \mathbf{c}\cdot\mathsf{ctx}_{\vec{\mathbf{r}}}$.
17. Compute $\mathsf{ds}_{\vec{\mathbf{z}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}\left(\mathsf{ctx}_{\vec{\mathbf{z}}}, \mathsf{sk}_i, \mathcal{S}\right)$.
18. ▶ Broadcast $\mathsf{ds}_{\vec{\mathbf{z}}_i}$ to all other participants in $\mathcal{S}$.
19. Upon receiving $\mathsf{ds}_{\vec{\mathbf{z}}_j}$ from all participants $P_j \in \mathcal{S}\setminus\{P_i\}$, compute $\vec{\mathbf{z}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \{\mathsf{ds}_{\vec{\mathbf{z}}_k}\}_{k\in\mathcal{S}}, \mathcal{S})$.
20. Output $(\vec{\mathbf{t}}, \pi := (\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}))$

Theorem 5.5. *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathtt{n} = \mathtt{n}(\lambda)$, $\mathtt{t} = \mathtt{t}(\lambda)$ such that $\mathtt{n} \geq \mathtt{t}$. Let $\mathfrak{s}_1, \mathfrak{s}_2 \geq \sqrt{2}\cdot\eta_\varepsilon(R^m)$. Suppose Π_{tHE} is semantically and simulation secure, and vanilla ALS (Construction 1) is zero-knowledge. In addition, suppose $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$, $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$ and $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ hold, where $\bar{n} = n+4$. Then, Construction 2 satisfies weak $(\mathtt{t}-1)$-ZK (Definition 4.3) against a semi-honest adversary that statically corrupts at most $\mathtt{t}-1$ provers and makes at most Q_H random oracle queries and Q_P proof queries.* [4]

Theorem 5.5 is proved in Appendix D.3. At a high level, we prove that if a semi-honest adversary can break the weak ($\mathtt{t}$-1) ZK property of Construction 2, then there exists an adversary that can break the ZK property of Construction 1. This is achieved by building a simulator that relies on the simulator of Construction 1 to simulate the contribution of honest provers against Construction 2's adversary.
For the rest of this section, we assume, without loss of generality, that $|\mathcal{S}| = \mathtt{t}$.

Theorem 5.6. *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathtt{n} = \mathtt{n}(\lambda)$, $\mathtt{t} = \mathtt{t}(\lambda)$ such that $\mathtt{n} \geq \mathtt{t}$. Let $\mathfrak{s}_1, \mathfrak{s}_2 \geq \sqrt{2}\cdot\eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{\mathtt{t}md}$. Assume Π_{tHE} has perfect evaluation correctness. Then Construction 2 is complete except with negligible probability.*

Proof. Let $\mathcal{S} \subseteq [\mathtt{n}]$, with $|\mathcal{S}| = \mathtt{t}$, be any set of provers, and let $(\vec{\mathbf{t}}, \pi := (\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \mathbf{c}, \vec{\mathbf{z}}))$ be a proof honestly generated by those provers. From line 20 of the Prove algorithm in Construction 2, and by the correctness of Π_{tHE} it follows that $\vec{\mathbf{z}} = \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i + \mathbf{c}\vec{\mathbf{r}}_i$, where $\vec{\mathbf{y}}_i \leftarrow D^m_{R,\mathfrak{s}_2}$, $\vec{\mathbf{r}}_i \leftarrow D^m_{R,\mathfrak{s}_1}$. By Lemma 2.4, the sums $\sum_{i\in\mathsf{S}} \vec{\mathbf{y}}_i$ and $\sum_{i\in\mathsf{S}} \vec{\mathbf{r}}_i$ are within negligible statistical distance of $\vec{\mathbf{y}}' \leftarrow D^m_{R,\mathfrak{s}_2\sqrt{\mathtt{t}}}$ and $\vec{\mathbf{r}}' \leftarrow D^m_{R,\mathfrak{s}_1\sqrt{\mathtt{t}}}$, respectively. Therefore, $\vec{\mathbf{z}}$ is statistically close to $\vec{\mathbf{z}}' = \vec{\mathbf{y}}' + \mathbf{c}\vec{\mathbf{r}}'$, and by Lemma 2.5, $\Pr[\|\vec{\mathbf{z}}\| > \beta] \leq 2^{-md+1} + \delta_{\vec{\mathbf{z}},\vec{\mathbf{z}}'} = \mathsf{negl}(\lambda)$, where $\delta_{\vec{\mathbf{z}},\vec{\mathbf{z}}'}$ denotes the statistical distance between $\vec{\mathbf{z}}$ and $\vec{\mathbf{z}}'$. Finally, since Π_{tHE} has perfect

[4] In Remark 4.2, we noted that when the statement-witness pair is not in the relation the output of protocol might revealed information about honest provers' inputs. For example, if $\mathbf{m}_1\mathbf{m}_2 \neq \mathbf{m}_3$, then $(\mathbf{f}_1\mathbf{f}_2 + \mathbf{c}\mathbf{f}_3 + \mathbf{f}_4 - \mathbf{v})/\mathbf{c}^2 = \mathbf{m}_1\mathbf{m}_2 - \mathbf{m}_3$ which we are unable to simulate. As previously mentioned, this can be mitigated by using a secure evaluation function protocol (e.g., [NO07]) to check that $\mathbb{w}$ is indeed in the relation before executing the protocol. However, this is not an issue for satisfying weak ($\mathtt{t}$-1)-ZK since we only consider witnesses that belong to the relation.

evaluation correctness, it follows that $\mathbf{t}_4$ and $\mathbf{v}$ are correctly computed. The rest follows by inspection and from the completeness of Construction 1. □

Theorem 5.7. *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathtt{n} = \mathtt{n}(\lambda) \geq \mathtt{t} = \mathtt{t}(\lambda)$. Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $d = d(\lambda)$ be lattice parameters. Let $\mathfrak{s}_1, \mathfrak{s}_2 \geq \sqrt{2} \cdot \eta_\varepsilon(R^m)$ be standard deviations. Set $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$ and $\beta' = \beta\sqrt{\mathtt{t}}$. Assume that (i) the $\mathsf{MSIS}_{n,m,q,8d\beta'}$ assumption holds, (ii) vanilla ALS (Construction 1) is weakly simulation-extractable, and (iii) Construction 2 is weakly $(\mathtt{t}-1)$-zero-knowledge (Theorem 5.5). Then Construction 2 satisfies threshold simulation extractability (Definition 4.4).*

We prove Theorem 5.7 in Appendix D.4.

6 Thresholdizing ENS

In this section, we present a rejection-free version of ENS [ENS20], prove it to be simulation-extractable, and then thresholdize it. ENS allows one to issue commit-and-prove-type proofs for the relation $\mathcal{R} := \{(\mathbb{x} := (A, \vec{u}) \in \mathbb{Z}_q^{\nu \times d} \times \mathbb{Z}_q^{\nu}, \mathbb{w} := \mathbf{s} \in R_q) : A \cdot \mathsf{NTT}(\mathbf{s}) = \vec{u} \wedge \mathsf{NTT}(\mathbf{s}) \in \{0, \pm 1\}^d\}$[5] For ease of exposition, we omit the optimizations based on ring automorphisms that were used by Esgin, Nguyen and Seiler to increase soundness while incurring only a minimal proof-size overhead. Although our variant yields larger proofs, it is easier to thresholdize. Nevertheless, in Section 7, we present tools that allow us to recover the ring-automorphism optimizations of Esgin, Nguyen, and Seiler.

Moreover, beyond dropping the ring-automorphism-based optimization, we slightly modify the ENS proof to make it more threshold-friendly. Let $\mathbb{x} := (A, \vec{u})$ be a valid statement. Whereas a standard ENS proof for $\mathbb{x}$ outputs an element $\mathbf{h} \in R_q$ satisfying $\mathsf{ct}(\mathbf{h}) = 0$, we instead compute $\mathbf{h}$ so that $\mathsf{ct}(\mathbf{h}) = \langle \vec{u}, \vec{v} \rangle$, where $\vec{v}$ is a challenge obtained from the random oracle. Computing $\mathbf{h}$ in this way lets us thresholdize this part of the protocol easily, using the linearity of Shamir secret sharing. Crucially, the modification preserves both the zero-knowledge and knowledge-soundness properties of the resulting protocol, since $\langle \vec{u}, \vec{v} \rangle$ is a public value derivable from the challenge $\vec{v}$ and the statement $\mathbb{x}$. We assume that $X^d + 1$ fully splits into irreducible factors modulo q, i.e., $X^d + 1 \equiv \prod_{i=1}^{d} (X^{d/h} - \zeta_i) \bmod q$, where each $\zeta_i \in \mathbb{Z}_q^\times$ is a $2d$-th primitive root of unity.

6.1 Simulation-Extractable ENS

Construction 3: vENS

For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be a prime. Let $m = m(\lambda)$ and $n = n(\lambda)$ be matrix dimensions. Let $\mathfrak{s}_1 = \mathfrak{s}_1(\lambda)$ and $\mathfrak{s}_2 = \mathfrak{s}_2(\lambda)$ be standard deviations, and $\beta = \beta(\lambda)$ be a norm bound. Let $\ell = \ell(\lambda)$ be a repetition parameter and $\mathcal{C}$ be the challenge distribution described in Section 2.4. Let $H_1 : R_q^{n+3\ell+1} \times R_q^{\ell} \times R_q^{n\ell} \to (\mathbb{Z}_q^{\nu})^{\ell}$ and $H_2 : R_q^{n+3\ell+1} \times R_q^{\ell} \times R_q^{n\ell} \times R_q^{2\ell} \to \mathcal{C}^{\ell}$ be random oracles.

- $\mathsf{Setup}(1^\lambda)$: Given $\lambda \in \mathbb{N}$, return $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1) \leftarrow\$ R_q^{n \times m} \times R_q^{(3\ell+1) \times m}$.
- $\mathsf{Prove}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(A, \vec{u}) := \mathbb{x}$, and $\mathbf{s} := \mathbb{w}$ such that $\mathsf{NTT}(\mathbf{s}) \in \{0, \pm 1\}^d$, proceed as follows:
  - **Commit Phase:**
    1. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
    2. Sample $\vec{\mathbf{r}} \leftarrow D_{R, \mathfrak{s}_1}^m$ and compute $\vec{\mathbf{t}} \leftarrow [\mathbf{B}_0^\top \mid \vec{\mathbf{b}}_1]^\top \vec{\mathbf{r}} + [(\mathbf{0}^n)^\top \mid \mathbf{s}]^\top$.
    3. Set $o := (\vec{\mathbf{r}}, 1)$.

[5] $\mathbb{w}$ can be extended to any length-k vector such that $d|k$, and the norm-bound on the NTT can be extended to $\{-\gamma, \ldots, 0, \ldots, \gamma\}$ for any $\gamma \in \mathbb{N}$, at the cost of larger proofs.

– **Prove Phase:**

1. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
2. For $\theta \in [\ell]$, do the following:
   (a) Sample $\mathbf{g}_\theta \leftarrow\!\$\ \{\mathbf{g} \in R_q : g_0 = 0\}$.
   (b) Sample $\vec{\mathbf{y}}_\theta \leftarrow D^m_{R,\mathfrak{s}_2}$ and compute $\vec{\mathbf{w}}_\theta \leftarrow \mathbf{B}_0\vec{\mathbf{y}}_\theta$.
   (c) Compute $\mathbf{t}_{\ell+2\theta} \leftarrow \langle\vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}\rangle + \langle\vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_\theta\rangle - 3\mathbf{s}\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta\rangle^2$.
   (d) Compute $\mathbf{t}_{\ell+2\theta+1} \leftarrow \langle\vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}\rangle + (3\mathbf{s}^2 - 1)\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta\rangle$.
   (e) Compute $\mathbf{v}_\theta \leftarrow \langle\vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_\theta\rangle + \langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta\rangle^3$.
3. Compute $\vec{\mathbf{t}}_2 \leftarrow [\vec{\mathbf{b}}_2 \mid \cdots \mid \vec{\mathbf{b}}_{1+\ell}]^\top\vec{\mathbf{r}} + [\mathbf{g}_1 \mid \cdots \mid \mathbf{g}_\ell]^\top$.
4. Set $\vec{\mathbf{t}}_3 := [\mathbf{t}_{\ell+2} \mid \cdots \mid \mathbf{t}_{3\ell+1}]^\top$ and $\vec{\mathbf{v}} := [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell]^\top$.
5. ▶ Compute $(\vec{v}_\theta)_{\theta=1}^\ell \leftarrow H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell)$, where $\vec{v}_\theta \in \mathbb{Z}_q^\nu$.
6. For $\theta \in [\ell]$, let $\boldsymbol{\psi}_\theta \leftarrow \mathsf{NTT}^{-1}(dA^\top\vec{v}_\theta)$.
7. Compute $\vec{\mathbf{h}} \leftarrow \mathbf{s} \cdot \begin{bmatrix}\boldsymbol{\psi}_1\\ \vdots \\ \boldsymbol{\psi}_\ell\end{bmatrix} + \begin{bmatrix}\mathbf{g}_1\\ \vdots \\ \mathbf{g}_\ell\end{bmatrix}$ and $\vec{\mathbf{v}}' \leftarrow \begin{bmatrix}\langle\boldsymbol{\psi}_1\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{y}}_1\rangle\\ \vdots \\ \langle\boldsymbol{\psi}_\ell\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{y}}_\ell\rangle\end{bmatrix}$.
8. ▶ Compute $(\mathbf{c}_\theta)_{\theta=1}^\ell \leftarrow H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}')$.
9. For $\theta \in [\ell]$, compute $\vec{\mathbf{z}}_\theta \leftarrow \vec{\mathbf{y}}_\theta + \mathbf{c}_\theta\vec{\mathbf{r}}$.
10. Return $\left(\vec{\mathbf{t}}, \pi := \left(\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^\ell\right)\right)$.

- $\mathsf{Verify}(\mathsf{pp}, \mathbb{x} := (A, \vec{u}), (\vec{\mathbf{t}}, \pi))$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(A, \vec{u}) := \mathbb{x}$ commitment $\vec{\mathbf{t}} \in R_q^{n+1}$ and proof $\left(\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^\ell\right) := \pi$, parse $\left[\vec{\mathbf{t}_0}^\top \mid \mathbf{t}_1\right]^\top := \vec{\mathbf{t}}$ and proceed as follows:

1. Abort and output 0 if any of the following are true:
   – There exists $\theta \in [\ell]$ such that $\|\vec{\mathbf{z}}_\theta\| > \beta$ or $\mathbf{B}_0\vec{\mathbf{z}_\theta} \neq \vec{\mathbf{w}}_\theta + \mathbf{c}_\theta\vec{\mathbf{t}}_0$.
   – $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell) \neq (\vec{v}_\theta)_{\theta=1}^\ell$.
   – $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}') \neq (\mathbf{c}_\theta)_{\theta=1}^\ell$.
   – $\mathsf{ct}(\vec{\mathbf{h}}) \neq [\langle\vec{u}, \vec{v}_1\rangle \mid \cdots \mid \langle\vec{u}, \vec{v}_\ell\rangle]^\top$.
2. For $\theta \in [\ell]$, compute $\boldsymbol{\psi}_\theta \leftarrow \mathsf{NTT}^{-1}(dA^\top\vec{v}_\theta)$, and $\boldsymbol{\tau}_\theta \leftarrow \boldsymbol{\psi}_\theta\mathbf{t}_1$.
3. Abort and output 0 if
$$\begin{bmatrix}\langle\boldsymbol{\psi}_1\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{z}}_1\rangle\\ \vdots \\ \langle\boldsymbol{\psi}_\ell\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{z}}_\ell\rangle\end{bmatrix} \neq \vec{\mathbf{v}}' + \begin{bmatrix}\mathbf{c}_1\\ \vdots \\ \mathbf{c}_\ell\end{bmatrix} \odot \left(\begin{bmatrix}\boldsymbol{\tau}_1\\ \vdots \\ \boldsymbol{\tau}_\ell\end{bmatrix} + \vec{\mathbf{t}}_2 - \vec{\mathbf{h}}\right)$$
4. Parse $[\mathbf{t}_{2+\ell} \mid \cdots \mid \mathbf{t}_{3\ell+1}] := \vec{\mathbf{t}}_3$ and $[\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell] := \vec{\mathbf{v}}$.
5. For $\theta \in [\ell]$, do the following:
   (a) Compute $[\mathbf{f}_{\theta,1} \mid \mathbf{f}_{\theta,2} \mid \mathbf{f}_{\theta,3}]^\top \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_{\ell+2\theta} \mid \vec{\mathbf{b}}_{\ell+2\theta+1}]^\top\vec{\mathbf{z}}_\theta - \mathbf{c}_\theta \cdot [\mathbf{t}_1 \mid \mathbf{t}_{\ell+2\theta} \mid \mathbf{t}_{\ell+2\theta+1}]^\top$.
   (b) If $\mathbf{f}_{\theta,1}(\mathbf{f}_{\theta,1} - \mathbf{c}_\theta)(\mathbf{f}_{\theta,1} + \mathbf{c}_\theta) + \mathbf{f}_{\theta,2} + \mathbf{c}_\theta \cdot \mathbf{f}_{\theta,3} \neq \mathbf{v}_\theta$, abort and output 0.
6. Output 1.

Theorem 6.1 (vENS Completeness)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Then Construction 3 is complete except with negligible probability.*

Theorem 6.2 (vENS ZK)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$. Then Construction 3 is zero-knowledge against*

any ppt *distinguisher making at most Q_H random oracle queries and Q_P proof queries, under the hardness of the Q_P-*$\mathsf{Hint\text{-}MLWE}^{\ell,\mathfrak{s}_2,\mathcal{C}}_{m-n-3\ell-1,n+3\ell+1,q,\mathfrak{s}_1}$ *assumption.*

We prove Theorem 6.1 in Appendix E.1 and Theorem 6.2 in Appendix E.2. The proofs follow the strategies developed for Theorems 5.1 and 5.2, adapted to account for the repetition factor ℓ.

Theorem 6.3 (vENS WUR)**.** *For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be prime, and let $n = n(\lambda)$, $m = m(\lambda)$, $d = d(\lambda)$ be lattice parameters. Let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Then Construction 3 has weak unique response with respect to Simulator 3 (see Definition 3.7) under the hardness of* $\mathsf{MSIS}_{n,m+1,q,2\sqrt{\beta^2+d}}$.

We prove Theorem 6.3 in Appendix E.3. At a high level, if two proofs π, π^* share their first message and have distinct third messages $(\vec{\mathbf{h}}, \vec{\mathbf{v}}') \neq (\vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**})$, then with overwhelming probability there are pairs $(\mathbf{c}_j, \vec{\mathbf{z}}_j)$ in π and $(\mathbf{c}_j^*, \vec{\mathbf{z}}_j^*)$ in π^* with $\mathbf{B}_0(\vec{\mathbf{z}}_j - \vec{\mathbf{z}}_j^*) + (\mathbf{c}_j^* - \mathbf{c}_j)\mathbf{t}_0 = \mathbf{0}^n$.

Theorem 6.4 (vENS Knowledge Soundness)**.** *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_i)$ for any $i \in [h]$ (see Lemma 2.6). Under the hardness of* $\mathsf{MSIS}_{n,m,q,8d\beta}$*, Construction 3 is adaptively knowledge-sound with knowledge error at most $2(Q+1)((3M)^\ell + 2(hM)^\ell + q^{-\ell} + M)$, where Q is the number of random oracle queries made by* P*. In addition, the extractor makes $5Q + 6$ queries to* P *in expectation.*

Similar to Theorem 5.4, we prove Theorem 6.4 by showing that vENS satisfies predicate special soundness (see Appendix E.4).

Corollary 6.4.1 (vENS Simulation-Extractable)**.** *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_i)$ for any $i \in [h]$ (see Lemma 2.6). Under the hardness of* $\mathsf{MSIS}_{n,m,q,8d\beta}$*, Construction 3 is weakly simulation-extractable.*

Proof. This follows from Theorems 6.3 and 6.4 and Lemma 3.1. □

6.2 Threshold ENS

In this section, we present a thresholdized version of Construction 3. As in the thresholdized version of Construction 1, we use a combination of a linear threshold secret-sharing scheme and a leveled tHE scheme: specifically, one supporting homomorphic evaluation of polynomials of total degree at most three.

Construction 4: Threshold vENS

Let $\Pi_{\mathsf{tHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{TDec}, \mathsf{Comb})$ be a threshold homomorphic encryption scheme supporting the evaluation of polynomials of total degree at most three and $\Pi_{\mathsf{SS}} = (\mathsf{Share}, \mathsf{Reconstruct})$ be Shamir secret sharing scheme. For any $\lambda \in \mathbb{N}$, let $q = q(\lambda)$ be a prime. Let $n = n(\lambda)$, $m = m(\lambda)$ be matrix dimensions. Let $\mathfrak{s}_1 = \mathfrak{s}_1(\lambda)$ and $\mathfrak{s}_2 = \mathfrak{s}_2(\lambda)$ be standard deviations, and $\beta = \beta(\lambda)$ be a norm bound. Let $\ell = \ell(\lambda)$ be a repetition parameter and $\mathcal{C}$ be the challenge distribution described in Section 2.4. Let $H_1 : R_q^{n+3\ell+1} \times R_q^\ell \times R_q^{n\ell} \to (\mathbb{Z}_q^\nu)^\ell$ and $H_2 : R_q^{n+3\ell+1} \times R_q^\ell \times R_q^{n\ell} \times R_q^{2\ell} \to \mathcal{C}^\ell$ be random oracles. We construct a threshold variant of Construction 3 as follows:

- $\mathsf{Setup}(1^\lambda, \mathtt{n}, \mathtt{t})$: Given the security parameter $\lambda \in \mathbb{N}$ and the threshold parameters $\mathtt{n}, \mathtt{t} \in \mathbb{N}$, do the following:
  1. Compute $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}) \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t})$.
  2. Sample $\mathbf{B}_0 \leftarrow\$ R_q^{n\times m}$ and $\mathbf{B}_1 \leftarrow\$ R_q^{(3\ell+1)\times m}$.
  3. Return $\mathsf{pp} = (\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1)$ and $\mathsf{paux}_i = \mathsf{sk}_i$ to each prover P_i.
- $\mathsf{WitnessShare}(\mathsf{pp}, \mathbb{w})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $\mathbf{s} := \mathbb{w}$, do the following:

1. Compute $\mathsf{ctx}_{\mathbf{s}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{s})$ and $\{[\![\mathbf{s}]\!]_i\}_{i\in[\mathtt{n}]} \leftarrow \Pi_{\mathsf{SS}}.\mathsf{Share}(\mathbf{s}, (\mathtt{n}, \mathtt{t}))$.
2. Set $\mathsf{aux}_w := \mathsf{ctx}_{\mathbf{s}}$.
3. For $i \in [\mathtt{n}]$, set $\mathsf{pwit}_i = [\![\mathbf{s}]\!]_i$ and send $(\mathsf{pwit}_i, \mathsf{aux}_w)$ to prover P_i.

- $\mathsf{Commit}(\mathsf{pp}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w)$: This is an interactive protocol executed by provers in $\mathcal{S}$. Each prover $\mathsf{P}_i \in \mathcal{S}$ has inputs $(\mathsf{pp}, \mathcal{S}, \mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{aux}_w)$ and proceeds as follows:
  1. Parse $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $\mathsf{sk}_i := \mathsf{paux}_i$, $[\![\mathbf{s}]\!]_i := \mathsf{pwit}_i$ and $\mathsf{ctx}_{\mathbf{s}} := \mathsf{aux}_w$.
  2. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_j \in R_q^m$.
  3. Sample $\vec{\mathbf{r}}_i \leftarrow D^m_{R,\mathfrak{s}_1}$ and compute $\vec{\mathbf{t}}_{0,i} \leftarrow \mathbf{B}_0\vec{\mathbf{r}}_i$.
  4. Use the set $\mathcal{S}$ to derive the i-th Lagrange coefficient λ_i, and then compute $\mathbf{t}_{1,i} \leftarrow \langle\vec{\mathbf{b}}_1, \vec{\mathbf{r}}_i\rangle + \lambda_i \cdot [\![\mathbf{s}]\!]_i$.
  5. ▶ Broadcast $(\vec{\mathbf{t}}_{0,i}, \mathbf{t}_{1,i})$ to all other participants in $\mathcal{S}$.
  6. Upon receiving $(\vec{\mathbf{t}}_{0,j}, \mathbf{t}_{1,j})$ from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$ proceed as follows:
     (a) Compute $\vec{\mathbf{t}}_0 \leftarrow \sum_{k\in\mathcal{S}} \vec{\mathbf{t}}_{0,k}$; $\mathbf{t}_1 \leftarrow \sum_{k\in\mathcal{S}} \mathbf{t}_{1,k}$.
     (b) Set $\mathsf{pr}_i := \vec{\mathbf{r}}_i$ and $\vec{\mathbf{t}} := [\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1]^\top$.
  7. Store pr_i and output $\vec{\mathbf{t}}$.

- $\mathsf{Prove}(\mathsf{pp}, \mathbb{x} := (A, \vec{u}), \vec{\mathbf{t}}, \mathcal{S}, \{(\mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i)\}_{i\in\mathcal{S}}, \mathsf{aux}_w)$: This is an interactive protocol executed by provers in $\mathcal{S}$. Each prover $\mathsf{P}_i \in \mathcal{S}$ has inputs $(\mathsf{pp}, \mathbb{x}, \mathcal{S}, \vec{\mathbf{t}}, \mathsf{paux}_i, \mathsf{pwit}_i, \mathsf{pr}_i, \mathsf{aux}_w)$ and proceeds as follows:
  1. Parse $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $[\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1]^\top := \vec{\mathbf{t}}$, $\mathsf{sk}_i := \mathsf{paux}_i$, $[\![\mathbf{s}]\!]_i := \mathsf{pwit}_i$, $\vec{\mathbf{r}}_i := \mathsf{pr}_i$, and $\mathsf{ctx}_{\mathbf{s}} := \mathsf{aux}_w$.
  2. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_j \in R_q^m$.
  3. Compute $\vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{\mathbf{r}}_i)$.
  4. For $\theta \in [\ell]$, do the following:
     (a) Sample $\mathbf{g}_{\theta,i} \leftarrow\!\!\$\, \{\mathbf{g} \in R_q : g_0 = 0\}$; $\vec{\mathbf{y}}_{\theta,i} \leftarrow D^m_{R,\mathfrak{s}_2}$
     (b) Compute $\vec{\mathbf{w}}_{\theta,i} \leftarrow \mathbf{B}_0\vec{\mathbf{y}}_{\theta,i}$; $\mathbf{t}_{\mathbf{g}_{\theta,i}} \leftarrow \langle\vec{\mathbf{b}}_{\theta+1}, \vec{\mathbf{r}}_i\rangle + \mathbf{g}_{\theta,i}$
     (c) Compute $\vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_{\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{\mathbf{y}}_{\theta,i})$; $\vec{\mathsf{ctx}}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i}\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i}\rangle)$
  5. Compute $\vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{B}'_1\vec{\mathbf{y}}_i)$; $\vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{B}'_1\vec{\mathbf{r}}_i)$ where $\mathbf{B}'_1 = [\vec{\mathbf{b}}_\ell \mid \ldots \mid \vec{\mathbf{b}}_{3\ell+1}]$, $\vec{\mathbf{y}}_i = [\vec{\mathbf{y}}_{1,i} \mid \ldots \mid \vec{\mathbf{y}}_{\theta,i}]$
  6. ▶ Broadcast $\big(\mathsf{ctx}_{\vec{\mathbf{r}}_i}, (\mathbf{t}_{\mathbf{g}_{\theta,i}}, \vec{\mathbf{w}}_{\theta,i}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,i}}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i}\rangle})_{\theta=1}^{\ell}\big), \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}_{\theta,i}}, \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{r}}_i}$ to other participants in $\mathcal{S}$.
  7. Let $\big(\mathsf{ctx}_{\vec{\mathbf{r}}_j}, (\mathbf{t}_{\mathbf{g}_{\theta,j}}, \vec{\mathbf{w}}_{\theta,j}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,j}}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,j}\rangle})_{\theta=1}^{\ell}\big), \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}_{\theta,j}}, \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{r}}_j}$ be the values received from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$
  8. For $\theta \in [\ell]$, compute:
     – $\mathbf{t}_{\mathbf{g}_\theta} \leftarrow \sum_{j\in\mathcal{S}} \mathbf{t}_{\mathbf{g}_{\theta,j}}$; $\vec{\mathbf{w}}_\theta \leftarrow \sum_{j\in\mathcal{S}} \vec{\mathbf{w}}_{\theta,j}$; $\vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_\theta} \leftarrow \sum_{j\in\mathcal{S}} \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_{\theta,j}}$
     – $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta\rangle} \leftarrow \sum_{j\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,j}\rangle}$
  9. Compute $\vec{\mathsf{ctx}}_{\vec{\mathbf{r}}} \leftarrow \sum_{j\in\mathcal{S}} \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_j}$.
  10. Compute $\vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}} \leftarrow \sum_{j\in\mathcal{S}} \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}_j}$; $\vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{r}}} \leftarrow \sum_{j\in\mathcal{S}} \vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{r}}_j}$
  11. Let $\vec{\mathsf{ctx}}_{\mathbf{B}'_1\vec{\mathbf{y}}} = \Big[\mathsf{ctx}_{\langle\vec{\mathbf{b}}_\ell, \vec{\mathbf{y}}_\ell\rangle} \mid \ldots \mid \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{3\ell+1}, \vec{\mathbf{y}}_{3\ell}\rangle}\Big]^\top$
  12. For $\theta \in [\ell]$, compute:
      (a) $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}\rangle} + \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_\theta\rangle} - 3 \cdot \mathsf{ctx}_{\mathbf{s}} \cdot \Big(\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta\rangle}\Big)^2$

(b) $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}} \leftarrow \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}} \rangle} + (3 \cdot (\mathsf{ctx}_{\mathbf{s}})^2 - 1) \cdot \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta \rangle}$

(c) $\mathsf{ctx}_{\mathbf{v}_\theta} \leftarrow \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_\theta \rangle} + \left(\mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta \rangle}\right)^3$.

13. For $\theta \in [\ell]$, compute:
    (a) $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}}, \mathcal{S})$
    (b) $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}}, \mathcal{S})$
    (c) $\mathsf{ds}_{\mathbf{v}_{\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}_{\mathbf{v}_\theta}, \mathsf{sk}_i)$.
14. ▶ Broadcast $(\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,i}}, \mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,i}}, \mathsf{ds}_{\mathbf{v}_{\theta,i}})_{\theta=1}^{\ell}$ to other participants in $\mathcal{S}$.
15. Upon receiving $(\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,j}}, \mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,j}}, \mathsf{ds}_{\mathbf{v}_{\theta,j}})_{\theta=1}^{\ell}$ from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$ proceed as follows:
    (a) For $\theta \in [\ell]$, compute:
        – $\mathbf{t}_{\ell+2\theta} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}}, \{\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,k}}\}_{k\in\mathcal{S}})$
        – $\mathbf{t}_{\ell+2\theta+1} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}}, \{\mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,k}}\}_{k\in\mathcal{S}})$
        – $\mathbf{v}_\theta \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}_{\mathbf{v}_\theta}, \{\mathsf{ds}_{\mathbf{v}_{\theta,k}}\}_{k\in\mathcal{S}})$.
16. Set $\vec{\mathbf{t}}_2 := [\mathbf{t}_{\mathbf{g}_1} \mid \cdots \mid \mathbf{t}_{\mathbf{g}_\ell}]^\top$.
17. Set $\vec{\mathbf{t}}_3 := [\mathbf{t}_{\ell+2} \mid \cdots \mid \mathbf{t}_{3\ell+1}]^\top$ and $\vec{\mathbf{v}} := [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell]^\top$.
18. ▶ Compute $(\vec{v}_\theta)_{\theta=1}^{\ell} \leftarrow H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell})$, where $\vec{v}_\theta \in \mathbb{Z}_q^\nu$.
19. For $\theta \in [\ell]$, let $\boldsymbol{\psi}_\theta \leftarrow \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta)$.
20. Use the set $\mathcal{S}$ to derive the i-th Lagrange coefficient λ_i.
21. Compute $\vec{\mathbf{h}}_i \leftarrow \lambda_i [\![\mathbf{s}]\!]_i \cdot \begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_\ell \end{bmatrix} + \begin{bmatrix} \mathbf{g}_{1,i} \\ \vdots \\ \mathbf{g}_{\ell,i} \end{bmatrix}$ and $\vec{\mathbf{v}}'_i \leftarrow \begin{bmatrix} \langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{y}}_{1,i} \rangle \\ \vdots \\ \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{y}}_{\ell,i} \rangle \end{bmatrix}$
22. ▶ Broadcast $(\vec{\mathbf{h}}_i, \vec{\mathbf{v}}'_i)$ to other participants in $\mathcal{S}$.
23. Upon receiving $(\vec{\mathbf{h}}_j, \vec{\mathbf{v}}'_j)$ from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$, proceed as follows: Compute $\vec{\mathbf{h}} \leftarrow \sum_{k\in\mathcal{S}} \vec{\mathbf{h}}_k$; $\vec{\mathbf{v}}' \leftarrow \sum_{k\in\mathcal{S}} \vec{\mathbf{v}}'_k$
24. ▶ Compute $(\mathbf{c}_\theta)_{\theta=1}^{\ell} \leftarrow H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}')$.
25. For $\theta \in [\ell]$, do the following:
    (a) Compute $\mathsf{ctx}_{\vec{\mathbf{z}}_\theta} \leftarrow \mathsf{ctx}_{\vec{\mathbf{y}}_\theta} + \mathbf{c}_\theta \cdot \mathsf{ctx}_{\vec{\mathbf{r}}}$; $\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}_{\vec{\mathbf{z}}_\theta}, \mathcal{S})$.
26. ▶ Broadcast $(\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}})_{\theta=1}^{\ell}$ to all other participants in $\mathcal{S}$.
27. Upon receiving $(\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,j}})_{\theta=1}^{\ell}$ from all participants $P_j \in \mathcal{S} \setminus \{P_i\}$, compute $\vec{\mathbf{z}}_\theta \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}_{\vec{\mathbf{z}}_\theta}, \{\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,k}}\}_{k\in\mathcal{S}})$ for $\theta \in [\ell]$.
28. Output $\left(\vec{\mathbf{t}}, \pi := \left(\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^{\ell}\right)\right)$.

Theorem 6.5. *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, and let $\mathtt{n} = \mathtt{n}(\lambda) \geq \mathtt{t} = \mathtt{t}(\lambda)$. Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $d = d(\lambda)$ be lattice parameters. Let $\mathfrak{s}_1, \mathfrak{s}_2 \geq \sqrt{2} \cdot \eta_\varepsilon(R^m)$. Suppose that Π_{tHE} is semantically and simulation secure, and vanilla ENS (Construction 3) is zero-knowledge.* [6] *In addition, suppose $\mathsf{Hint\text{-}MLWE}^{\ell,\mathfrak{s}_2,\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$, $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$, and $\ell\text{-}\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_2}$ hold, where $\bar{n} = n + 3\ell + 1$. Then Construction 4 satisfies weak $(\mathtt{t}-1)$-ZK (see Definition 4.3) against any semi-honest adversary that statically corrupts at most $\mathtt{t}-1$ provers and makes at most Q_H random oracle queries and Q_P proof queries.*

[6] Similar to the case of ALS, computing a thresholdized proof on an invalid witness $\mathbf{s} \in R_q$ for a statement $\mathbb{x} = (A, \vec{u})$ reveals nontrivial leakage about $\mathbf{s}$, beyond the fact that either $A \cdot \mathsf{NTT}(\mathbf{s}) \neq \vec{u}$ or $\mathsf{NTT}(\mathbf{s}) \notin \{0, \pm 1\}^d$. However, since we assume that witnesses are always valid, this is not an issue.

Theorem 6.6. *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathtt{n} = \mathtt{n}(\lambda) \geq \mathtt{t} = \mathtt{t}(\lambda)$. Let $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $d = d(\lambda)$ be lattice parameters. Let $\mathfrak{s}_1, \mathfrak{s}_2 \geq \sqrt{2} \cdot \eta_\varepsilon(R^m)$ be standard deviations. Set $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$ and $\beta' = \beta\sqrt{\mathtt{t}}$. Assume that (i) the $\mathsf{MSIS}_{n,m+1,q,8d\beta'}$ assumption holds, (ii) vanilla ENS (Construction 3) is weakly simulation-extractable, and (iii) Construction 4 is weakly $(\mathtt{t}-1)$-zero-knowledge (Theorem 6.5). Then Construction 4 satisfies threshold simulation extractability (Definition 4.4).*

We prove Theorem 6.5 in Appendix E.5, and Theorem 6.6 in Appendix E.6.

7 Hardness of MLWE with Secrets Fixed by Automorphisms

Throughout this section, fix powers of two $\ell < h \leq d$ and consider the ring $\mathsf{R}_{h,\ell} = \mathbb{Z}_q[X]/(X^{h/\ell}+1)$. From Esgin, Nguyen and Seiler [ENS20], it follows that $\mathsf{R}_{h,\ell} \cong R_q^{\langle\sigma_{2h/\ell+1}\rangle}$ via the evaluation map $E_{X^{\ell d/h}} : \mathsf{R}_{h,\ell} \to R_q^{\langle\sigma_{2h/\ell+1}\rangle}$ that maps $\mathbf{c}(X) \in \mathsf{R}_{h,\ell}$ to $\mathbf{c}(X^{\ell d/h}) \in R_q^{\langle\sigma_{2h/\ell+1}\rangle}$, where $R_q^{\langle\sigma_{2h/\ell+1}\rangle} \subseteq R_q$ is the subring of R_q made of elements fixed by the automorphisms generated by $\sigma_{2h/\ell+1}$. [7] For any $\xi \geq 0$, let $S_{\xi,\mathsf{R}_{h,\ell}} = \{\mathbf{c}(X^{\ell d/h}) \in R_q : \mathbf{c} \in \mathsf{R}_{h,\ell} \wedge \|\mathbf{c}\|_\infty \leq \xi\}$. We prove that $\widehat{\mathsf{MLWE}}$ with secrets sampled from $U(S^n_{\xi,\mathsf{R}_{h,\ell}})$ is hard assuming the hardness of $\widehat{\mathsf{MLWE}}$ and $\widehat{\mathsf{ext\text{-}MLWE}}$ with secrets sampled from $U(R_q)$. This will be sufficient because if $S_{\xi,\mathsf{R}_{h,\ell}} \setminus \{0\} \subseteq R_q^\times$, then the assumption on the hardness of $\widehat{\mathsf{ext\text{-}MLWE}}$ can be discarded.

Here the $\widehat{\mathsf{ext\text{-}MLWE}}$ assumption states that $\widehat{\mathsf{MLWE}}$ is hard even when an adversary is given hints of the form $\langle\vec{\mathbf{e}}, \vec{\mathbf{z}}\rangle$, where $\vec{\mathbf{e}}$ is the $\widehat{\mathsf{MLWE}}$ error vector and $\vec{\mathbf{z}}$ is a vector chosen by the adversary. σ represents the canonical embedding [LPR13].

Theorem 7.1. *Let $\lambda, n, m, k, q, \ell, h \in \mathbb{N}_{>0}$ such that $\ell < h \leq d$ are powers of two and q is an unramified prime with inertia degree $r > 0$ in R. Let $g = d/r$ and $\xi \geq 0$ such that $(\frac{nhr}{\ell d})\log(2\xi+1) \geq kr\log q + \omega(\log\lambda) + \log g$. Let $\varepsilon = \mathsf{negl}(\lambda)$, $B = \max_{\mathbf{x}\in S_{\xi,\mathsf{R}_{h,\ell}}}\|\sigma(\mathbf{x})\|_\infty$, $\alpha \geq \sqrt{2}\cdot\eta_\varepsilon((q^{-1}R)^n)$, $\gamma = \alpha B\sqrt{n}$, and $\beta = \alpha B\sqrt{2n}$. If the hardness of $\widehat{\mathsf{ext\text{-}MLWE}}^m_{k,n,q,\psi,S^n_{\xi,\mathsf{R}_{h,\ell}}}(U(R_q))$, $\widehat{\mathsf{MLWE}}_{k,m,q,D_\gamma}(U(R_q))$, and $\widehat{\mathsf{ext\text{-}MLWE}}^m_{k,n,q,\psi,\{0^d\}}(U(R_q))$ hold, where $\psi = D_{\frac{1}{q}R,\alpha}$, then $\widehat{\mathsf{MLWE}}_{n,m,q,\Psi_{\leq\beta}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ is hard.*

The proof of Theorem 7.1 follows the template of Boudgoust et al. [BJRW23] for the case of $\widehat{\mathsf{MLWE}}$ with uniform secrets of bounded ℓ_∞-norm over R_q, and is given in Appendix F.

Acknowledgments

The authors thank Anna Lysyanskaya for fruitful discussions on this project. This work was developed in connection with the workshop Foundations and Applications of Zero-Knowledge Proofs hosted at the International Centre for Mathematical Sciences (ICMS) in Edinburgh. Scott Griffy was supported by NSF Grants 2312241, 2154170, and 2247305 and the Ethereum Foundation.

References

[AAB+24] Marius A. Aardal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. Aggregating falcon signatures with LaBRADOR. In Leonid Reyzin and Douglas Stebila, editors,

[7] The strict inequality $\ell < h$ guarantees that $\sigma_{2h/\ell+1}$ generates the *full* stabilizer of $X^{\ell d/h}$ in $(\mathbb{Z}/2d)^\times$. At $\ell = h$ the stabilizer is the entire (non-cyclic) Galois group.

CRYPTO 2024, Part I, volume 14920 of *LNCS*, pages 71–106. Springer, Cham, August 2024. https://doi.org/10.1007/978-3-031-68376-3_3

[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-Shamir transformation of multi-round interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 113–142. Springer, Cham, November 2022. https://doi.org/10.1007/978-3-031-22318-1_5

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 470–499. Springer, Cham, August 2020. https://doi.org/10.1007/978-3-030-56880-1_17

[ASY22] Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Round-optimal lattice-based threshold signatures, revisited. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *ICALP 2022*, volume 229 of *LIPIcs*, pages 8:1–8:20. Schloss Dagstuhl, July 2022. https://doi.org/10.4230/LIPIcs.ICALP.2022.8

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. https://doi.org/10.1109/SP.2014.36

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Cham, September 2018. https://doi.org/10.1007/978-3-319-98113-0_20

[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 Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 565–596. Springer, Cham, August 2018. https://doi.org/10.1007/978-3-319-96884-1_19

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012. https://doi.org/10.1145/2090236.2090262

[BJRW23] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module learning with errors with short distributions. *Journal of Cryptology*, 36(1):1, January 2023. https://doi.org/10.1007/s00145-022-09441-3

[BL26] Katharina Boudgoust and Oleksandra Lapiha. Leftover hash lemma(s) over cyclotomic rings. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part IV*, volume 16545 of *LNCS*, pages 62–91. Springer, Cham, May 2026. https://doi.org/10.1007/978-3-032-25327-9_3

[BLNS23] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. A framework for practical anonymous credentials from lattices. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 384–417. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38545-2_13

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013. https://doi.org/10.1145/2488608.2488680

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Cham, August 2019. https://doi.org/10.1007/978-3-030-26948-7_7

[BS23] Ward Beullens and Gregor Seiler. LaBRADOR: Compact proofs for R1CS from module-SIS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 518–548. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38554-4_17

[BTT22] Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi. MuSig-L: Lattice-based multi-signature with single-round online phase. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 276–305. Springer, Cham, August 2022. https://doi.org/10.1007/978-3-031-15979-4_10

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011. https://doi.org/10.1109/FOCS.2011.12

[CATZ24] Rutchathon Chairattana-Apirom, Stefano Tessaro, and Chenzhi Zhu. Partially non-interactive two-round lattice-based threshold signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 268–302. Springer, Singapore, December 2024. https://doi.org/10.1007/978-981-96-0894-2_9

[CG08] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In Peng Ning, Paul F. Syverson, and Somesh Jha, editors, *ACM CCS 2008*, pages 345–356. ACM Press, October 2008. https://doi.org/10.1145/1455770.1455814

[CL03] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02*, volume 2576 of *LNCS*, pages 268–289. Springer, Berlin, Heidelberg, September 2003. https://doi.org/10.1007/3-540-36413-7_20

[CL06] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 78–96. Springer, Berlin, Heidelberg, August 2006. https://doi.org/10.1007/11818175_5

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002. https://doi.org/10.1145/509907.509980

[Des88] Yvo Desmedt. Society and group oriented cryptography: A new concept. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 120–127. Springer, Berlin, Heidelberg, August 1988. https://doi.org/10.1007/3-540-48184-2_8

[DF90] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 307–315. Springer, New York, August 1990. https://doi.org/10.1007/0-387-34805-0_28

[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 Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248. Springer, Cham, May 2024. https://doi.org/10.1007/978-3-031-58723-8_8

[EG14] Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649. Springer, Berlin, Heidelberg, March 2014. https://doi.org/10.1007/978-3-642-54631-0_36

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Cham, December 2020. https://doi.org/10.1007/978-3-030-64834-3_9

[ESLR23] Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and VRFs. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 484–517. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38554-4_16

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the Fiat-Shamir transform. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT 2012*, volume 7668 of *LNCS*, pages 60–79. Springer, Berlin, Heidelberg, December 2012. https://doi.org/10.1007/978-3-642-34931-7_5

[FLL25] Andrea Flamini, Eysa Lee, and Anna Lysyanskaya. Multi-holder anonymous credentials from BBS signatures. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 325–357. Springer, Cham, August 2025. https://doi.org/10.1007/978-3-032-01887-8_11

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Berlin, Heidelberg, August 1987. https://doi.org/10.1007/3-540-47721-7_12

[GGJ+23] Sanjam Garg, Aarushi Goel, Abhishek Jain, Guru-Vamsi Policharla, and Sruthi Sekar. zkSaaS: Zero-knowledge SNARKs as a service. In Joseph A. Calandrino and Carmela Troncoso, editors, *USENIX Security 2023*, pages 4427–4444. USENIX Association, August 2023. https://www.usenix.org/conference/usenixsecurity23/presentation/garg

[GGJ+25] Sanjam Garg, Aarushi Goel, Abhishek Jain, Bhaskar Roberts, and Sruthi Sekar. Malicious security in collaborative zk-SNARKs: More than meets the eye. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 396–428. Springer, Cham, August 2025. https://doi.org/10.1007/978-3-032-01907-3_13

[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 Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *PQCrypto 2024, Part II*, pages 266–300. Springer, Cham, June 2024. https://doi.org/10.1007/978-3-031-62746-0_12

[GKV10] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A group signature scheme from lattice assumptions. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 395–412. Springer, Berlin, Heidelberg, December 2010. https://doi.org/10.1007/978-3-642-17373-8_23

[GM17] Jens Groth and Mary Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable SNARKs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 581–612. Springer, Cham, August 2017. https://doi.org/10.1007/978-3-319-63715-0_20

[GMPW20] Nicholas Genise, Daniele Micciancio, Chris Peikert, and Michael Walter. Improved discrete gaussian and subgaussian analysis for lattice cryptography. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 623–651. Springer, Cham, May 2020. https://doi.org/10.1007/978-3-030-45374-9_21

[GOP+25] Chaya Ganesh, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Fiat–shamir bulletproofs are non-malleable (in the random oracle model). *Journal of Cryptology*, 38(1):11, 2025. https://doi.org/10.1007/s00145-024-09525-2

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. https://doi.org/10.1145/1374376.1374407

[HMM+26] Lucjan Hanzlik, Mark Manulis, Marzio Mula, Alan Pulval-Dady, Tjerand Silde, and Daniel Slamanig. BRaccoon: Concurrently secure blind lattice signatures from raccoon. Cryptology ePrint Archive, Paper 2026/1084, 2026. https://eprint.iacr.org/2026/1084

[JRS17] Aayush Jain, Peter M. R. Rasmussen, and Amit Sahai. Threshold fully homomorphic encryption. Cryptology ePrint Archive, Report 2017/257, 2017. https://eprint.iacr.org/2017/257

[KLSS23] Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580. Springer, Cham, August 2023. https://doi.org/10.1007/978-3-031-38554-4_18

[KNYY19] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 622–651. Springer, Cham, May 2019. https://doi.org/10.1007/978-3-030-17656-3_22

[LLNW18] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based zero-knowledge arguments for integer relations. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 700–732. Springer, Cham, August 2018. https://doi.org/10.1007/978-3-319-96881-0_24

[LN17] Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 293–323. Springer, Cham, April / May 2017. https://doi.org/10.1007/978-3-319-56620-7_11

[LNO+26] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Michał Osadnik, Gregor Seiler, and Patrick Steuer. LaZer: Lattice-based zero-knowledge arguments for lattice relations. "Preview Writeup": In anticipation of a package submission to the NIST Threshold Call, 2026. https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/LaZer-PW02.pdf

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022. https://doi.org/10.1007/978-3-031-15979-4_3

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1051–1070. ACM Press, November 2020. https://doi.org/10.1145/3372297.3417894

[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 215–241. Springer, Cham, May 2021. https://doi.org/10.1007/978-3-030-75245-3_9

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124. Springer, Berlin, Heidelberg, February / March 2013. https://doi.org/10.1007/978-3-642-36362-7_8

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Berlin, Heidelberg, May 2013. https://doi.org/10.1007/978-3-642-38348-9_3

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 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 Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Cham, April / May 2018. https://doi.org/10.1007/978-3-319-78381-9_8

[LSS24] Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3125–3137. ACM Press, October 2024. https://doi.org/10.1145/3658644.3690330

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Berlin, Heidelberg, April 2012. https://doi.org/10.1007/978-3-642-29011-4_43

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004. https://doi.org/10.1109/FOCS.2004.72

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 735–763. Springer, Berlin, Heidelberg, May 2016. https://doi.org/10.1007/978-3-662-49896-5_26

[NO07] Takashi Nishide and Kazuo Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 343–360. Springer, Berlin, Heidelberg, April 2007. https://doi.org/10.1007/978-3-540-71677-8_23

[OB22] Alex Ozdemir and Dan Boneh. Experimenting with collaborative zk-SNARKs: Zero-knowledge proofs for distributed secrets. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022*, pages 4291–4308. USENIX Association, August 2022. https://www.usenix.org/conference/usenixsecurity22/presentation/ozdemir

[PKN+25] Rafaël Del Pino, Shuichi Katsumata, Guilhem Niot, Michael Reichle, and Kaoru Takemure. Unmasking traccoon: A lattice-based threshold signature with an efficient identifiable abort protocol. In *Advances in Cryptology – CRYPTO 2025, Part VI*, volume 16005 of *Lecture Notes in Computer Science*, pages 423–456. Springer, 2025. https://doi.org/10.1007/978-3-032-01887-8_14

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009. https://doi.org/10.1145/1568318.1568324

[Sei18] Gregor Seiler. Faster AVX2 optimized NTT multiplication for ring-LWE lattice cryptography. Cryptology ePrint Archive, Paper 2018/039, 2018. https://eprint.iacr.org/2018/039

[Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979. https://doi.org/10.1145/359168.359176

A Additional Preliminaries

A.1 Algebraic Number Theory Background

The m-th cyclotomic field is a field extension $K = \mathbb{Q}(\zeta_m)$ of degree $\varphi(m) \in \mathbb{N}$, where $\zeta_m \in \mathbb{C}$ is an m-th root of unity and $\varphi(\cdot)$ is Euler's totient function. In our setting, m is a power of two and $K \cong \mathbb{Q}(X)/(X^d + 1)$, with $d = m/2$. The integral closure of $\mathbb{Z}$ in K, also known as the ring of integers of K, is $R = \mathbb{Z}[\zeta_m] \cong \mathbb{Z}[X]/(X^d + 1)$, which is a free $\mathbb{Z}$-module of rank d with basis $\{\zeta_m^i\}_{i=0}^{d-1}$. Let $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R} \cong \mathbb{R}^d$ be the field tensor product and $\mathbb{T} = K_{\mathbb{R}}/R$ be a real torus of dimension d. For any $q \in \mathbb{N}$, let $R_q = R/qR \cong \mathbb{Z}_q[X]/(X^d + 1)$.

Canonical embedding. The field K admits d ring homomorphisms (embeddings) $\sigma_i : K \hookrightarrow \mathbb{C}$, with $i \in \mathbb{Z}_{2d}^\times$, such that σ_i fixes $\mathbb{Q}$ and $\sigma_i(\zeta_{2d}) = \zeta_{2d}^i$. When we identify ζ_{2d} with X, $\sigma_i(X) = X^i$. These homomorphisms are in one-to-one correspondence with elements in the Galois group of K, $\mathrm{Gal}(K/\mathbb{Q})$. Let $H = \{\vec{x} \in \mathbb{R}^r \times \mathbb{C}^{2s} : \forall i \in [s], x_{r+s+i} = \overline{x_{r+i}}\} \subset \mathbb{C}^d$, with $r + 2s = d$. The *canonical embedding* $\sigma : K \to H$ is defined as $\sigma(\mathbf{x}) = (\sigma_i(\mathbf{x}))_{i \in \mathbb{Z}_{2d}^\times}$ for all $\mathbf{x} \in K$. Given that $H \cong \mathbb{R}^d$ (conf. [LPR13, Section 2.1]), one obtains a natural definition of $\|\sigma(\mathbf{x})\|_p$ for any $p \in [1, \infty)$ and $\|\sigma(\mathbf{x})\|_\infty = \max_{i \in \mathbb{Z}_{2d}^\times} |\sigma_i(\mathbf{x})|$. Also, σ can be interpreted as an embedding of K into $K_{\mathbb{R}}$, which will be useful to express distributions over K. For any $\vec{\mathbf{x}} \in K^n$, $\sigma(\vec{\mathbf{x}}) = (\sigma(\mathbf{x}_1), \ldots, \sigma(\mathbf{x}_n)) \in H^n$ and $\|\sigma(\vec{\mathbf{x}})\|_p = (\sum_{i \in [n]} \|\sigma(\mathbf{x}_i)\|_p^p)^{1/p}$ for any $p \in [1, \infty)$ and $\|\sigma(\vec{\mathbf{x}})\|_\infty = \max_{i \in [n]} \|\sigma(\mathbf{x}_i)\|_\infty$.

It is worth nothing that since we are working with powers of two cyclotomic fields, $\|\sigma(\vec{\mathbf{x}})\|_2 = \sqrt{d}\|\vec{\mathbf{x}}\|$, where $\|\vec{\mathbf{x}}\|$ correspond to the norm of $\vec{\mathbf{x}}$ under the coefficient embedding.

Ideals in R. An ideal $\mathcal{I} \subseteq R$ is a non-empty additive subgroup that is closed under multiplication by elements from R, i.e., for any $\mathbf{a} \in \mathcal{I}$ and $\mathbf{r} \in R$, $\mathbf{ra} \in \mathcal{I}$. The norm of $\mathcal{I}$, denoted by $\mathcal{N}(\mathcal{I})$, is defined as $\mathcal{N}(\mathcal{I}) = |R/\mathcal{I}|$ if $\mathcal{I} \neq (0)$. Otherwise, $\mathcal{N}(\mathcal{I}) = 0$. In addition, for any non-empty ideal $\mathcal{J} \neq (0)$ such that $\mathcal{J} \subseteq \mathcal{I}$, we have $\mathcal{N}(\mathcal{I}) = \frac{|R/\mathcal{J}|}{|\mathcal{I}/\mathcal{J}|}$.

Lemma A.1 ([BL26], adapted)**.** *Let $q \in \mathbb{N}$ be an unramified prime with inertia degree $r > 0$ in R and $g = d/r$. Then, $X^d + 1 \equiv \prod_{i=1}^g \Phi_i(X) \bmod q$, where each $\Phi_i(X)$ is an irreducible polynomial of degree r in $\mathbb{F}_q[X]$. And, for any ideal $\mathcal{I} \subseteq R_q$, we have $\mathcal{I} = (\prod_{j \in G} \Phi_j(X))$ and $\mathcal{N}(\mathcal{I}) = q^{r|G|}$, where $G \subseteq [g]$, with $(1) = R_q$ corresponding to $G = \emptyset$.*

Definition A.1 (Expansion factor of R)**.** *Let $\delta_{R,p} = \max_{(\mathbf{a},\mathbf{b}) \in R^2} \frac{\|\mathbf{a} \cdot \mathbf{b}\|_p}{\|\mathbf{a}\|_p \cdot \|\mathbf{b}\|_p}$ denote the expansion factor of R with respect to the ℓ_p-norm [BGV12, LPR13], with $p \in [1, \infty]$. Since we are working with powers of two cyclotomic fields, it follows that $\delta_{R,2} \leq \sqrt{d}$ by Cauchy-Schwarz and $\delta_{R,\infty} \leq d$. This quantity will be useful to bound the noise growth during homomorphic operations.*

A.2 Additional Lattice Background

Remark A.1. Given $\lambda \in \mathbb{N}$, let $\mathfrak{s}_1 \geq \bar{\mathfrak{s}}_1\sqrt{2}$ and $\mathfrak{s}_2 \geq \bar{\mathfrak{s}}_2\sqrt{2}$ be standard deviations such that $\bar{\mathfrak{s}}_1, \bar{\mathfrak{s}}_2 \geq \sqrt{2} \cdot \eta_\varepsilon(R)$ for some $\varepsilon = \mathsf{negl}(\lambda)$. Then $\mathsf{Hint\text{-}MLWE}_{n,m,q,\mathfrak{s}_1}^{\ell,\mathfrak{s}_2,\mathcal{C}}$ is at least as hard as $\mathsf{Hint\text{-}MLWE}_{n,m,q,\bar{\mathfrak{s}}_1}^{\ell,\bar{\mathfrak{s}}_2,\mathcal{C}}$. This follows from Lemma 2.4 and the fact that the secret and hints in $\mathsf{Hint\text{-}MLWE}_{n,m,q,\bar{\mathfrak{s}}_1}^{\ell,\bar{\mathfrak{s}}_2,\mathcal{C}}$ can be shifted by discrete Gaussians with standard deviations $\tilde{\mathfrak{s}}_i = \sqrt{\mathfrak{s}_i^2 - \bar{\mathfrak{s}}_i^2}$ for $i \in \{1, 2\}$ to obtain samples from $\mathsf{Hint\text{-}MLWE}_{n,m,q,\mathfrak{s}_1}^{\ell,\mathfrak{s}_2,\mathcal{C}}$.

Lemma A.2 ([BJRW23])**.** *Let $M \subseteq K^n$ be an R-module, which is a lattice under the canonical embedding, and $\vec{\mathbf{u}}, \vec{\mathbf{z}} \in K^n$. Let $\beta, \gamma > 0$ such that $(1/\beta^2 + \alpha^2/\gamma^2)^{-1/2} \geq \eta_\varepsilon(M)$, where*

$\alpha = \max_{j \in \mathbb{Z}_{2d}^\times} \sqrt{\sum_{i \in [n]} |\sigma_j(\mathbf{z}_i)|^2}$ *and* $\varepsilon \in (0, 1/2)$. *Let* $\tau : [d] \to \mathbb{Z}_{2d}^\times$ *be a bijection. Let* $Y = \langle \vec{\mathbf{z}}, \vec{\mathbf{v}} \rangle + \mathbf{e}$ *where* $\vec{\mathbf{v}} \sim D_{M+\vec{\mathbf{u}},\beta}$ *and* $\mathbf{e} \in K_{\mathbb{R}}$ *such that* $\mathbf{e} \sim D_\gamma$. *The statistical distance between the distribution of* Y *and the continuous Gaussian distribution* $D_{\mathsf{diag}(\vec{r})}$ *is at most* 2ε, *where* $\vec{r} \in \mathbb{R}^d$ *and* $r_j = \sqrt{\gamma^2 + \beta^2 \sum_{i \in [n]} |\sigma_{\tau(j)}(\mathbf{z}_i)|^2}$.

Using the smoothing parameter, we recall the lemma of [GKV10, Lemma 1] below and modify it for our purposes in this paper.

Lemma A.3 ([GKV10] Lemma 1, adapted)**.** *Let* $\varepsilon \in (0, 1/2)$ *and* $\vec{u}$ *be sampled from the continuous Gaussian distribution* $D_{\mathfrak{s}_1}$ *defined over* $\mathbb{R}^d$, *with* $\mathfrak{s}_1 > 0$. *If* $\vec{v}$ *is sampled from* $\vec{u} + D_{R-\vec{u},\mathfrak{s}_2}$, *with* $\mathfrak{s}_2 \geq \eta_\varepsilon(R)$, *then* $\vec{v}$ *is within statistical distance* 8ε *of* $D_{R,\mathfrak{s}}$, *where* $\mathfrak{s}^2 = \mathfrak{s}_1^2 + \mathfrak{s}_2^2$.

Another useful lemma that will help in reducing from the MLWE problem with continuous Gaussian to the MLWE problem with discrete Gaussian is the following lemma from [GPV08] about sub-lattices:

Lemma A.4 ([GPV08] Corollary 2.8)**.** *Let* Λ, Λ' *be* n*-dimensional lattices with* $\Lambda' \subset \Lambda$. *Then, for any* $\varepsilon \in (0, \frac{1}{2})$, $\mathfrak{s} \geq \eta_\varepsilon(\Lambda')$, $\vec{c} \in \mathbb{R}^n$, *the distribution of* $D_{\Lambda,\mathfrak{s},\vec{c}} \bmod \Lambda'$ *is within statistical distance at most* 2ε *of uniform over* Λ/Λ'.

We find that when $\Lambda = R^n$ and $\Lambda' = qR^n$, then Λ/Λ' is defined by $\Lambda \mod q$ and Lemma A.4 implies that a Gaussian sample from the lattice Λ when modded out by q results in a uniform element of R_q^n.

A.3 Additional Lattice-Based Assumptions

Informally, the decision variant of the Module Learning with Errors (MLWE) problem [LS15] asks an efficient adversary to distinguish between an oracle that returns noisy inner products and an oracle that returns elements sampled uniformly at random. We consider two versions of MLWE: (1) where the noise added to inner products is sampled from a continuous distribution defined over $K_{\mathbb{R}}$, and (2) where the noise is sampled from a discrete distribution defined over R. Note that for some families of discrete distributions that admit continuous counterparts, the hardness of (2) is implied by the hardness of (1) (cf. [Reg09, Lemma 4.3], [LPR13, Lemma 2.23], [GKV10, Lemma 1]).

Definition A.2 (MLWE)**.** *Let* n, m, q *be positive integers. Let* $\mathcal{D}$ *be a distribution over* R_q *and* ψ *be a distribution over* $K_{\mathbb{R}}$. *The Module Learning With Errors, denoted by* $\widehat{\mathsf{MLWE}}_{n,m,q,\psi}(\mathcal{D})$, *asks a* ppt *adversary* $\mathcal{A}$ *to distinguish* $(\mathbf{A}, q^{-1}\mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}} \bmod R)$ *from* $(\mathbf{A}, \vec{\mathbf{u}})$ *for any* $\mathbf{A} \hookleftarrow\$\ R_q^{m \times n}$, $\vec{\mathbf{s}} \leftarrow \mathcal{D}^n$, $\vec{\mathbf{e}} \leftarrow \psi^m$, *and* $\vec{\mathbf{u}} \hookleftarrow\$\ \mathbb{T}^m$. *We also consider the case where* $\vec{\mathbf{s}}$ *and* $\vec{\mathbf{e}}$ *are both sampled from the same distribution* χ *defined over* R *such that* $\mathcal{A}$ *has to distinguish between* $(\mathbf{A}, \mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}} \mod q)$ *and* $(\mathbf{A}, \vec{\mathbf{u}})$, *with* $\vec{\mathbf{u}} \hookleftarrow\$\ R_q^m$. *This case is denoted by* $\mathsf{MLWE}_{n,m,q,\chi}$, *and if* $\chi = D_{R,\mathfrak{s}}$ *with* $\mathfrak{s} > 0$, *we simply use* $\mathsf{MLWE}_{n,m,q,\mathfrak{s}}$.

We recall the definition of the Extended MLWE (ext-MLWE) problem in Definition A.3 that was introduced by Brakerski et al. [BLP+13] for the case $d = 1$ and generalized by Boudgoust et al. [BJRW23]. In brief, the ext-MLWE states that MLWE is still hard even if an adversary is allowed to obtain hints of the form $\langle \vec{\mathbf{e}}, \vec{\mathbf{z}} \rangle$ where $\vec{\mathbf{e}}$ is the error vector and $\vec{\mathbf{z}}$ is a vector chosen by the adversary. This assumption will be useful to prove the hardness of MLWE with secrets sampled from a distribution that works nicely with our protocols. Note that this version of ext-MLWE is different from the one used in some lattice-based zero-knowledge protocols [LNS21], [LNP22] where the adversary receives hints of the form $\mathsf{sign}(\langle \vec{\mathbf{e}}, \vec{\mathbf{z}} \rangle)$ where $\mathsf{sign}(a)$ returns 1 if $a \geq 0$ and 0 otherwise.

Definition A.3 (ext-MLWE)**.** *Let* $n, m, q, t \in \mathbb{N}_{>0}$. *Let* $\mathcal{Z} \subseteq R^m$, $\mathcal{D}$ *be a distribution over* R_q, *and* ψ *be a distribution over* $q^{-1}R$. *The Extended-MLWE, denoted by* $\widehat{\mathsf{ext\text{-}MLWE}^t}_{n,m,q,\psi,\mathcal{Z}}(\mathcal{D})$, *asks a* ppt *adversary* $\mathcal{A}$

to distinguish $(\mathbf{A}, q^{-1}\mathbf{AS}+\mathbf{E} \bmod R, \mathbf{E}^\top \vec{\mathbf{z}})$ *from* $(\mathbf{A}, \mathbf{U}, \mathbf{E}^\top \vec{\mathbf{z}})$ *where* $\vec{\mathbf{z}} \in \mathcal{Z}$ *is chosen by* $\mathcal{A}$, $\mathbf{A} \hookleftarrow\$ R_q^{m\times n}$, $\mathbf{S} \leftarrow \mathcal{D}^{n\times t}$, $\mathbf{E} \leftarrow \psi^{m\times t}$, *and* $\mathbf{U} \hookleftarrow\$ (q^{-1}R/R)^{m\times t}$. *Moving forward, we will omit the superscript* t *whenever* $t = 1$.

Lemma A.5 (Hardness of ext-MLWE [BJRW23], adapted)**.** *Let* $n, m, d, q \in \mathbb{N}_{>0}$ *such that* d *is a power of two and* q *is an unramified prime with inertia degree* $r > 0$ *in* R. *Let* $\mathcal{Z} \subseteq R^m$ *be a non-empty set such that for any* $\vec{\mathbf{x}} \in \mathcal{Z}$, *if* $\mathbf{x}_i \neq 0$ *then* $\mathbf{x}_i \bmod q \in R_q^\times$. *Let* $B = \max_{\vec{\mathbf{x}}\in\mathcal{Z}} \|\sigma(\vec{\mathbf{x}})\|_\infty$, $g = d/r$, $\varepsilon \in (0, 1/2)$, $\alpha \geq \eta_\varepsilon(q^{-1}R)$, *and* $\beta = \alpha\sqrt{4B^2+1}$. *Then, there is an efficient reduction from* $\widehat{\mathsf{MLWE}}_{n,m,q,D_\alpha}(U(R_q))$ *to* $\widehat{\mathsf{ext\text{-}MLWE}}_{n+1,m+1,q,D_{\frac{1}{q}R,\beta},\mathcal{Z}}(U(R_q))$ *with an additive loss upper bounded by* $\frac{g}{q^{nr}} + \frac{33\varepsilon}{2}$.

Lemma A.6 (Hardness of Hint-MLWE [KLSS23])**.** *Let* $n, m, q, \ell \in \mathbb{N}$ *and* $\mathcal{C}$ *be a distribution over* R. *Let* $B > 0$ *be a real number such that* $\sum_{j=0}^{\ell-1} \|\mathbf{c}_j\|_1^2 \leq B$ *for any* $(\mathbf{c}_0, ..., \mathbf{c}_{\ell-1})$ *sampled from* $\mathcal{C}$. *Let* $\varepsilon \in (0, 1/2]$. *For any* $\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s} > 0$ *such that* $1/\mathfrak{s} = (1/\mathfrak{s}_1^2 + B/\mathfrak{s}_2^2)$ *and* $\mathfrak{s} \geq \sqrt{2}\cdot\eta_\varepsilon(R)$, *there exists an efficient reduction from* $\mathsf{MLWE}_{n,m,q,\mathfrak{s}}$ *to* $\mathsf{Hint\text{-}MLWE}^{\ell,\mathfrak{s}_2,\mathcal{C}}_{n,m,q,\mathfrak{s}_1}$ *with an additive loss upper bounded by* $2\varepsilon(n+m)$.

A.4 Proof of Lemma 2.1

Proof of Lemma 2.1. By the Chinese Remainder Theorem, $R_q \cong (\mathbb{F}_{q^r})^{d/r}$. Note that the d/r copies of $\mathbb{F}_{q^r}$ are isomorphic but not necessarily identical. Since $\mathbf{A}$ is uniform over R_q, it follows that its residues over $\mathbb{F}_{q^r}$ are uniform and independent. We start with E_1. By looking at one of the copies of $\mathbb{F}_{q^r}$, it follows that E_1 happens if and only if the top left $n-k$ by $n-k$ submatrix of $\mathbf{A}$ has rank $n-k$ over $\mathbb{F}_{q^r}$. By a standard counting argument, this happens with probability $\alpha = \prod_{i=1}^{n-k}(1-q^{-ir})$, and by the independence across the d/r components, we have $\Pr[\mathsf{E}_1] = \alpha^{d/r}$. Focusing on α and using the fact that q is large so that $\log(1-q^{-ir}) \approx -q^{-ir}$, we have $\alpha = \exp(O(-q^{-r}))$. Using the Taylor series expansion of $\exp(\cdot)$ and the binomial expansion, we have $\Pr[\mathsf{E}_1] = 1 - O(\frac{d}{r}q^{-r})$.

We now analyze E_2. Conditioned on E_1, the first $n-k$ columns span a vector space of dimension $n-k$ over $\mathbb{F}_{q^r}$. The next k columns extend this to rank n with probability $\beta = \prod_{i=1}^{k}(1-q^{-ir})$. By using a series of arguments similar to the case of E_1, we obtain $\Pr[\mathsf{E}_2] = 1 - O(\frac{d}{r}q^{-r})$. □

B Additional Background on Proof Systems

Definition B.1 (Interactive Proof)**.** *An interactive proof for an NP relation* $\mathcal{R}$ *is a tuple of* ppt *algorithms* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$*: Given a security parameter* $\lambda \in \mathbb{N}$, *the* Setup *outputs some public parameters* pp, *which are also known as the common reference string.*
- $\langle \mathsf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}) \leftrightharpoons \mathsf{V}(\mathsf{pp}, \mathbb{x})\rangle \to (b; \mathsf{tr})$*: This is an interactive protocol between a* ppt *prover* P *and a* ppt *verifier* V. P *receives* $(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ *as input and* V *receives* $(\mathsf{pp}, \mathbb{x})$ *as input. At the end of the protocol* V *outputs a bit* $b \in \{0, 1\}$, *and* tr *is the transcript of the interaction.*

In addition, Π *is said to be public-coin if the random choices of* V *are included in* tr, *and it is said to be* $(2\mu+1)$*-round, with* $\mu \in \mathbb{N}$, *if* P *and* V *interact in a round-robin fashion where* P *sends the first and last messages over* $\mu+1$ *interactions.*

In this work, we focus on non-interactive zero-knowledge proofs obtained from interactive proofs. For this, it suffices that the underlying interactive proof be honest-verifier zero-knowledge, as this yields zero-knowledge in the ROM.

Definition B.2 (HVZK). *A public-coin interactive proof* $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ *for an NP relation* $\mathcal{R}$ *is said to be honest-verifier zero-knowledge (HVZK) if there exists a* ppt *simulator* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that for any* ppt *distinguisher* D*,* $\lambda \in \mathbb{N}$*, and* $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$*, the following distributions are indistinguishable:*

$$\left\{ (\mathsf{pp}, \mathsf{tr}) : \begin{array}{r} (\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Sim}_1(1^\lambda) \\ \mathsf{tr} \leftarrow \mathsf{Sim}_2(\mathsf{pp}, \mathbb{x}) \end{array} \right\} \overset{c}{\approx} \left\{ (\mathsf{pp}, \mathsf{tr}) : \begin{array}{r} \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (b; \mathsf{tr}) \leftarrow \langle \mathsf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}) \leftrightharpoons \mathsf{V}(\mathsf{pp}, \mathbb{x}) \rangle \end{array} \right\}$$

Definition B.3 (NIZK in ROM). *A non-interactive zero-knowledge proof in the ROM for an NP relation* $\mathcal{R}$ *is a tuple of* ppt *algorithms* $\Pi^{\mathcal{R}}_{\mathsf{NIZK}} = (\mathsf{Setup}, \mathsf{Prove}^H, \mathsf{Verify}^H)$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$*: Given a security parameter* $\lambda \in \mathbb{N}$ *, the* Setup *outputs some public parameters* pp*.*
- $\mathsf{Prove}^H(\mathsf{pp}, \mathbb{x}, \mathbb{w}) \to \pi$*: Given* pp*, a statement-witness pair* $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$*, the* Prove *algorithm, with access to a random oracle* H*, outputs a proof* π*.*
- $\mathsf{Verify}^H(\mathsf{crs}, \mathbb{x}, \pi) \to 1/0$*: Given* pp*, a statement* $\mathbb{x}$*, and a proof* π*. The deterministic* Verify *algorithm, with access to a random oracle* H*, outputs* 1 *if* π *is a valid proof with respect to* $\mathbb{x}$ *and* $\mathcal{R}$*. Otherwise, it outputs* 0*.*

In addition, $\Pi^{\mathcal{R}}_{\mathsf{NIZK}}$ *should satisfy the following properties:*

- ***Completeness****: For any* $\lambda \in \mathbb{N}$ *and* $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}$*, let* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $\pi \leftarrow \mathsf{Prove}^H(\mathsf{pp}, \mathbb{x}, \mathbb{w})$*. Then* $\Pr[\mathsf{Verify}^H(\mathsf{pp}, \mathbb{x}, \pi) = 1] \geq 1 - \mathsf{negl}(\lambda)$*.*
- ***Zero-knowledge****: There exists a* ppt *simulator* $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ *such that for any* ppt *distinguisher* D*, there exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*, we have* $\{\mathsf{pp} : \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)\} \overset{c}{\approx} \{\mathsf{pp} : (\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Sim}_1(1^\lambda)\}$ *and*

$$\left| \begin{array}{l} \Pr\left[\mathsf{D}^{\mathsf{Prove}^H(\mathsf{pp}, \cdot, \cdot)}(1^\lambda, \mathsf{pp}) = 1 | \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)\right] = 1 \\ \quad - \Pr\left[\mathsf{D}^{\mathsf{Sim}_2^H(\mathsf{pp}, \mathsf{st}, \cdot, \cdot)}(1^\lambda, \mathsf{pp}) \mid (\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{Sim}_1(1^\lambda)\right] = 1 \end{array} \right| \leq \mathsf{negl}(\lambda)$$

  where Sim_2 *can program* H*. In addition, it aborts if* $(\mathbb{x}, \mathbb{w}) \notin \mathcal{R}$*. Otherwise, it discards* $\mathbb{w}$ *and outputs* π *given only* $\mathsf{pp}, \mathsf{st}, \mathbb{x}$ *as inputs.*

Moving forward, to ease notation, we omit the superscript H in the description of $\Pi^{\mathcal{R}}_{\mathsf{NIZK}}$ and its algorithms.

An additional property that $\Pi^{\mathcal{R}}_{\mathsf{NIZK}}$ should satisfy is soundness, i.e., it should be hard for a ppt adversary to output a valid proof for a statement that is not in the language associated with $\mathcal{R}$. We review an adaptive form of knowledge soundness in Definition B.4. However, in this work, we consider a stronger notion called *weak simulation extractability* [FKMV12], which, informally, states that if an adversary is able to output a valid proof for some statement even when it has access to simulated proofs, then there exists an efficient algorithm, called an *extractor*, that, with black-box access to the adversary, can recover a valid witness from the adversary's proof in expected polynomial time.

Definition B.4 (Adaptive Knowledge Soundness for NIZKs [AFK22]). *A NIZK* $\Pi^{\mathcal{R}}_{\mathsf{NIZK}} = (\mathsf{Setup}, \mathsf{Prove}^H, \mathsf{Verify}^H)$ *in the ROM for an NP relation* $\mathcal{R}$ *is said to be adaptively knowledge sound with knowledge error* $\kappa : \mathbb{N} \times \mathbb{N} \to [0, 1]$ *if there exists an extractor* Ext *running in expected polynomial time such that for every* ppt *prover* P^* *making at most* Q *random oracle queries there is a polynomial* poly *such that for all* $\lambda \in \mathbb{N}$ *the following holds. Let* $\epsilon(\mathsf{P}^*)$ *denote the probability that* P^**, on input* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*, outputs a valid statement-proof pair* $(\mathbb{x}, \pi)$*. Then* Ext*, given black-box rewindable access to* P^**, satisfies*

$$\Pr[(\mathsf{Verify}^H(\mathsf{crs}, \mathbb{x}, \pi) = 1) \wedge (\mathbb{x}, \mathbb{w}) \in \mathcal{R} \mid \mathbb{w} \leftarrow \mathsf{Ext}^{\mathsf{P}^*}(\mathsf{crs}, \mathbb{x}, \pi)] \geq \frac{\epsilon(\mathsf{P}^*) - \kappa(\lambda, Q)}{\mathsf{poly}(\lambda)}$$

Remark B.1. In general, a public-coin $(2\mu+1)$-round interactive proof $\Pi = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for an NP relation $\mathcal{R}$, with $\mu \geq 1$, can be converted into a non-interactive proof in the ROM via the Fiat–Shamir transformation [FS87]. More specifically, V's i-th message during the i-th round, for $i \in [\mu]$, is replaced by a query to a random oracle H on a string containing the P's and V's messages from the first $i-1$ rounds with P's i-th message. In addition, if Π is HVZK and the first prover's message has high (pseudo-)min-entropy, then its Fiat–Shamir transformation is zero-knowledge, that is, a NIZK (see [GOP+25][Section 2.7] for a more detailed argument).

C Threshold Homomorphic Encryption

Definition C.1 (Threshold HE [MW16, BGG+18])**.** *Let $\mathcal{F} = \{f_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of functions where each $f_\lambda \in \mathcal{F}$ is a $k(\lambda)$-ary function with domain $(\mathcal{X}_\lambda)^{k(\lambda)}$. A threshold homomorphic encryption for $\mathcal{F}$ is a tuple of* ppt *algorithm $\Pi_{\mathsf{tHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{TDec}, \mathsf{Comb})$ with the following syntax:*

- $\mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t}) \to (\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathtt{n}\})$*: Given a security parameter $\lambda \in \mathbb{N}$, and threshold access parameters $(\mathtt{n}, \mathtt{t})$, with $\mathtt{n} \geq \mathtt{t}$ and $\mathtt{n} = \mathsf{poly}(\lambda)$,* KGen *outputs a public key* pk*, and a secret key share sk_i to each participant P_i.*
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ctx}$*: Given* pk *and $m \in \mathcal{X}_\lambda$,* Enc *outputs a ciphertext* ctx*.*
- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ctx}_1, \ldots, \mathsf{ctx}_k) \to \hat{\mathsf{ctx}}$*: Given* pk*, a function $f \in \mathcal{F}$ and ciphertexts $\mathsf{ctx}_1, \ldots, \mathsf{ctx}_k$,* Eval *outputs a ciphertext* $\hat{\mathsf{ctx}}$*.*
- $\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}) \to \mathsf{ds}_i$ *: Given a secret key share sk_i and a ciphertext* ctx*,* TDec *outputs a decryption share ds_i.*
- $\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}, D) \to \tilde{m}$ *: Given* pk*, a ciphertext* ctx*, and a set $D = \{\mathsf{ds}_i\}_{i\in\mathcal{S}}$, where $\mathcal{S} \subseteq [\mathtt{n}]$ such that $|\mathcal{S}| \geq t$, the deterministic* Comb *algorithm outputs a message $\tilde{m}$.*

In addition, Π_{tHE} should satisfy the following properties:

- ***Evaluation correctness and Compactness****: For any $\lambda \in \mathbb{N}$, $f \in \mathcal{F}$, $k = \mathsf{poly}(\lambda)$, $m_1, \ldots, m_k \in \mathcal{X}_\lambda$, $\mathtt{t} = \mathtt{t}(\lambda)$, $\mathtt{n} = \mathtt{n}(\lambda)$ with $\mathtt{n} \geq \mathtt{t}$, and $\mathcal{S} \subseteq [\mathtt{n}]$ with $|\mathcal{S}| \geq \mathtt{t}$, we have $|\hat{\mathsf{ctx}}| \leq \mathsf{poly}(\lambda)$ and $\Pr[\tilde{m} \neq f(m_1, \ldots, m_k)] \leq \mathsf{negl}(\lambda)$, where $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}) \leftarrow \mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t})$, $\hat{\mathsf{ctx}} \leftarrow \mathsf{Eval}(\mathsf{pk}, f, \{\mathsf{Enc}(\mathsf{pk}, m_i)\}_{i\in[k]})$, and $\tilde{m} \leftarrow \mathsf{Comb}(\mathsf{pk}, \hat{\mathsf{ctx}}, \{\mathsf{TDec}(\mathsf{sk}_i, \hat{\mathsf{ctx}})\}_{i\in\mathcal{S}})$. We say Π_{tHE} has perfect evaluation correctness if this probability is 0.*
- ***Semantic Security****: Π_{tHE} satisfies semantic security under static corruption if for any* ppt *adversary $\mathcal{A}$, there exists a negligible function* negl *such that for any λ, we have*

$$\Pr\left[\begin{array}{r} b \leftarrow\$ \{0,1\}; (\mathtt{n}, \mathtt{t}, \mathsf{st}_{\mathcal{A},1}) \leftarrow \mathcal{A}(1^\lambda); (\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathtt{n}}) \leftarrow \mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t}); \\ (\mathcal{S}, \mathcal{V}, \mathsf{st}_{\mathcal{A},2}) \leftarrow \mathcal{A}(\mathsf{st}_{\mathcal{A},1}, \mathsf{pk}); (m_0, m_1, \mathsf{st}_{\mathcal{A},3}) \leftarrow \mathcal{A}(\mathsf{st}_{\mathcal{A},2}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathcal{V}}); \\ \mathsf{ctx}_{m_b} \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b); b' \leftarrow \mathcal{A}(\mathsf{st}_{\mathcal{A},3}, \mathsf{ctx}_{m_b}) : \\ (b = b') \wedge \mathcal{S} \subseteq [\mathtt{n}] \wedge \mathcal{V} \subset \mathcal{S} \wedge |\mathcal{V}| < \mathtt{t} \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

- ***Simulation Security****: Π_{tHE} satisfies simulation security under static corruption if there exists a stateful* ppt *simulator $\mathsf{Sim} = (\mathsf{SimKGen}, \mathsf{SimTDec})$ such that for any* ppt *adversary, there exists a negligible function such that for any $\lambda \in \mathbb{N}$, $\Pr\left[\mathsf{Sim\text{-}Exp}^{\mathcal{A},\mathsf{Sim}}(1^\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$, where* Sim-Exp *is defined in Fig. 3.*

In addition, given $\vec{m} \in (\mathcal{X}_\lambda)^k$, we will use $\vec{\mathsf{ctx}}_{\vec{m}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \vec{m})$ to denote the vector obtained by individually encrypting each component of $\vec{m}$.

We review a construction of threshold homomorphic encryption in Appendix G.1.

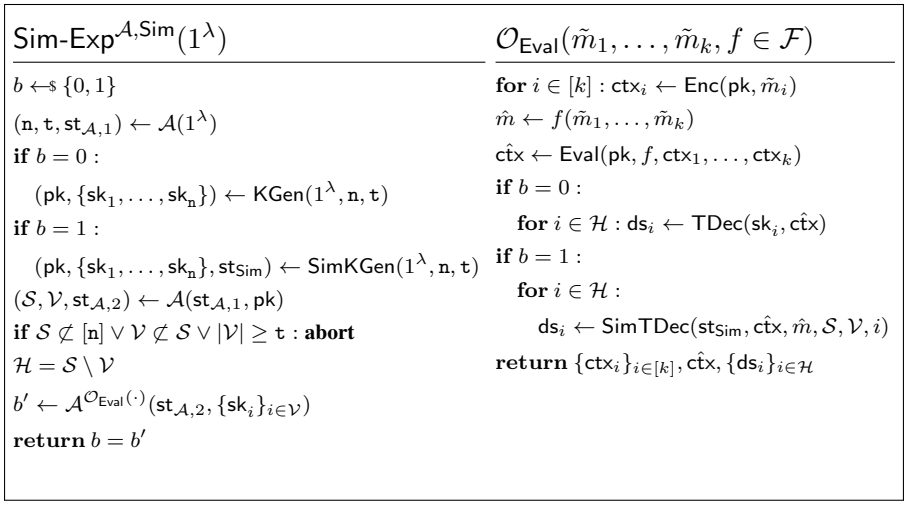

Figure 3: Threshold HE Simulation Security Experiment

D Proofs of Section 5

D.1 Proof of Theorem 5.2

Proof of Theorem 5.2. Let D be a ppt distinguisher that performs Q_H random oracle queries and Q_P proof queries. We now proceed via a sequence of hybrids:

- Hyb 0: D interacts with an oracle that runs the Prove algorithm of Construction 1 on each proof query and answers to random oracle queries via lazy sampling. Let E_0 be the event that D outputs 1 in this hybrid.

- Hyb 1: Similar to Hyb 0, except during Setup, instead of sampling $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow\$ R_q^{(n+4)\times m}$, sample $\mathbf{V} \leftarrow\$ \mathrm{GL}_{n+4}(R_q)$ and $\mathbf{A} \leftarrow\$ R_q^{(n+4)\times(m-n-4)}$, and then, compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{n+4} \mid \mathbf{A}]$ and keep $\mathbf{V}$ in memory. Let E_1 be the event that D outputs 1 in this hybrid. Unless the first $n+4$ columns of $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow\$ R_q^{(n+4)\times m}$ are dependent, Hyb 0 and Hyb 1 are identical. Therefore, by Lemma 2.1, $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq O(hq^{-d/h}) \leq \mathsf{negl}(\lambda)$.

- Hyb 2: Similar to Hyb 1, except during the execution of a proof query on a valid witness $\mathbb{w} := (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3) \in R_q^3$, the oracle samples $\mathbf{c} \leftarrow \mathcal{C}$ after sampling $\vec{\mathbf{r}}$. Then, when it is time to query H, it checks if $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) = \perp$. If so, it programs $H([\vec{\mathbf{t}} \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) \leftarrow \mathbf{c}$. Otherwise, it aborts. Let E_2 be the event that D outputs 1 in this hybrid.

  Clearly, Hyb 2 is identical to Hyb 1 unless the oracle aborts, and this happens only when $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v})$ was previously programmed by either a random oracle or proof query. Let coll denote this event. For this to happen, we must have $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top = \vec{\mathbf{t}'}$, where $\vec{\mathbf{t}'} \in R_q^{n+4}$ is a value that appeared in one of those earlier queries. For the i-th proof query, there are at most $Q_H + i$ instances where the random oracle could have been programmed with respect to $\vec{\mathbf{t}'}$, and since D performs at most Q_P proof queries, it follows, by a succession of union bounds, that $\Pr[\mathsf{coll}] \leq Q_P(Q_H+Q_P)\cdot\Pr[[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top = \vec{\mathbf{t}'}]$. Given that

$$[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top = \mathbf{V} \cdot [\mathbf{I}_{n+4} \mid \mathbf{A}]\vec{\mathbf{r}} + [\mathbf{0}^n \mid \mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3 \mid \mathbf{m}_4]^\top$$

  where $\mathbf{m}_4 = \langle \vec{\mathbf{b}}_3, \vec{\mathbf{y}} \rangle - \mathbf{m}_2\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle - \mathbf{m}_1\langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}} \rangle$, it follows that $[\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top = \vec{\mathbf{t}'}$ if and only if

$$\vec{\mathbf{r}}_0 = \mathbf{V}^{-1} \cdot (\vec{\mathbf{t}'} - [\mathbf{0}^n \mid \mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3 \mid \mathbf{m}_4]^\top) - \mathbf{A}\vec{\mathbf{r}}_1$$

where $[\vec{\mathbf{r}}_0^\top \mid \vec{\mathbf{r}}_1^\top]^\top := \vec{\mathbf{r}} \leftarrow D_{R,\mathfrak{s}_1}^m$ and $\vec{\mathbf{r}}_1 \in R_q^{m-n-4}$ and $\vec{\mathbf{r}}_0 \in R_q^{n+4}$. Therefore, by Lemma 2.3, we have $\Pr[\mathsf{Coll}] \leq Q_P(Q_H + Q_P) \cdot 2^{-d(n+4)} \leq \mathsf{negl}(\lambda)$.

Thus, we conclude that $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \Pr[\mathsf{Coll}] \leq \mathsf{negl}(\lambda)$.

- Hyb 3: Similar to Hyb 2, except during the execution of a proof query on a valid witness, the oracle samples $\vec{\mathbf{z}} \leftarrow D_{R,\mathfrak{s}_2}^m + \mathbf{c}\vec{\mathbf{r}}$ after sampling $\vec{\mathbf{r}}$ and $\mathbf{c}$. Given $\vec{\mathbf{z}}$, $\mathbf{c}$, $\vec{\mathbf{t}}$ and $\mathbf{t}_4$, the vectors $\vec{\mathbf{w}}$ and $\mathbf{v}$ are fully determined. Thus, the oracle derives these two values in the same way as the simulator. Let E_3 be the event that D outputs 1 in this hybrid. Clearly, these changes are syntactic, and it follows that $\Pr[\mathsf{E}_2] = \Pr[\mathsf{E}_3]$.

- Hyb 4: Similar to Hyb 3, except during the execution of a proof query, the oracle runs SimProve with the witness, i.e., at Step 3 of SimProve, it computes $[\vec{\mathbf{t}} \mid \mathbf{t}_4]^\top \leftarrow \mathbf{V} \cdot \vec{\mathbf{u}} + [\mathbf{0}^n \mid \mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3 \mid \mathbf{m}_4^*]^\top$, where $\vec{\mathbf{u}} \leftarrow\!\!\$\ R_q^{n+4}$, and $\mathbf{m}_4^* \leftarrow \langle \vec{\mathbf{b}}_3, \vec{\mathbf{z}} \rangle - \mathbf{c}\mathbf{u}'_3 - \mathbf{m}_2(\langle \vec{\mathbf{b}}_1, \vec{\mathbf{z}} \rangle - \mathbf{c}\mathbf{u}'_1) - \mathbf{m}_1(\langle \vec{\mathbf{b}}_2, \vec{\mathbf{z}} \rangle - \mathbf{c}\mathbf{u}'_2)$, with $[(\vec{\mathbf{u}}'_0)^\top \mid \mathbf{u}'_1 \mid \mathbf{u}'_2 \mid \mathbf{u}'_3 \mid \mathbf{u}'_4]^\top := \mathbf{V}\vec{\mathbf{u}}$. Let E_4 be the event that D outputs 1 in this hybrid.

  Claim 1. *Under the hardness of* Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-n-4,n+4,q,\mathfrak{s}_1}$, *it follows that* $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \leq \mathsf{negl}(\lambda)$.

- Hyb 5: Similar to Hyb 4, except during the execution of a proof query, the oracle discards the witness and runs SimProve. Let E_5 be the event that D outputs 1 in this hybrid. Since $\vec{\mathbf{u}} \leftarrow\!\!\$\ R_q^{n+4}$ and $\mathbf{V} \in \mathrm{GL}_{n+4}(R_q)$, it follows that $\vec{\mathbf{t}}$ and $\mathbf{t}_4$ are uniformly random in both Hyb 4 and Hyb 5. Hence, $\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_5]$

Therefore, $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_5]| \leq \mathsf{negl}(\lambda)$. This completes the proof.

Proof of Claim 1. We proceed by contraposition. Suppose $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| > \delta$, where δ is non-negligible. We use D to build a ppt adversary $\mathcal{A}$ that breaks the Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-n-4,n+4,q,\mathfrak{s}_1}$ with an advantage of at least δ. $\mathcal{A}$ plays the role of D's oracle and upon receiving an instance $(\mathbf{A}, \mathbf{U}, (\mathbf{c}_i, \vec{\mathbf{z}}_i)_{i \in [Q_P]})$, where $[\vec{\mathbf{u}}_1 \mid \cdots \mid \vec{\mathbf{u}}_{Q_P}] := \mathbf{U} \in R_q^{(n+4) \times Q_P}$, $\mathcal{A}$ proceeds as follows:

- Sample $\mathbf{V} \leftarrow\!\!\$\ \mathrm{GL}_{n+4}(R_q)$ and set $\mathsf{pp} := \mathbf{V} \cdot [\mathbf{I}_{n+4} \mid \mathbf{A}]$.
- **Proof Query**: Upon receiving the i-th proof query on a valid witness $\mathbb{w}_i := (\mathbf{m}_{i,1}, \mathbf{m}_{i,2}, \mathbf{m}_{i,3})$, use $(\vec{\mathbf{u}}_i, \mathbf{c}_i, \vec{\mathbf{z}}_i)$ to answer as follows: (1) parse $[(\vec{\mathbf{u}}'_{i,0})^\top \mid \mathbf{u}'_{i,1} \mid \mathbf{u}'_{i,2} \mid \mathbf{u}'_{i,3} \mid \mathbf{u}'_{i,4}]^\top := \mathbf{V}\vec{\mathbf{u}}_i$, (2) compute $\vec{\mathbf{t}}_i \leftarrow [(\vec{\mathbf{u}}'_{i,0})^\top \mid \mathbf{u}'_{i,1} \mid \mathbf{u}'_{i,2} \mid \mathbf{u}'_{i,3}]^\top + [\mathbf{0}^n \mid \mathbf{m}_{i,1} \mid \mathbf{m}_{i,2} \mid \mathbf{m}_{i,3}]^\top$, and $\mathbf{t}_{i,4} \leftarrow \mathbf{u}'_{i,4} + \langle \vec{\mathbf{b}}_3, \vec{\mathbf{z}}_i \rangle - \mathbf{c}_i\mathbf{u}'_{i,3} - \mathbf{m}_{i,2}(\langle \vec{\mathbf{b}}_1, \vec{\mathbf{z}}_i \rangle - \mathbf{c}_i\mathbf{u}'_{i,1}) - \mathbf{m}_{i,1}(\langle \vec{\mathbf{b}}_2, \vec{\mathbf{z}}_i \rangle - \mathbf{c}_i\mathbf{u}'_{i,2})$, (3) recover $\vec{\mathbf{w}}_i$ and $\mathbf{v}_i$ from $\vec{\mathbf{z}}_i$, $\mathbf{c}_i$, $\vec{\mathbf{t}}_i$, and $\mathbf{t}_{i,4}$ and (4) program $H([\vec{\mathbf{t}}_i^\top \mid \mathbf{t}_{i,4}]^\top, \vec{\mathbf{w}}_i, \mathbf{v}_i) \leftarrow \mathbf{c}_i$.
- **Random Oracle Query**: Upon receiving a random oracle query $\mathsf{qry} \in R_q^{n+4} \times R_q^n \times R_q$, if $H(\mathsf{qry}) = \perp$, sample $\mathbf{c} \leftarrow \mathcal{C}$, and then, set $H(\mathsf{qry}) \leftarrow \mathbf{c}$. Return $H(\mathsf{qry})$.
- Finally, $\mathcal{A}$ outputs whatever D outputs.

Since D runs in poly-time, $\mathcal{A}$ also runs in poly-time. In addition, if $\mathbf{U} = [\mathbf{I}_{n+4} \mid \mathbf{A}] \cdot \mathbf{R}$ where $\mathbf{R} = [\vec{\mathbf{r}}_1 \mid \cdots \mid \vec{\mathbf{r}}_{Q_P}]$ with $\vec{\mathbf{r}}_i \leftarrow D_{R,\mathfrak{s}_1}^m$, $\mathcal{A}$ perfectly simulates Hyb 3, and if $\mathbf{U} \leftarrow\!\!\$\ R_q^{(n+4) \times Q_P}$, it perfectly simulates Hyb 4. This completes the proof of the claim. □

□

D.2 Predicate Special Soundness of ALS

We now prove that Construction 1 satisfies predicate-special-soundness. Before doing so, we recall that it is a 3-round protocol (see Remark B.1). Assuming that the difference between two challenges sampled from $\mathcal{C}$ is invertible with overwhelming probability, we require a 3-tree of transcripts to extract a weak binding opening (see Definition 2.4).

Lemma D.1. *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_j)$ for any $j \in [h]$ (see Lemma 2.6). Consider the following predicate system $\mathfrak{C}$:*

- ***Challenge predicates**:*
  - $\mathfrak{C}^{\mathsf{chal}}_{1,1}(\mathbf{c}_1) = 1$ *for any* $\mathbf{c}_1 \leftarrow\!\$\, \mathcal{C}$.
  - $\mathfrak{C}^{\mathsf{chal}}_{1,i}(\mathbf{c}_1, \ldots, \mathbf{c}_i) = 1$ *if and only if* $\mathbf{c}_1 - \mathbf{c}_i \in R_q^\times$ *for any* $i \in \{2, 3\}$.
- ***Commitment predicate**:*
  - $\mathfrak{C}^{\mathsf{prop}}_1(t_1, t_2) = 1$, *with* $t_i \in \mathscr{T}_2$, *if and only if the extracted witness* $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$, *extracted from* t_1 *and* t_2, *is a weak opening (see Definition 2.4) and, parsing* $[\mathbf{m}_1^* | \ldots | \mathbf{m}_4^*]^\top := \vec{\mathbf{m}}^*$, *we have* $\mathbf{m}_1^* \mathbf{m}_2^* = \mathbf{m}_3^*$.
  - $\mathfrak{C}^{\mathsf{bind}}_1(t_1, t_2, t_3) = 1$, *with* $t_i \in \mathscr{T}_2$, *if and only if, given two extracted witnesses* $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$, *extracted from* t_1 *and* t_2, *and* $(\bar{\mathbf{c}}', \vec{\mathbf{r}}^{*\prime}, \vec{\mathbf{m}}^{*\prime})$, *extracted from* t_1 *and* t_3, *we have* $(\vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*) = (\vec{\mathbf{r}}^{*\prime}, \vec{\mathbf{m}}^{*\prime})$.

$\mathfrak{C}^{\mathsf{chal}}_{1,1}$ has failure density $p^{\mathsf{chal}}_{1,1} = 0$, $\mathfrak{C}^{\mathsf{chal}}_{1,i}$ has failure density $p^{\mathsf{chal}}_{1,i} = hM$, and $(\mathfrak{C}^{\mathsf{prop}}_1, \mathfrak{C}^{\mathsf{bind}}_1)$ has failure density $p^{\mathsf{com}}_1 = 2M$. Furthermore, under the hardness of $\mathsf{MSIS}_{n,m,q,8d\beta}$, the interactive version of Construction 1 is $(3, \mathfrak{C})$-predicate-special-sound.

Proof. First, we analyze the failure density of each predicate. For $\mathfrak{C}^{\mathsf{chal}}_{1,1}$, since it always outputs 1, we have $p^{\mathsf{chal}}_{1,1} = 0$. For $\mathfrak{C}^{\mathsf{chal}}_{1,i}$, with $2 \leq i \leq 3$, the quantity $p^{\mathsf{chal}}_{1,i}$ represents an upper bound on the fraction of challenges $\mathbf{c}_i \leftarrow\!\$\, \mathcal{C} \setminus \{\mathbf{c}_1, \ldots, \mathbf{c}_{i-1}\}$ that collide with $\mathbf{c}_1$ over any prime ideal dividing $(q, X^d + 1)$, relative to the total challenge space $\mathcal{C}$ (see Definition 3.4). Since sampling without replacement can only reduce this fraction, we can set $p^{\mathsf{chal}}_{1,i}$ to be the fraction of challenges sampled uniformly at random from $\mathcal{C}$ that collide with $\mathbf{c}_1$, which is exactly the probability that given a uniformly random $\mathbf{c}_i \leftarrow\!\$\, \mathcal{C}$ there exist $j \in [h]$ such that $\mathbf{c}_i - \mathbf{c}_1 = 0 \bmod (X^{d/h} - \zeta_j)$. By Lemma 2.6, it follows that $p^{\mathsf{chal}}_{1,i} = hM$.

Now consider the commitment predicate $(\mathfrak{C}^{\mathsf{prop}}_1, \mathfrak{C}^{\mathsf{bind}}_1)$. Assuming that the BDLOP commitment is binding, the quantity p^{com}_1 upper bounds on the fraction of challenges $\mathbf{c}_3 \leftarrow\!\$\, \mathcal{C} \setminus \{\mathbf{c}_1, \mathbf{c}_2\}$ for which the extracted $\bar{\mathbb{w}}^* := (\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$ (see Appendix A.2 for a sketch of the extraction procedure), with $[\mathbf{m}_1^* | \ldots | \mathbf{m}_4^*]^\top := \vec{\mathbf{m}}^*$, is either not a weak opening (see Definition 2.4) or satisfies $\mathbf{m}_1^* \mathbf{m}_2^* \neq \mathbf{m}_3^*$, relative to the total challenge space $\mathcal{C}$ (see Definition 3.5). Without loss of generality, since all three transcripts in the tree are valid, $\bar{\mathbb{w}}^*$ is necessarily a weak opening. It therefore suffices to bound the fraction of challenges for which $\mathbf{m}_1^* \mathbf{m}_2^* \neq \mathbf{m}_3^*$. As before, since we are upper bounding, we consider the fraction of $\mathbf{c}_3 \leftarrow\!\$\, \mathcal{C}$ for which we have $\mathbf{m}_1^* \mathbf{m}_2^* \neq \mathbf{m}_3^*$ while still producing accepting transcripts. Since $\bar{\mathbb{w}}^*$ is a weak opening and given the commitment $[\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top := [\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top$ that is shared by all transcripts, we have $\mathbf{B}_1 \vec{\mathbf{r}}^* + \vec{\mathbf{m}}^* = [\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$, where $\mathbf{B}_1$ is the matrix in the pp. Let $\vec{\mathbf{y}}^* \in R_q^m$ be the unique value computed from all challenge-third-message pair $(\mathbf{c}_j, \vec{\mathbf{z}}_j)$ in the j-th transcript, for $1 \leq j \leq 3$, such that $\vec{\mathbf{z}}_j = \vec{\mathbf{y}}^* + \mathbf{c}_j \vec{\mathbf{r}}^*$. From the verification procedure (see Step 5.1) on the third transcript, we have

$$[\mathbf{f}_{1,3} \mid \cdots \mid \mathbf{f}_{4,3}]^\top = \mathbf{B}_1 \vec{\mathbf{z}}_3 - \mathbf{c}_3 [\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$$

$$= \mathbf{B}_1\vec{\mathbf{y}}^* - \mathbf{c}_3[\mathbf{m}_1^*|\dots|\mathbf{m}_4^*]^\top \tag{2}$$

$$\text{and } \mathbf{v} = \mathbf{f}_{1,3}\mathbf{f}_{2,3} + \mathbf{c}_3\mathbf{f}_{3,3} + \mathbf{f}_{4,3} \tag{3}$$

From Eqs. (2) and (3) and by parsing $[\vec{\mathbf{b}_1} \mid \cdots \mid \vec{\mathbf{b}_4}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}_i} \in R_q^m$, we have

$$\begin{aligned}&\langle\vec{\mathbf{b}_1}, \vec{\mathbf{y}}^*\rangle\langle\vec{\mathbf{b}_2}, \vec{\mathbf{y}}^*\rangle + \langle\vec{\mathbf{b}_4}, \vec{\mathbf{y}}^*\rangle - \mathbf{v}\\ &+ \mathbf{c}_3\left(\langle\vec{\mathbf{b}_3}, \vec{\mathbf{y}}^*\rangle - \mathbf{m}_2^*\langle\vec{\mathbf{b}_1}, \vec{\mathbf{y}}^*\rangle - \mathbf{m}_1^*\langle\vec{\mathbf{b}_2}, \vec{\mathbf{y}}^*\rangle - \mathbf{m}_4^*\right)\\ &- \mathbf{c}_3^2(\mathbf{m}_3^* - \mathbf{m}_1^*\mathbf{m}_2^*) = 0\end{aligned} \tag{4}$$

Since $\mathbf{m}_1^*\mathbf{m}_2^* \neq \mathbf{m}_3^*$, there exists a prime ideal $(q, X^{d/h} - \zeta_k)$, with $k \in [h]$, over which the left-hand side of Equation 4 is a non-zero polynomial of degree at most 2 in $\mathbf{c}_3$. Since $\mathbb{Z}_q[X]/(X^{d/h} - \zeta_k)$ is a field, the left-hand side of Equation 4 has at most two distinct roots. By Lemma 2.6, the probability that $\mathbf{c}_3 \leftarrow\!\!\$\ \mathcal{C}$ collides with one of those roots is at most $2M$. Therefore, $p_1^{\mathsf{com}} = 2M$.

To compute p_1^{com}, we required $\mathfrak{C}_1^{\mathsf{bind}}$ to always output 1 (as required by Definition 3.5), which in turn implies that the BDLOP commitment is binding with respect to weak openings. If we extract two weak openings $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \vec{\mathbf{m}}^*)$ and $(\bar{\mathbf{c}}', \vec{\mathbf{r}^*}', \vec{\mathbf{m}^*}')$ such that either $\vec{\mathbf{r}}^* \neq \vec{\mathbf{r}^*}'$ or $\vec{\mathbf{m}}^* \neq \vec{\mathbf{m}^*}'$, which implies that $\mathfrak{C}_1^{\mathsf{bind}}$ outputted 0, then by Lemma 2.7, one can recover an $\mathsf{MSIS}_{n,m,q,8d\beta}$ solution.

Finally, the predicate system $\mathfrak{C}$ contains all the necessary predicates that need to be satisfied to successfully extract a witness. Therefore, the interactive version of Construction 1 is $(3, \mathfrak{C})$-predicate-special-sound. □

Remark D.1 (Reducing Soundness Error). We note that it is possible to reduce the knowledge-soundness error of Construction 1 by a factor of $2\left(\frac{h+1}{h+2}\right)$ by adopting a predicate system similar to that of Aardal et al. [AAB+24, Appendix I], as follows: For the challenge predicates, we set $\mathfrak{C}_{1,3}^{\mathsf{chal}}$ to always be 1, i.e., we do not check if $\mathbf{c}_1 - \mathbf{c}_3 \in R_q^\times$, and for the binding component of the commitment predicate, $\mathfrak{C}_1^{\mathsf{bind}}(t_1, t_2, t_3)$, we check if $\vec{\mathbf{z}}_1 - \mathbf{c}_1\vec{\mathbf{r}} = \vec{\mathbf{z}}_2 - \mathbf{c}_2\vec{\mathbf{r}} = \vec{\mathbf{z}}_3 - \mathbf{c}_3\vec{\mathbf{r}}$, where $\vec{\mathbf{r}}$ is the witness extracted from t_1 and t_2.

D.3 Proof of Theorem 5.5

Remark D.2 (Linearized form of verification). A simple fact that will be useful later in the thresholdization of Construction 1 is that Steps 3 and 4 of the verification procedure can be expressed as an equality test involving a linear equation. Indeed, letting $\vec{\mathbf{f}} = \mathbf{B}_1\vec{\mathbf{z}} - \mathbf{c}[\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$, it follows that $\mathbf{v} = \langle\vec{\mathbf{e}}_2, \vec{\mathbf{f}} \otimes \vec{\mathbf{f}}\rangle + \mathbf{c} \cdot \langle\vec{\mathbf{e}}_3, \vec{\mathbf{f}}\rangle + \langle\vec{\mathbf{e}}_4, \vec{\mathbf{f}}\rangle$, where $\vec{\mathbf{e}}_2$ is the second standard basis vector in R_q^{16}, and $\vec{\mathbf{e}}_3$ and $\vec{\mathbf{e}}_4$ are the third and fourth standard basis vectors in R_q^4, respectively. The advantage of this representation is that, for any shift $\vec{\mathbf{d}} \in R_q^4$, $\mathbf{v} + \Delta_{\mathbf{v}} = \langle\vec{\mathbf{e}}_2, (\vec{\mathbf{f}} + \vec{\mathbf{d}}) \otimes (\vec{\mathbf{f}} + \vec{\mathbf{d}})\rangle + \mathbf{c} \cdot \langle\vec{\mathbf{e}}_3, (\vec{\mathbf{f}} + \vec{\mathbf{d}})\rangle + \langle\vec{\mathbf{e}}_4, (\vec{\mathbf{f}} + \vec{\mathbf{d}})\rangle$, with $\Delta_{\mathbf{v}} = \langle\vec{\mathbf{e}}_2, \vec{\mathbf{f}} \otimes \vec{\mathbf{d}} + \vec{\mathbf{d}} \otimes (\vec{\mathbf{f}} + \vec{\mathbf{d}})\rangle + \mathbf{c} \cdot \langle\vec{\mathbf{e}}_3, \vec{\mathbf{d}}\rangle + \langle\vec{\mathbf{e}}_4, \vec{\mathbf{d}}\rangle$.

Proof of Theorem 5.5. Let $\mathsf{Sim}_{\mathsf{tHE}} = (\mathsf{SimKGen}, \mathsf{SimTDec})$ be the simulator for Π_{tHE} defined by simulation security which we described in Section 2.6 and $\mathsf{Sim}_{\mathsf{vALS}} = (\mathsf{SimSetup}, \mathsf{SimProve}, \mathsf{SimRO})$ be the simulator for Construction 1 (see Simulator 1). Now, we are ready to describe the simulator.

Simulator 2: Simulator of threshold vALS

The simulator initializes the random tape of the adversary with a uniformly random string. In addition, it tracks and answers random oracle queries.

- $\mathsf{SimSetup}(1^\lambda, \mathtt{n}, \mathtt{t})$: Given $\lambda \in \mathbb{N}$, proceed as follows:
  1. Compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathsf{Sim}_{\mathsf{vALS}}.\mathsf{SimSetup}(1^\lambda)$.
  2. Compute $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}, \mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}) \leftarrow \mathsf{Sim}_{\mathsf{tHE}}.\mathsf{SimKGen}(1^\lambda, \mathtt{n}, \mathtt{t})$.
  3. Initialize an empty state $\mathsf{st}_{\mathsf{Sim}}$ and store $\mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}$.

4. Return $\mathsf{pp} = (\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1)$ and $\mathsf{paux}_i = \mathsf{sk}_i$ to each prover P_i

- $\mathsf{SimWitShare}(\mathsf{pp})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, do the following:
  1. For $i \in [3]$, compute $\mathsf{ctx}_{\mathbf{m}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$ and $\{[\![\mathbf{m}_i]\!]_j\}_{j\in[\mathtt{n}]} \leftarrow \Pi_{\mathsf{SS}}.\mathsf{Share}(\mathbf{0}, (\mathtt{n}, \mathtt{t}))$.
  2. Set $\mathsf{aux}_w := (\mathsf{ctx}_{\mathbf{m}_1}, \mathsf{ctx}_{\mathbf{m}_2}, \mathsf{ctx}_{\mathbf{m}_3})$.
  3. For $i \in [\mathtt{n}]$, set $\mathsf{pwit}_i := ([\![\mathbf{m}_1]\!]_i, [\![\mathbf{m}_2]\!]_i, [\![\mathbf{m}_3]\!]_i)$ and send $(\mathsf{pwit}_i, \mathsf{aux}_w)$ to prover P_i.

- $\mathsf{SimCom}(\mathsf{pp}, \mathcal{S}, \mathcal{V}, \{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{V}}, \mathsf{aux}_w, \mathsf{st}_{\mathsf{Sim}})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, the total set of participating provers $\mathcal{S}$, a set of corrupted parties $\mathcal{V}$ with their initial inputs $(\{(\mathsf{paux}_i, \mathsf{pwit}_i)\}_{i\in\mathcal{V}}, \mathsf{aux}_w)$, and a simulator state $\mathsf{st}_{\mathsf{Sim}}$, proceed as follows:
  1. If $\mathcal{S} \not\subseteq [\mathtt{n}]$ or $\mathcal{V} \not\subset \mathcal{S}$ or $|\mathcal{V}| \geq \mathtt{t}$: abort.
  2. Let $\mathcal{H} \leftarrow \mathcal{S} \setminus \mathcal{V}$. (*this is the set of honest provers*)
  3. Run $\mathsf{Sim}_{\mathsf{vALS}}.\mathsf{SimProve}(\mathbf{B}_0, \mathbf{B}_1, \mathbb{x}; \mathfrak{s}_1\sqrt{|\mathcal{H}|}, \mathfrak{s}_2\sqrt{|\mathcal{H}|})$ with parameters corresponding to a proof generated by provers in $\mathcal{H}$ to obtain $(\vec{\mathbf{t}}' := [(\vec{\mathbf{t}}'_0)^\top \mid \mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3]^\top, \pi := (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}, \vec{\mathbf{z}}'))$, i.e., during the execution of $\mathsf{Sim}_{\mathsf{vALS}}.\mathsf{SimProve}$, use $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to sample $\vec{\mathbf{z}}'$.
  4. Let $j \leftarrow\$ \mathcal{H}$. (*this will be the last honest prover to reply*).
  5. ▶ Upon receiving $\{[\vec{\mathbf{t}}^\top_{0,k} \mid \mathbf{t}_{1,k} \mid \mathbf{t}_{2,k} \mid \mathbf{t}_{3,k}]^\top\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
     (a) For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $[\vec{\mathbf{t}}^\top_{0,i} \mid \mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]^\top \leftarrow\$ R_q^{n+3}$ and return that as its output.
     (b) For P_j, compute $\vec{\mathbf{t}}_{0,j} \leftarrow \vec{\mathbf{t}}'_0 - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{t}}_{0,i}$ and $[\mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}] \leftarrow [\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] - \sum_{i\in\mathcal{S}\setminus\{j\}}[\mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]$. Then, return $[\vec{\mathbf{t}}^\top_{0,j} \mid \mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}]^\top$ as its output.
  6. Compute $\vec{\mathbf{t}} \leftarrow [(\vec{\mathbf{t}}'_0 + \sum_{i\in\mathcal{V}} \vec{\mathbf{t}}_{0,i})^\top \mid \mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3]^\top$.
  7. Store $(\mathcal{S}, \mathcal{V}, \vec{\mathbf{t}}, j, (\vec{\mathbf{t}}', \pi := (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}, \vec{\mathbf{z}}')))$ in $\mathsf{st}_{\mathsf{Sim}}$.

- $\mathsf{SimProve}(\mathsf{pp}, \mathcal{S}, \mathcal{V}, \vec{\mathbf{t}}, \mathsf{st}_{\mathsf{Sim}})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, the total set of participating provers $\mathcal{S}$, a set of corrupted provers $\mathcal{V}$, a commitment $\vec{\mathbf{t}}$, and a simulator state $\mathsf{st}_{\mathsf{Sim}}$, proceed as follows:
  1. Use $(\mathcal{S}, \mathcal{V}, \vec{\mathbf{t}})$ to recover the tuple $(\mathcal{S}, \mathcal{V}, \vec{\mathbf{t}}, j, (\vec{\mathbf{t}}', \pi := (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}, \vec{\mathbf{z}}')))$ from $\mathsf{st}_{\mathsf{Sim}}$. If no such tuple exists, abort.
  2. Parse $[\vec{\mathbf{t}}_0 \mid \mathbf{t}_1 \mid \mathbf{t}_2 \mid \mathbf{t}_3] := \vec{\mathbf{t}}$ and $[\vec{\mathbf{t}}'_0 \mid \mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] := \vec{\mathbf{t}}'$. *(Note that* $[\mathbf{t}_1 \mid \mathbf{t}_2 \mid \mathbf{t}_3] = [\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3]$ *by construction.)*
  3. For $k \in \mathcal{V}$, use the adversary's random tape to sample $\vec{\mathbf{r}}_k \leftarrow D^m_{R,\mathfrak{s}_1}$, and $\vec{\mathbf{y}}_k \leftarrow D^m_{R,\mathfrak{s}_2}$. Then, compute $\vec{\mathbf{w}}_{\mathcal{V}} \leftarrow \mathbf{B}_0 \cdot \sum_{k\in\mathcal{V}} \vec{\mathbf{y}}_k$ and $\vec{\mathbf{z}}_{\mathcal{V}} \leftarrow \sum_{k\in\mathcal{V}} \vec{\mathbf{y}}_k + \mathbf{c} \cdot \vec{\mathbf{r}}_k$. *(The contribution of the adversary).*
  4. Let $\vec{\mathbf{f}} \leftarrow \mathbf{B}_1\vec{\mathbf{z}}' - \mathbf{c} \cdot [\mathbf{t}'_1 \mid \cdots \mid \mathbf{t}'_4]^\top$.
  5. Compute $\Delta_{\mathbf{v}} \leftarrow \langle \vec{\mathbb{e}}_2, \vec{\mathbf{f}} \otimes \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}} + \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}} \otimes (\vec{\mathbf{f}} + \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}})\rangle + \mathbf{c} \cdot \langle \vec{\mathbf{e}}_3, \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}}\rangle + \langle \vec{\mathbf{e}}_4, \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}}\rangle$, where $\vec{\mathbb{e}}_2$ is the second standard basis vector in R_q^{16}, and $\vec{\mathbf{e}}_3$ and $\vec{\mathbf{e}}_4$ are the third and fourth standard basis vectors in R_q^4.
  6. Compute $\mathbf{v} \leftarrow \mathbf{v}' + \Delta_{\mathbf{v}}$.
  7. ▶ If $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}'_4]^\top, \vec{\mathbf{w}}' + \vec{\mathbf{w}}_{\mathcal{V}}, \mathbf{v}) \neq \perp$ abort. Else, set $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}'_4]^\top, \vec{\mathbf{w}}' + \vec{\mathbf{w}}_{\mathcal{V}}, \mathbf{v}) \leftarrow \mathbf{c}$.
  8. Retrieve $\mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}$ from $\mathsf{st}_{\mathsf{Sim}}$.
  9. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_4]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_j \in R_q^m$.
  10. ▶ Upon receiving $\{(\vec{\mathbf{w}}_k, \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_k}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_k}, \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_k}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}_k\rangle})\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
      (a) For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $\vec{\mathbf{w}}_i \leftarrow\$ R_q^n$ and compute

- $\mathsf{ctx}_{\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$, $\mathsf{ctx}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$
- $\vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^4)$, $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}}_i \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$

Finally, return $(\vec{\mathbf{w}}_i, \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_i}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i}, \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_i}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}}_i \rangle})$ as its output.

(b) For P_j, proceed like the other honest provers. However, instead of sampling $\vec{\mathbf{w}}_j$ uniformly at random, compute $\vec{\mathbf{w}}_j \leftarrow \vec{\mathbf{w}}' - \sum_{i \in \mathcal{H} \setminus \{j\}} \vec{\mathbf{w}}_i$. Return $(\vec{\mathbf{w}}_j, \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_j}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_j}, \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_j}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}}_j \rangle})$ as its output.

11. ▶ Upon receiving $\{(\mathsf{ds}_{\mathbf{t}_{4,k}}, \mathsf{ds}_{\mathbf{v}_k})\}_{k \in \mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
    (a) Compute $\vec{\mathsf{ctx}}_{\vec{\mathbf{y}}} \leftarrow \sum_{k \in \mathcal{S}} \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_k}$ and $\vec{\mathsf{ctx}}_{\vec{\mathbf{r}}} \leftarrow \sum_{k \in \mathcal{S}} \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_k}$.
    (b) Compute $\vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}} \leftarrow \sum_{k \in \mathcal{S}} \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_k}$ and $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}} \rangle} \leftarrow \sum_{k \in \mathcal{S}} \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}}_k \rangle}$.
    (c) Parse $\left[\mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle} \mid \cdots \mid \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{y}} \rangle}\right]^\top := \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}}$.
    (d) Compute $\mathsf{ctx}_{\mathbf{t}_4} \leftarrow \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}} \rangle} + \mathsf{ctx}_{\langle \vec{\mathbf{b}}_3, \vec{\mathbf{y}} \rangle} - \mathsf{ctx}_{\mathbf{m}_1} \cdot \mathsf{ctx}_{\langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}} \rangle} - \mathsf{ctx}_{\mathbf{m}_2} \cdot \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle}$.
    (e) Compute $\mathsf{ctx}_{\mathbf{v}} \leftarrow \mathsf{ctx}_{\langle \vec{\mathbf{b}}_4, \vec{\mathbf{y}} \rangle} + \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}} \rangle} \cdot \mathsf{ctx}_{\langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}} \rangle}$.
    (f) For all provers P_i, with $i \in \mathcal{H}$, :
        - Compute $\mathsf{ds}_{\mathbf{t}_{4,i}} \leftarrow \mathsf{Sim}_{\mathsf{tHE}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}, \mathsf{ctx}_{\mathbf{t}_4}, \mathbf{t}'_4, \mathcal{S}, \mathcal{V}, i)$.
        - Compute $\mathsf{ds}_{\mathbf{v}_i} \leftarrow \mathsf{Sim}_{\mathsf{tHE}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}, \mathsf{ctx}_{\mathbf{v}}, \mathbf{v}, \mathcal{S}, \mathcal{V}, i)$.
        - Return $(\mathsf{ds}_{\mathbf{t}_{4,i}}, \mathsf{ds}_{\mathbf{v}_i})$ to all provers in $\mathcal{V}$.
12. ▶ Upon receiving $\{\mathsf{ds}_{\vec{\mathbf{z}}_i}\}_{i \in \mathcal{V}}$ from provers in $\mathcal{V}$, do the following: compute $\mathsf{ctx}_{\vec{\mathbf{z}}} \leftarrow \mathsf{ctx}_{\vec{\mathbf{y}}} + \mathbf{c} \cdot \mathsf{ctx}_{\vec{\mathbf{r}}}$. Then, for each prover P_i, with $i \in \mathcal{H}$, compute $\mathsf{ds}_{\vec{\mathbf{z}}_i} \leftarrow \mathsf{Sim}_{\mathsf{tHE}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\mathsf{tHE}}}, \mathsf{ctx}_{\vec{\mathbf{z}}}, \vec{\mathbf{z}}' + \vec{\mathbf{z}}_{\mathcal{V}}, \mathcal{S}, \mathcal{V}, i)$ and return that as its output.

- $\mathsf{SimRO}(\mathsf{qry})$: Given a query $\mathsf{qry} \in R_q^{n+4} \times R_q^n \times R_q$, if $H(\mathsf{qry}) = \perp$, sample $\mathbf{c} \leftarrow \mathcal{C}$, and then, set $H(\mathsf{qry}) \leftarrow \mathbf{c}$. Return $H(\mathsf{qry})$.

Let $(\mathsf{D}, \mathcal{A})$ be a semi-honest ppt adversary in the weak $(\mathtt{t}\text{-}1)$-ZK experiment that performs at most Q_H random oracle queries and Q_P proof queries. We now proceed via a sequence of hybrids that interpolate between the weak $(\mathtt{t}\text{-}1)$-ZK experiment 0 and the weak $(\mathtt{t}\text{-}1)$-ZK experiment 1 (Definition 4.3):

- Hyb 0: This is the weak $(\mathtt{t}\text{-}1)$-ZK experiment 0 where Commit and Prove are instantiated as in Construction 2. Let E_0 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid.

- Hyb 1: Similar to Hyb 0, except the challenger samples a uniformly random string, uses it to initialize $(\mathsf{D}, \mathcal{A})$'s random tape, and stores this string in memory. Let E_1 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. This is purely syntactic, so $\Pr[\mathsf{E}_0] = \Pr[\mathsf{E}_1]$.

- Hyb 2: Similar to Hyb 1, except during Setup, the challenger samples $\mathbf{Q} \leftarrow\!\!\$\ \mathrm{GL}_n(R_q)$ and $\bar{\mathbf{B}}_0 \leftarrow\!\!\$\ R_q^{n \times m-n}$. Then, it sets $\mathbf{B}_0 \leftarrow \mathbf{Q} \cdot [\mathbf{I}_n \mid \bar{\mathbf{B}}_0]$, and keeps $\mathbf{Q}$ in memory. Let E_2 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Lemma 2.1, it follows that $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq O(hq^{-d/h}) \leq \mathsf{negl}(\lambda)$.

- Hyb 3: Similar to Hyb 2, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, where $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3) := \mathbb{w}$, the challenger executes $\Pi_{\mathsf{tCP}}.\mathsf{WitnessShare}$ and stores $\mathbb{w}$ in memory. Next, at the start of the execution of the Prove algorithm, it proceeds as follows:

  1. Given $\mathcal{S}$, the set of provers, and $\mathcal{V}$, the set of corrupted provers, compute $\mathcal{H} \leftarrow \mathcal{S} \setminus \mathcal{V}$ and store that in memory.
  2. Use $(\mathsf{D}, \mathcal{A})$'s randomness to sample $\vec{\mathbf{r}}_i \leftarrow D^m_{R,\mathfrak{s}_1}$ and $\vec{\mathbf{y}}_i \leftarrow D^m_{R,\mathfrak{s}_2}$ for each $i \in \mathcal{V}$.
  3. Sample $\vec{\mathbf{r}}_i \leftarrow D^m_{R,\mathfrak{s}_1}$ and $\vec{\mathbf{y}}_i \leftarrow D^m_{R,\mathfrak{s}_2}$ for each $i \in \mathcal{H}$.
  4. Compute $\vec{\mathbf{w}} \leftarrow \mathbf{B}_0 \cdot \sum_{i \in \mathcal{S}} \vec{\mathbf{y}}_i$.

5. Compute $\mathbf{t}_4 \leftarrow \langle \vec{\mathbf{b}}_4, \sum_{i\in\mathcal{S}} \vec{\mathbf{r}}_i\rangle + \langle \vec{\mathbf{b}}_3, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle - \mathbf{m}_2\langle \vec{\mathbf{b}}_1, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle - \mathbf{m}_1\langle \vec{\mathbf{b}}_2, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle$.
6. Compute $\mathbf{v} \leftarrow \langle \vec{\mathbf{b}}_4, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle + \langle \vec{\mathbf{b}}_1, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle \cdot \langle \vec{\mathbf{b}}_2, \sum_{i\in\mathcal{S}} \vec{\mathbf{y}}_i\rangle$.
7. Sample $\mathbf{c} \leftarrow \mathcal{C}$.
8. If $H([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) = \perp$, program $H([\vec{\mathbf{t}} \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v}) \leftarrow \mathbf{c}$. Otherwise, abort.

Let E_3 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By the correctness of Construction 2 and the fact that $(\mathsf{D}, \mathcal{A})$ is semi-honest, the values $\vec{\mathbf{w}}$, $\mathbf{t}_4$ and $\mathbf{v}$ computed in Steps 4, 5 and 6 are exactly the same values derived by all provers. Therefore, Hyb 2 and Hyb 3 are identical unless the programming of H fails. This happens if there exists a vector $\vec{\mathbf{w}}' \in R_q^n$ such that $(\mathsf{D}, \mathcal{A})$ queried H on an input that contained $\vec{\mathbf{w}}'$ during one of its $Q_P(Q_H + Q_P)$ queries and $\vec{\mathbf{w}} = \vec{\mathbf{w}}'$. Note that $\vec{\mathbf{w}} = \vec{\mathbf{w}}_j + \sum_{i\in\mathcal{S}\setminus\{j\}} \vec{\mathbf{w}}_i$, where j is the index of an honest prover in $\mathcal{S}$, which exists since $(\mathsf{D}, \mathcal{A})$ controls at most $\mathtt{t}$-1 provers. Since P_j is honest, we have $\vec{\mathbf{w}}_j = \mathbf{Q} \cdot [\mathbf{I}_n \mid \bar{\mathbf{B}}_0]\vec{\mathbf{y}}_j$, and this implies that $\vec{\mathbf{w}} = \vec{\mathbf{w}}'$ if and only if $\vec{\mathbf{y}}_{j,0} = \mathbf{Q}^{-1}(\vec{\mathbf{w}}' - \sum_{i\in\mathcal{S}\setminus\{j\}} \vec{\mathbf{w}}_i) - \bar{\mathbf{B}}_0\vec{\mathbf{y}}_{j,1}$, where $[\vec{\mathbf{y}}_{j,0}^\top \mid \vec{\mathbf{y}}_{j,1}^\top]^\top := \vec{\mathbf{y}}_j \leftarrow D_{R,\mathfrak{s}_2}^m$. Therefore, by Lemma 2.3, it follows that $|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq Q_P(Q_H + Q_P) \cdot 2^{-dn}$.

- Hyb 4 : Similar to Hyb 3, except during Setup, the challenger replaces Π_{tHE} with $\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}$, i.e., it computes $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}, \mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}) \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimKGen}(1^\lambda, \mathtt{n}, \mathtt{t})$ and stores $\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}$ in memory. Next, during the execution of the Prove protocol, it proceeds as follows:

  1. Using the vectors $(\vec{\mathbf{r}}_i, \vec{\mathbf{y}}_i)$ for $i \in \mathcal{V}$ sampled with $(\mathsf{D}, \mathcal{A})$'s randomness, compute $\vec{\mathbf{z}}_{\mathcal{V}} \leftarrow \sum_{i\in\mathcal{V}}(\vec{\mathbf{y}}_i + \mathbf{c} \cdot \vec{\mathbf{r}}_i)$.
  2. Analogously, compute $\vec{\mathbf{z}}_{\mathcal{H}} \leftarrow \sum_{i\in\mathcal{H}}(\vec{\mathbf{y}}_i + \mathbf{c} \cdot \vec{\mathbf{r}}_i)$.
  3. Let $\vec{\mathbb{e}}_2$ be the second standard basis vector in R_q^{16}, and let $\vec{\mathbf{e}}_3$ and $\vec{\mathbf{e}}_4$ be the third and fourth standard basis vectors in R_q^4. Compute $\vec{\mathbf{f}} \leftarrow \mathbf{B}_1 \cdot \vec{\mathbf{z}}_{\mathcal{H}} - \mathbf{c} \cdot [\mathbf{t}_1 \mid \cdots \mid \mathbf{t}_4]^\top$, where $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3$ are the values computed during the execution of the Commit protocol, and the set
  $$\mathbf{v}' \leftarrow \langle \vec{\mathbb{e}}_2, \vec{\mathbf{f}} \otimes \vec{\mathbf{f}}\rangle + \mathbf{c} \cdot \langle \vec{\mathbf{e}}_3, \vec{\mathbf{f}}\rangle + \langle \vec{\mathbf{e}}_4, \vec{\mathbf{f}}\rangle,$$
  $$\Delta_{\mathbf{v}} \leftarrow \langle \vec{\mathbb{e}}_2, \vec{\mathbf{f}} \otimes \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}} + \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}} \otimes (\vec{\mathbf{f}} + \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}})\rangle + \mathbf{c} \cdot \langle \vec{\mathbf{e}}_3, \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}}\rangle + \langle \vec{\mathbf{e}}_4, \mathbf{B}_1\vec{\mathbf{z}}_{\mathcal{V}}\rangle.$$
  4. Compute $\mathbf{v} \leftarrow \mathbf{v}' + \Delta_{\mathbf{v}}$
  5. Let $\mathsf{ctx}_{\mathbf{t}_4}$, $\mathsf{ctx}_{\mathbf{v}}$, and $\mathsf{ctx}_{\vec{\mathbf{z}}}$ be the ciphertexts corresponding to $\mathbf{t}_4$, $\mathbf{v}$, and $\vec{\mathbf{z}}$, respectively, computed during the execution of the Prove protocol.
  6. Upon receiving $\{(\mathsf{ds}_{\mathbf{t}_{4,k}}, \mathsf{ds}_{\mathbf{v}_k}, \mathsf{ds}_{\vec{\mathbf{z}}_k})\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, then, for all provers $\mathsf{P}_i \in \mathcal{H}$, compute
     - $\mathsf{ds}_{\mathbf{t}_{4,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{t}_4}, \mathbf{t}_4, \mathcal{S}, \mathcal{V}, i)$,
     - $\mathsf{ds}_{\mathbf{v}_i} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{v}}, \mathbf{v}, \mathcal{S}, \mathcal{V}, i)$.
     - $\mathsf{ds}_{\vec{\mathbf{z}}_i} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\vec{\mathbf{z}}}, \vec{\mathbf{z}}_{\mathcal{V}} + \vec{\mathbf{z}}_{\mathcal{H}}, \mathcal{S}, \mathcal{V}, i)$.

     and return those as its output.

  Let E_4 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Remark D.2, the way $\mathbf{v}$ is computed in Step 4 of this hybrid is identical to the way it is computed in Construction 2. Consequently, by inspection and by the correctness of Π_{tHE}, the values $\mathbf{t}_4$, $\mathbf{v}$ and $\vec{\mathbf{z}}$ obtained by provers in Hyb 4 are indistinguishable from those obtained in Hyb 3. Hence, the only difference between Hyb 4 and Hyb 3 lies in the generation of (1) the public and secret encryption keys of all parties and (2) the decryption shares of the honest parties. Therefore, by the simulation security of Π_{tHE}, it follows that $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \leq \mathsf{negl}(\lambda)$.

- Hyb 5: Similar to Hyb 4, except ciphertexts produced by honest provers are replaced with ciphertexts of zero. Specifically, during the execution of the Prove protocol, upon receiving $\{(\vec{\mathbf{w}}_k, \mathsf{ctx}_{\vec{\mathbf{y}}_k}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_k}, \mathsf{ctx}_{\vec{\mathbf{r}}_k}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}_k\rangle})\}_{k\in\mathcal{V}}$ from the provers in $\mathcal{V}$, the challenger computes, for each $\mathsf{P}_i \in \mathcal{H}$,

  - $\mathsf{ctx}_{\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$ and $\mathsf{ctx}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$,
  - $\vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^4)$ and $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}_i\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$,

  and returns $(\vec{\mathbf{w}}_i, \vec{\mathsf{ctx}}_{\vec{\mathbf{y}}_i}, \vec{\mathsf{ctx}}_{\mathbf{B}_1\vec{\mathbf{y}}_i}, \vec{\mathsf{ctx}}_{\vec{\mathbf{r}}_i}, \mathsf{ctx}_{\langle\vec{\mathbf{b}}_4,\vec{\mathbf{r}}_i\rangle})$ as P_i's output.

  Let E_5 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Since the correct values of $\mathbf{t}_4$, $\mathbf{v}$, and $\vec{\mathbf{z}}$ are used to simulate the decryption shares, Hyb 4 and Hyb 5 are indistinguishable unless $(\mathsf{D}, \mathcal{A})$ can distinguish encryptions of $\mathbf{0}^j$, for $1 \le j \le m$, from encryptions of other messages. Therefore, by the semantic security of Π_{tHE}, $|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \le \mathsf{negl}(\lambda)$.

- Hyb 6 : Similar to Hyb 5, except the vectors $\vec{\mathbf{w}}_i$ produced by all but one honest prover are replaced with random vectors over R_q^n. Specifically, the challenger samples $j \leftarrow\!\!\$\ \mathcal{H}$ and keeps it in memory; this index corresponds to the last honest prover to reply. Then, during the execution of the Prove protocol, the challenger proceeds as follows:

  - For each $\mathsf{P}_i \in \mathcal{H} \setminus \{j\}$, sample $\vec{\mathbf{w}}_i \leftarrow\!\!\$\ R_q^n$ and return that as P_i's output.
  - For P_j, compute $\vec{\mathbf{w}}_j \leftarrow \mathbf{B}_0 \cdot \left(\sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_i\right) - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{w}}_i$, and return that as P_j's output.

  Let E_6 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Observe that $\mathbf{B}_0 = \mathbf{Q} \cdot [\mathbf{I}_n \mid \bar{\mathbf{B}}_0]$, where $\mathbf{Q} \in \mathrm{GL}_n(R_q)$ and $\bar{\mathbf{B}}_0 \leftarrow\!\!\$\ R_q^{n\times(m-n)}$, so that each $\vec{\mathbf{w}}_i = \mathbf{B}_0\vec{\mathbf{y}}_i$ is an MLWE sample. However, note that $\vec{\mathbf{y}}_i$ is released as a noisy vector as part of $\vec{\mathbf{z}}$. Hence, by the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ assumption, we can gradually replace the contribution of each prover $\mathsf{P}_i \in \mathcal{H} \setminus \{j\}$ with a random sample across the Q_P proof queries performed by $(\mathsf{D}, \mathcal{A})$. Thus, we obtain the following claim, which we prove later.

  Claim 2. $|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \le \mathtt{n} \cdot Q_P \cdot \epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}}$, *where* $\epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}}$ *is the advantage of a* ppt *distinguisher against the* $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ *assumption.*

  Since the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ assumption holds by hypothesis, we conclude that $|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \le \mathsf{negl}(\lambda)$.

- Hyb 7 : Similar to Hyb 6, except during the execution of Setup, the challenger samples $\mathbf{V} \leftarrow\!\!\$\ \mathrm{GL}_{n+4}(R_q)$ and $\mathbf{A} \leftarrow\!\!\$\ R_q^{(n+4)\times m-n-4}$, and then, computes $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{n+4} \mid \mathbf{A}]$ and keeps $\mathbf{V}$ in memory. Let E_7 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Lemma 2.1, we have $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]| \le O(hq^{-d/h}) \le \mathsf{negl}(\lambda)$.

- Hyb 8 : Similar to Hyb 7, except during the execution of Commit, the contributions of all but one honest prover are replaced with random vectors over R_q^{n+3}. Specifically, the challenger proceeds as follows:

  - For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $[\vec{\mathbf{t}}_{0,i}^\top \mid \mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]^\top \leftarrow\!\!\$\ R_q^{n+3}$ and return it as P_i's output.

– Compute

$$\vec{\mathbf{t}}'_0 \leftarrow \mathbf{B}_0 \cdot (\sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i) \text{ and}$$
$$[\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_2 \mid \vec{\mathbf{b}}_3]^\top \cdot (\sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i) + \sum_{i\in\mathcal{H}} \lambda_i [[\![\mathbf{m}]\!]_{1,i} \mid [\![\mathbf{m}]\!]_{2,i} \mid [\![\mathbf{m}]\!]_{3,i}]^\top$$

– For P_j, compute $\vec{\mathbf{t}}_{0,j} \leftarrow \vec{\mathbf{t}}'_0 - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{t}}_{0,i}$ and $[\mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}] \leftarrow [\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] - \sum_{i\in\mathcal{H}\setminus\{j\}} [\mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]$. Then, return $[\vec{\mathbf{t}}^\top_{0,j} \mid \mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}]^\top$ as its output.

Let E_8 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By an argument similar to the one developed in Hyb 6, but applied to $\vec{\mathbf{r}}_i$, we have $|\Pr[\mathsf{E}_7] - \Pr[\mathsf{E}_8]| \leq \mathtt{n} \cdot Q_P \cdot \epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}}$, where $\epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}}$ is the advantage of a ppt distinguisher against the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$ assumption. Since the assumption holds by hypothesis, we conclude that $|\Pr[\mathsf{E}_7] - \Pr[\mathsf{E}_8]| \leq \mathsf{negl}(\lambda)$.

- Hyb 9 : Similar to Hyb 8, except during the execution of Commit and Prove, the challenger replaces $\sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i$ and $\sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_i$ with $\vec{\mathbf{r}}_\mathcal{H} \leftarrow D^m_{R,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$ and $\vec{\mathbf{y}}_\mathcal{H} \leftarrow D^m_{R,\mathfrak{s}_2\sqrt{|\mathcal{H}|}}$, respectively.

  Let E_9 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Lemma 2.4, $|\Pr[\mathsf{E}_8] - \Pr[\mathsf{E}_9]| \leq \mathsf{negl}(\lambda)$.

- Hyb 10 : Similar to Hyb 9, except during the execution of Commit, the challenger sets $[\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_2 \mid \vec{\mathbf{b}}_3]^\top \cdot \vec{\mathbf{r}}_\mathcal{H} + [\mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3]^\top$, and for the last honest prover P_j, it sets $[\mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}] \leftarrow [\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] - \sum_{i\in\mathcal{S}\setminus\{j\}} [\mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]$. Next, during the execution of Prove, it sets $\mathbf{t}_4 \leftarrow \langle \vec{\mathbf{b}}_4, \vec{\mathbf{r}}_\mathcal{H} \rangle + \langle \vec{\mathbf{b}}_3, \vec{\mathbf{y}}_\mathcal{H} \rangle - \mathbf{m}_2 \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\mathcal{H} \rangle - \mathbf{m}_1 \langle \vec{\mathbf{b}}_2, \vec{\mathbf{y}}_\mathcal{H} \rangle$.

  Let E_{10} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Following the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$ assumption, implied by the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$ assumption (see Remark A.1), the $\mathbf{t}_4$ is indistinguishable from its value in Hyb 9, and the contribution of P_j during Commit is indistinguishable from that in Hyb 9. Hence, we obtain the following claim that we prove later.

  Claim 3. $|\Pr[\mathsf{E}_9] - \Pr[\mathsf{E}_{10}]| \leq 2 \cdot Q_P \cdot \epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}}$, *where* $\epsilon_{\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}}$ *is the advantage of a* ppt *distinguisher against the* $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$ *assumption.*

  Since the assumption holds by hypothesis, we have $|\Pr[\mathsf{E}_9] - \Pr[\mathsf{E}_{10}]| \leq \mathsf{negl}(\lambda)$.

- Hyb 11 : Similar to Hyb 10, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger runs vALS.Prove (see Construction 1) with $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$ and standard deviations $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to obtain $(\vec{\mathbf{t}}', \pi := (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}, \vec{\mathbf{z}}'))$.

  During the execution of Commit protocol, upon receiving $\{[\vec{\mathbf{t}}^\top_{0,k} \mid \mathbf{t}_{1,k} \mid \mathbf{t}_{2,k} \mid \mathbf{t}_{3,k}]^\top\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, the challenger parses $[(\vec{\mathbf{t}}'_0)^\top \mid \mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3]^\top := \vec{\mathbf{t}}'$ and uses those values to compute the last honest prover P_j's contribution. Next, during the execution of Prove, it proceeds as follows:

  – During the computation of $\vec{\mathbf{w}}$, set the contribution of P_j to be $\vec{\mathbf{w}}_j \leftarrow \vec{\mathbf{w}}' - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{w}}_i$.

  – Set $\mathbf{t}_4 := \mathbf{t}'_4$ and use it to simulate honest provers' decryption shares.

  – Set $\vec{\mathbf{z}}_\mathcal{H} := \vec{\mathbf{z}}'$.

Let E_{11} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. This is just a syntactic. Therefore, $\Pr[\mathsf{E}_{10}] = \Pr[\mathsf{E}_{11}]$.

- Hyb 12 : Similar to Hyb 11, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger only uses $\mathbb{w}$ to run $\Pi_{\mathsf{tCP}}.\mathsf{WitnessShare}$. Next, it runs $\mathsf{Sim}_{\mathsf{vALS}}.\mathsf{SimProve}$ with $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$ and standard deviations $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to obtain $(\vec{\mathbf{t}}', \pi := (\mathbf{t}'_4, \vec{\mathbf{w}}', \mathbf{v}', \mathbf{c}, \vec{\mathbf{z}}'))$. Let E_{12} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By the zero-knowledge property of vALS, it follows that $|\Pr[\mathsf{E}_{11}] - \Pr[\mathsf{E}_{12}]| \leq \mathsf{negl}(\lambda)$.

- Hyb 13 : Similar to Hyb 12, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger discards $\mathbb{w}$ and executes $\mathsf{SimWitShare}$. Let E_{13} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. From Hyb 12, the executions of Commit and Prove are independent of the witness; therefore, the only difference is in the execution of the WitnessShare algorithm, and by the security of Shamir secret sharing and the semantic security of Π_{tHE}, it follows that $|\Pr[\mathsf{E}_{12}] - \Pr[\mathsf{E}_{13}]| \leq \mathsf{negl}(\lambda)$.

  Finally, note that Hyb 13 is identical to the weak $(\mathtt{t}\text{-}1)$-ZK Experiment 1 (Definition 4.3).

Claim 2. We prove the claim via a sequence of hybrids that interpolate between Hyb 5 and Hyb 6. Let j be the index of the last honest prover, and write $\mathcal{H} \setminus \{j\} = \{\mathsf{P}_1, \ldots, \mathsf{P}_{|\mathcal{H}|-1}\}$. Adopt the convention that Hyb $(1, 0) :=$ Hyb 5 and Hyb $(k, 0) :=$ Hyb $(k-1, |\mathcal{H}|-1)$ for $k > 1$. For $k \in \{1, \ldots, Q_P\}$ and $\ell \in \{1, \ldots, |\mathcal{H}|-1\}$, let Hyb (k, ℓ) be similar to Hyb $(k, \ell-1)$, except that, for the k-th proof query, the contribution of P_ℓ to the computation of $\vec{\mathbf{w}}$ is replaced by a random vector in R_q^n, and the contribution of P_j is adjusted accordingly, i.e., $\vec{\mathbf{w}}_j \leftarrow \mathbf{B}_0\big(\vec{\mathbf{y}}_j + \sum_{i\in[\ell]} \vec{\mathbf{y}}_i\big) - \sum_{i\in[\ell]} \vec{\mathbf{w}}_i$. Clearly, in Hyb $(Q_P, |\mathcal{H}|-1)$, the contributions of all honest provers except P_j are replaced by random vectors for all proof queries, which is precisely Hyb 6.

Now, by the $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ assumption, Hyb (k, ℓ) is indistinguishable from Hyb $(k, \ell-1)$. Indeed, given a tuple $(\bar{\mathbf{B}}_0, \vec{\mathbf{v}}, (1, \vec{\mathbf{y}}' = \vec{\mathbf{y}}_j + \vec{\mathbf{y}}_\ell))$ that corresponds to an $\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-n,n,q,\mathfrak{s}_2}$ challenge, we can set $\vec{\mathbf{w}}_\ell = \mathbf{Q}\vec{\mathbf{v}}$ and use $\vec{\mathbf{y}}'$ to compute the contribution of P_j during the k-th proof query. If $\vec{\mathbf{v}} = [\mathbf{I} \mid \bar{\mathbf{B}}_0]\vec{\mathbf{y}}_\ell$, then this corresponds to Hyb $(k, \ell-1)$, and if $\vec{\mathbf{v}}$ is random, then $\vec{\mathbf{w}}_\ell$ is random since $\mathbf{Q} \in \mathrm{GL}_n(R_q)$, and this corresponds to Hyb (k, ℓ). Since we have $Q_P \cdot (|\mathcal{H}|-1)$ hybrids in which such a replacement is made, and $|\mathcal{H}| \leq \mathtt{n}$, the claim readily follows. □

Claim 3. We use a sequence of hybrids that interpolate between Hyb 9 and Hyb 10.

- Hyb 0: This is Hyb 9.

- Hyb 1: Similar to Hyb 0, except $\vec{\mathbf{t}}'_0 \hookleftarrow\$\ R_q^n$, $[\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] \hookleftarrow\$\ R_q^3$ and $\mathbf{t}_4 \leftarrow \mathbf{t}'_4 + \langle\vec{\mathbf{b}}_4, \sum_{i\in\mathcal{V}} \vec{\mathbf{r}}_i\rangle + \langle\vec{\mathbf{b}}_3, \sum_{i\in\mathcal{V}} \vec{\mathbf{y}}_i\rangle - \mathbf{m}_2\langle\vec{\mathbf{b}}_1, \sum_{i\in\mathcal{V}} \vec{\mathbf{y}}_i\rangle - \mathbf{m}_1\langle\vec{\mathbf{b}}_2, \sum_{i\in\mathcal{V}} \vec{\mathbf{y}}_i\rangle$, where $\mathbf{t}'_4 \hookleftarrow\$\ R_q$. Clearly, this change is indistinguishable from Hyb 0 by the Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$.

- Hyb 2: Similar to Hyb 1, except $\mathbf{t}_4 \hookleftarrow\$\ R_q$ and during the execution of Commit, for the last honest prover P_j, set $[\mathbf{t}_{1,j} \mid \mathbf{t}_{2,j} \mid \mathbf{t}_{3,j}] \leftarrow [\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] - \sum_{i\in\mathcal{S}\setminus\{j\}}[\mathbf{t}_{1,i} \mid \mathbf{t}_{2,i} \mid \mathbf{t}_{3,i}]$. Given that, from Hyb 1, $[\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3]$ is random, the contribution of P_j is perfectly indistinguishable from its contribution in Hyb 1, in addition, since in Hyb 1, $\mathbf{t}'_4$ is random, the distribution of $\mathbf{t}_4$ in this hybrid is indistinguishable from that in Hyb 1.

- Hyb 3: Similar to Hyb 2, except $\vec{\mathbf{t}}'_0 \leftarrow \mathbf{B}_0 \cdot \vec{\mathbf{r}}_\mathcal{H}$, $[\mathbf{t}'_1 \mid \mathbf{t}'_2 \mid \mathbf{t}'_3] \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_2 \mid \vec{\mathbf{b}}_3]^\top \cdot \vec{\mathbf{r}}_\mathcal{H} + [\mathbf{m}_1 \mid \mathbf{m}_2 \mid \mathbf{m}_3]^\top$, and $\mathbf{t}_4 \leftarrow \langle\vec{\mathbf{b}}_4, \vec{\mathbf{r}}_\mathcal{H}\rangle + \langle\vec{\mathbf{b}}_3, \vec{\mathbf{y}}_\mathcal{H}\rangle - \mathbf{m}_2\langle\vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\mathcal{H}\rangle - \mathbf{m}_1\langle\vec{\mathbf{b}}_2, \vec{\mathbf{y}}_\mathcal{H}\rangle$. This change is indistinguishable from Hyb 2 by the Q_P-$\mathsf{Hint\text{-}MLWE}^{1,\mathfrak{s}_2\sqrt{|\mathcal{H}|},\mathcal{C}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$. Finally, note that this hybrid is identical to Hyb 10.

□

This completes the proof. □

D.4 Proof of Theorem 5.7

Proof of Theorem 5.7. Let $\mathsf{Sim} = (\mathsf{SimSetup}, \mathsf{SimCom}, \mathsf{SimProve}, \mathsf{SimRO})$ be the simulator described in the proof of Theorem 5.5 (Simulator 2). For any $\lambda \in \mathbb{N}$, let $\mathcal{A}$ be a semi-honest ppt adversary that wins the threshold simulation extractability experiment (see Definition). We show that under $\mathsf{MSIS}_{n,m,q,8d\beta'}$ assumption, $\mathcal{A}$ success probability is negligible.

First note that the outputs of the thresholdize proof is identical to a single-prover proof at the exception of the $\sqrt{\mathtt{t}}$ factor in the norm-bound. Therefore, if the proof $(\vec{\mathbf{t}}^*, \pi^* := (\mathbf{t}_4^*, \vec{\mathbf{w}}^*, \mathbf{v}^*, c^*, \vec{\mathbf{z}}^*))$ has the same first message as one of the proof produced by interacting with the simulator, i.e., there exists $(\vec{\mathbf{t}}, \pi := [\mathbf{t}_4, \vec{\mathbf{w}}, \mathbf{v}, \vec{\mathbf{z}}]) \in \mathcal{Q}_{\mathbb{x}}$ such that $([(\vec{\mathbf{t}}^*)^\top \mid \mathbf{t}_4^*]^\top, \vec{\mathbf{w}}^*, \mathbf{v}^*) = ([\vec{\mathbf{t}}^\top \mid \mathbf{t}_4]^\top, \vec{\mathbf{w}}, \mathbf{v})$ then using the same strategy used to prove weak unique response of vALS (see proof of Theorem 5.3), we can obtain an MSIS solution of ℓ_2-norm $2\beta' < 8d\beta'$. Hence, we assume that the proof generated by $\mathcal{A}$ is fresh and does not share a first message with any of the proof generated by interacting with the simulator.

Let $\mathsf{Ext}_{\mathsf{vALS}}$ be the extractor of vALS, which exists by the predicate special-soundess of vALS (see Theorem 5.4). We construct an extractor Ext that extract a witness from $\mathcal{A}$ using $\mathsf{Ext}_{\mathsf{vALS}}$ as follows:

- Ext controls the random oracle and emulate Sim for $\mathcal{A}$.
- Upon receiving a proof $(\vec{\mathbf{t}}^*, \pi^*) \notin \mathcal{Q}_{\mathbb{x}}$ from $\mathcal{A}$, Ext calls $\mathsf{Ext}_{\mathsf{vALS}}$ and keeps using Sim to answer to $\mathcal{A}$'s subsequent proof queries.
- When $\mathsf{Ext}_{\mathsf{vALS}}$ issue a rewind command, Ext rewinds both $\mathcal{A}$ and Sim to ensure that simulated queries are consistent.

Clearly, by Theorem 5.4, we extract a valid witness (weak opening) from $\mathcal{A}$ except with negligible probability. □

E Proofs of Section 6

E.1 Proof of Theorem 6.1

Proof of Theorem 6.1. Let $\left(\vec{\mathbf{t}}, \pi := \left(\vec{\mathbf{t}}_2, \vec{\mathbf{t}_3}, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^{\ell}\right)\right)$ be an honest proof generated by executing the Prove algorithm of Construction 3 on a statement-witness pair $(\mathbb{x} := (A, \vec{u}), \mathbb{w} := \mathbf{s})$. Following an argument similar to the one developed in the proof of Theorem 5.1, $\|\vec{\mathbf{z}}_\theta\| \leq \beta$, for all $\theta \in [\ell]$, except with probability at most $\ell \cdot 2^{-md+1}$, which is negligible. The rest of the proof follows by inspection. □

E.2 Proof of Theorem 6.2

Proof of Theorem 6.2. First, we describe the simulator.

Simulator 3: $\mathsf{Sim}_{\mathsf{vENS}}$

The simulator tracks and answers random oracle queries.

- $\mathsf{SimSetup}(1^\lambda)$: Given $\lambda \in \mathbb{N}$, proceed as follows:
  1. Sample $\mathbf{V} \leftarrow\!\$\ \mathrm{GL}_{n+3\ell+1}(R_q)$ and $\mathbf{A} \leftarrow\!\$\ R_q^{(n+3\ell+1)\times(m-n-3\ell-1)}$.
  2. Compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{n+3\ell+1} \mid \mathbf{A}]$, where $\mathbf{B}_0 \in R_q^{n\times m}$ and $\mathbf{B}_1 \in R_q^{(3\ell+1)\times m}$.
  3. Return $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$, and keep $\mathbf{V}$ in memory.

- $\mathsf{SimProve}(\mathsf{pp}, \mathbb{x})$: Given $(\mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$ and $(A, \vec{u}) := \mathbb{x}$ proceed as follows:
  1. Sample $\vec{\mathbf{u}} \leftarrow\!\$\ R_q^{n+3\ell+1}$, and $\vec{\mathbf{r}} \leftarrow D^m_{R,\mathfrak{s}_1}$.
  2. For $\theta \in [\ell]$, do the following:
     - Sample $\vec{v}_\theta \leftarrow\!\$\ \mathbb{Z}_q^\nu, \mathbf{c}_\theta \leftarrow\!\$\ \mathcal{C}$.
     - Compute $\vec{\mathbf{z}}_\theta \leftarrow D^m_{R,\mathfrak{s}_2} + \mathbf{c}_\theta \vec{\mathbf{r}}$.
     - Sample $\mathbf{h}'_\theta \leftarrow\!\$\ \{\mathbf{h} \in R_q : h_0 = 0\}$ and set $\mathbf{h}_\theta \leftarrow \mathbf{h}'_\theta + \langle \vec{u}, \vec{v}_\theta \rangle$.
  3. Set $\vec{\mathbf{h}} := [\mathbf{h}_1 \mid \cdots \mid \mathbf{h}_\ell]^\top$.
  4. Compute $[\vec{\mathbf{t}}^\top \mid \vec{\mathbf{t}}_2^\top \mid \vec{\mathbf{t}}_3^\top]^\top \leftarrow \mathbf{V}\vec{\mathbf{u}}$, with $\vec{\mathbf{t}} \in R_q^{n+1}, \vec{\mathbf{t}}_2 \in R_q^\ell, \vec{\mathbf{t}}_3 \in R_q^{2\ell}$.
  5. Parse $\left[\vec{\mathbf{t}}_0^\top \mid \mathbf{t}_1\right]^\top := \vec{\mathbf{t}}$, and $[\mathbf{t}_{\ell+2} \mid \cdots \mid \mathbf{t}_{3\ell+1}]^\top := \vec{\mathbf{t}}_3$.
  6. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
  7. For $\theta \in [\ell]$, do the following:
     (a) Compute $\vec{\mathbf{w}}_\theta \leftarrow \mathbf{B}_0 \vec{\mathbf{z}}_\theta - \mathbf{c}_\theta \vec{\mathbf{t}}_0$.
     (b) Compute $\begin{bmatrix} \mathbf{f}_{\theta,1} \\ \mathbf{f}_{\theta,2} \\ \mathbf{f}_{\theta,3} \end{bmatrix} \leftarrow \begin{bmatrix} \vec{\mathbf{b}}_1^\top \\ \vec{\mathbf{b}}_{\ell+2\theta}^\top \\ \vec{\mathbf{b}}_{\ell+2\theta+1}^\top \end{bmatrix} \vec{\mathbf{z}}_\theta - \mathbf{c}_\theta \cdot \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_{\ell+2\theta} \\ \mathbf{t}_{\ell+2\theta+1} \end{bmatrix}$.
     (c) Compute $\mathbf{v}_\theta \leftarrow \mathbf{f}_{\theta,1}(\mathbf{f}_{\theta,1} - \mathbf{c}_\theta)(\mathbf{f}_{\theta,1} + \mathbf{c}_\theta) + \mathbf{f}_{\theta,2} + \mathbf{c}_\theta \cdot \mathbf{f}_{\theta,3}$.
     (d) Compute $\boldsymbol{\psi}_\theta \leftarrow \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta)$, and $\boldsymbol{\tau}_\theta \leftarrow \boldsymbol{\psi}_\theta \mathbf{t}_1$.
  8. Set $\vec{\mathbf{v}} := [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell]^\top$.
  9. Let
  $$\vec{\mathbf{v}}' \leftarrow \begin{bmatrix} \langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{z}}_1 \rangle \\ \vdots \\ \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{z}}_\ell \rangle \end{bmatrix} - \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_\ell \end{bmatrix} \odot \left(\begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_\ell \end{bmatrix} + \vec{\mathbf{t}}_2 - \vec{\mathbf{h}} \right)$$
  10. ▶ If $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell) \neq \perp$, abort. Else, set $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell) \leftarrow (\vec{v}_\theta)_{\theta=1}^\ell$.
  11. ▶ If $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}') \neq \perp$, abort. Else, set $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}') \leftarrow (\mathbf{c}_\theta)_{\theta=1}^\ell$.
  12. Return $\left(\vec{\mathbf{t}}, \pi := \left(\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^\ell\right)\right)$.

- $\mathsf{SimRO}(j, \mathsf{qry})$: Given a query qry if $H_j(\mathsf{qry}) = \perp$, then proceed as follows: If $j = 1$, sample $(\vec{v}_\theta)_{\theta=1}^\ell \leftarrow\!\$\ (\mathbb{Z}_q^\nu)^\ell$ and set $\mathsf{output} := (\vec{v}_\theta)_{\theta=1}^\ell$. If $j = 2$, sample $(\mathbf{c}_\theta)_{\theta=1}^\ell \leftarrow\!\$\ \mathcal{C}^\ell$ and set $\mathsf{output} := (\mathbf{c}_\theta)_{\theta=1}^\ell$. Then set $H_j(\mathsf{qry}) \leftarrow \mathsf{output}$. Return $H_j(\mathsf{qry})$.

Let D be a ppt distinguisher that performs Q_H random oracle queries and Q_P proof queries. We now proceed via a sequence of hybrids:

- $\mathsf{Hyb}\ 0$: This is the real game where each H_i is implemented by lazy sampling and commitments and proofs are computed using the Prove algorithm of Construction 3.

- $\mathsf{Hyb}\ 1$: Similar to $\mathsf{Hyb}\ 0$, except during Setup, sample $\mathbf{V} \leftarrow\!\$\ \mathrm{GL}_{n+3\ell+1}(R_q)$ and $\mathbf{A} \leftarrow\!\$\ R_q^{(n+3\ell+1)\times(m-n-3\ell-1)}$, and then compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{n+3\ell+1} \mid \mathbf{A}]$ and keep $\mathbf{V}$ in memory. By Lemma 2.1, the statistical distance between $\mathsf{Hyb}\ 1$ and $\mathsf{Hyb}\ 0$ is $O(hq^{-d/h})$, which is negligible.

- $\mathsf{Hyb}\ 2$: Similar to $\mathsf{Hyb}\ 1$, except during Prove on $\mathbb{x}$, sample $(\vec{v}_\theta)_{\theta=1}^\ell \leftarrow\!\$\ (\mathbb{Z}_q^\nu)^\ell$ and $(\mathbf{c}_\theta)_{\theta=1}^\ell \leftarrow\!\$\ \mathcal{C}^\ell$ first. Then execute the remaining operations while skipping random-oracle queries. Upon obtaining a complete transcript $(\vec{\mathbf{t}}, \pi := (\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^\ell))$, program

$H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}) \leftarrow (\vec{v}_\theta)_{\theta=1}^{\ell}$ and $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}') \leftarrow (\mathbf{c}_\theta)_{\theta=1}^{\ell}$.

Unless the programming of H_1 and H_2 in Hyb 2 creates an inconsistency, that is, unless the programmed values overwrite existing values in H_1 or H_2, Hyb 2 is identical to Hyb 1. Such an inconsistency occurs only if D has already queried H_1 or H_2 on one of the inputs at which we later program them. Let coll denote this event. Then, by a succession of union bounds, we have $\Pr[\mathsf{coll}] \leq 2Q_P(Q_H + Q_P) \cdot \Pr[(\vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3) = \vec{\mathbf{t}}']$, where $\vec{\mathbf{t}}' \in R_q^{n+3\ell+1}$ is a value appearing in one of D's queries. Since $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top = \mathbf{V} \cdot [\mathbf{I}_{n+3\ell+1} \mid \mathbf{A}]$, where $\mathbf{V} \in \mathrm{GL}_{n+3\ell+1}(R_q)$, Lemma 2.3 implies that $\Pr[\mathsf{coll}] \leq 2Q_P(Q_H + Q_P) \cdot 2^{-d(n+3\ell+1)}$, which is negligible.

- Hyb 3: Similar to Hyb 2, except Prove is replaced by SimProve from Simulator 3 while taking into account the witness $\mathbf{s}$. More specifically, Step 4 is replaced with $[\vec{\mathbf{t}}^\top \mid \vec{\mathbf{t}}_2^\top \mid \vec{\mathbf{t}}_3^\top]^\top \leftarrow \mathbf{V}\vec{\mathbf{u}} + [\mathbf{0}^n \mid \mathbf{s} \mid \vec{\mathbf{g}}^\top \mid \mathbf{m}_{2+\ell} \mid \cdots \mid \mathbf{m}_{3\ell+1}]^\top$, where each $\vec{\mathbf{g}}$ is uniformly sampled over R_q^ℓ such that for each component $\mathbf{g}_\theta$, $\mathsf{ct}(\mathbf{g}_\theta) = 0$, and $\mathbf{m}_\theta$ is computed using the $\vec{\mathbf{z}}_\theta$, $\vec{\mathbf{r}}$, $\mathbf{s}$.

  Using an argument similar to the one developed in the proof of Claim 1, Hyb 3 is computationally indistinguishable from Hyb 2 under the hardness of the Q_P-Hint-MLWE$^{\ell,\mathfrak{s}_2,\mathcal{C}}_{m-n-3\ell-1,n+3\ell+1,q,\mathfrak{s}_1}$ assumption.

- Hyb 4: Similar to Hyb 3, except during the execution of a proof query, discard the witness and run SimProve. Note that in both Hyb 4 and Hyb 3, $\vec{\mathbf{u}}$ is sampled uniformly over $R_q^{n+3\ell+1}$ and therefore hides any information about the witness. Hence, the two hybrids are perfectly indistinguishable. Finally, Hyb 4 corresponds to the execution with Simulator 3.

This completes the proof. □

E.3 Proof of Theorem 6.3

Proof of Theorem 6.3. Given a simulated commitment-proof $\vec{\mathbf{t}}^*$, $\pi^*(\vec{\mathbf{t}}_2^*, \vec{\mathbf{t}}_3^*, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}_\theta^*, \vec{v}_\theta^*)_{\theta=1}^{\ell}, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta^*, \vec{\mathbf{z}}_\theta^*)_{\theta=1}^{\ell})$, where the elements in blue represent the challenges output in the second and fourth moves, let $\vec{\mathbf{t}}, \pi := (\vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta, \vec{v}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}', (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta)_{\theta=1}^{\ell})$ be an adversary's response in the weak unique response experiment. If $(\vec{\mathbf{h}}, \vec{\mathbf{v}}') = (\vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**})$, then $(\mathbf{c}_\theta)_{\theta=1}^{\ell} = (\mathbf{c}_\theta^*)_{\theta=1}^{\ell}$, so there exists $i \in [\ell]$ such that $\vec{\mathbf{z}}_i \neq \vec{\mathbf{z}}_i^*$. Thus, by the verification algorithm, we obtain an MSIS solution with respect to $\mathbf{B}_0$. Now, if $(\vec{\mathbf{h}}, \vec{\mathbf{v}}') \neq (\vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**})$, we have $(\mathbf{c}_\theta)_{\theta=1}^{\ell} \neq (\mathbf{c}_\theta^*)_{\theta=1}^{\ell}$ except with probability $Q_{H_2}/|\mathcal{C}|^\ell$, where Q_{H_2} is an upper bound on the number of H_2-oracle queries performed by the adversary. Hence, there exists $j \in [\ell]$ such that $\mathbf{B}_0(\vec{\mathbf{z}}_j - \vec{\mathbf{z}}_j^*) = (\mathbf{c}_j - \mathbf{c}_j^*)\vec{\mathbf{t}}_0$. Since $\vec{\mathbf{t}}_0 \leftarrow\!\$\ R_q^n$ by Simulator 3, and $(\mathbf{c}_j - \mathbf{c}_j^*) \in R_q^\times$ except with probability hM, it follows that $(\mathbf{c}_j - \mathbf{c}_j^*)\vec{\mathbf{t}}_0 \neq \mathbf{0}^n$ except with probability at most $hM + q^{-dn}$, by the union bound. Therefore, $(\vec{\mathbf{z}}_j - \vec{\mathbf{z}}_j^*, \mathbf{c}_j^* - \mathbf{c}_j)$ is an MSIS solution of norm at most $2\sqrt{\beta^2 + d}$ with respect to $[\mathbf{B}_0 \mid \vec{\mathbf{t}}_0]$. □

E.4 Predicate Special Soundness of ENS

We now prove that the interactive version of Construction 3, obtained by replacing random oracle calls with random challenges, satisfies predicate-special-soundess. Recall that it is a 5-round protocol with $\mu = 2$ (see Remark B.1). Since the fifth round consists of zero-knowledge openings of a BDLOP commitment, three transcripts in $\mathscr{T}_3$ are needed to extract a weak binding opening. Moreover, because the third round contains messages that are computed based on the verifier's challenges, we query the prover twice to ensure that these challenges were not guessed. Thus, the extraction requires a $\vec{k} = (2, 3)$-tree of transcripts. Following Aardal et al. [AAB+24, Appendix I], we define the required predicate system in reverse order: first for the second level, which corresponds to a $(1, 3)$-tree, and then for the first level, which corresponds to a $(2, 3)$-tree.

Lemma E.1. *For any $\lambda \in \mathbb{N}$, let $\varepsilon = \mathsf{negl}(\lambda)$, $\mathfrak{s}_1, \mathfrak{s}_2 \geq \eta_\varepsilon(R^m)$, and $\beta = (d\mathfrak{s}_1 + \mathfrak{s}_2)\sqrt{md}$. Let M be an upper bound on the maximum probability of $\mathcal{C} \bmod (X^{d/h} - \zeta_j)$ for any $j \in [h]$ (see Lemma 2.6). Let $\mathbb{x} := (A, \vec{u}) \in \mathbb{Z}_q^{\nu \times d} \times \mathbb{Z}_q^{\nu}$ be any valid statement of interest. Consider the following predicate system $\vec{\mathfrak{C}}$:*

- ***Level*** *2: Recall that $k_2 = 3$ (see Definition 3.3).*
  - ***Challenge predicates****: Here, we check the conditions necessary to extract a weak opening.*
    - $\mathfrak{C}_{2,1}^{\mathsf{chal}}((\mathbf{c}_\theta^{(1)})_{\theta=1}^{\ell}) = 1$ *for any* $(\mathbf{c}_\theta^{(1)})_{\theta=1}^{\ell} \leftarrow\!\$\, \mathcal{C}^\ell$.
    - $\mathfrak{C}_{2,j}^{\mathsf{chal}}((\mathbf{c}_\theta^{(1)})_{\theta=1}^{\ell}, \ldots, (\mathbf{c}_\theta^{(j)})_{\theta=1}^{\ell}) = 1$ *if and only if, for* $j \in \{2,3\}$ *there exists* $v \in [\ell]$ *such that* $\mathbf{c}_v^{(1)} - \mathbf{c}_v^{(j)} \in R_q^\times$.
  - ***Commitment predicates****:*
    - $\mathfrak{C}_2^{\mathsf{prop}}(t_1, t_2) = 1$*, with* $t_i \in \mathscr{T}_3$*, if and only if the extracted witness* $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*)$ *is a weak opening, satisfies* $\mathsf{NTT}(\mathbf{s}^*) \in \{0, \pm 1\}^d$ *and* $\mathsf{ct}(\vec{\mathbf{x}} + \vec{\mathbf{g}}^* - \vec{\mathbf{h}}) = 0^\ell$*, where* $\vec{\mathbf{h}}$ *is one of the elements in the prover's second message shared by* t_1, t_2*,* $\vec{\mathbf{x}} = [\boldsymbol{\psi}_1 \mathbf{s}^* \mid \cdots \mid \boldsymbol{\psi}_\ell \mathbf{s}^*]^\top$*, and* $\boldsymbol{\psi}_\theta = \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta)$ *with* $(\vec{v}_\theta)_{\theta=1}^{\ell}$ *corresponding to the level-1 challenge of* t_1 *and* t_2.
    - $\mathfrak{C}_2^{\mathsf{bind}}(t_1, t_2, t_3) = 1$*, with* $t_i \in \mathscr{T}_3$*, if and only if, given two extracted witnesses* $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*)$*, extracted from* t_1 *and* t_2*, and* $(\bar{\mathbf{c}}', \vec{\mathbf{r}}^{*\prime}, \mathbf{s}^{*\prime}, \vec{\mathbf{g}}^{*\prime}, \vec{\mathbf{m}}^{*\prime})$*, extracted from* t_1 *and* t_3*, we have* $(\vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*) = (\vec{\mathbf{r}}^{*\prime}, \mathbf{s}^{*\prime}, \vec{\mathbf{g}}^{*\prime}, \vec{\mathbf{m}}^{*\prime})$.
- ***Level*** *1: Recall that $k_1 = 2$ (see Definition 3.3).*
  - ***Commitment predicates****:*
    - $\mathfrak{C}_1^{\mathsf{prop}}(t_1) = 1$*, with* $t_1 \in \mathscr{T}_2$*, if and only if, given the extracted witness* $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*)$*, we have* $A \cdot \mathsf{NTT}(\mathbf{s}^*) = \vec{u}$.
    - $\mathfrak{C}_1^{\mathsf{bind}}(t_1, t_2) = 1$*, with* $t_i \in \mathscr{T}_2$*, under the same conditions as* $\mathfrak{C}_2^{\mathsf{bind}}$.

Then, at level 2, $\mathfrak{C}_{2,1}^{\mathsf{chal}}$ has failure density $p_{2,1}^{\mathsf{chal}} = 0$, $\mathfrak{C}_{2,j}^{\mathsf{chal}}$ has failure density $p_{2,j}^{\mathsf{chal}} = (hM)^\ell$, and $(\mathfrak{C}_2^{\mathsf{prop}}, \mathfrak{C}_2^{\mathsf{bind}})$ has failure density $p_2^{\mathsf{com}} = (3M)^\ell + M$. At level 1, $(\mathfrak{C}_1^{\mathsf{prop}}, \mathfrak{C}_1^{\mathsf{bind}})$ has failure density $p_1^{\mathsf{com}} = q^{-\ell}$. Furthermore, under the hardness of $\mathsf{MSIS}_{n,m,q,8d\beta}$, the interactive version of Construction 3 is $(\vec{k}, \vec{\mathfrak{C}})$ predicate-special-sound, with $\vec{k} = (2, 3)$.

Proof of Lemma E.1. First, we analyze level 2. At this level, the prover receives a tuple of challenges $(\mathbf{c}_\theta)_{\theta=1}^{\ell} \leftarrow\!\$\, \mathcal{C}^\ell$ and outputs a tuple $(\vec{\mathbf{z}}_\theta)_{\theta=1}^{\ell}$ such that, for each $i \in [\ell]$, $\mathbf{B}_0 \vec{\mathbf{z}}_\theta = \vec{\mathbf{w}}_\theta + \mathbf{c}_\theta \vec{\mathbf{t}}_0$, where $\mathbf{B}_0 \in R_q^{n \times m}$ is part of the public parameters, and $\vec{\mathbf{t}}_0, \vec{\mathbf{w}}_\theta \in R_q^n$ are contained in the prover's first message. Conditioned on the BDLOP commitment being binding, an argument similar to that used for the challenge predicate of Construction 1 (see the proof of Lemma D.1), but applied to $\mathcal{C}^\ell$, gives $p_{2,2}^{\mathsf{chal}} = p_{2,3}^{\mathsf{chal}} = (hM)^\ell$. Moreover, $p_{2,1}^{\mathsf{chal}} = 0$, since no condition is imposed on the first challenge.

We now bound p_2^{com}. This quantity upper-bounds the fraction of challenges $(\mathbf{c}_\theta^{(3)})_{\theta=1}^{\ell} \in \mathcal{C}^\ell \setminus \{(\mathbf{c}_\theta^{(1)})_{\theta=1}^{\ell}, (\mathbf{c}_\theta^{(2)})_{\theta=1}^{\ell}\}$ for which there exists a valid third transcript such that, for the extracted witness $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*)$ (see Appendix A.2 for a sketch of the extraction procedure), either $\mathsf{NTT}(\mathbf{s}^*) \notin \{0, \pm 1\}^d$ or $\mathsf{ct}(\vec{\mathbf{x}} + \vec{\mathbf{g}}^* - \vec{\mathbf{h}}) \neq 0^\ell$, where $\vec{\mathbf{h}}$ is one of the elements in the prover's second message shared by $t_1, t_2, t_3 \in \mathscr{T}_3$, $\vec{\mathbf{x}} = [\boldsymbol{\psi}_1 \mathbf{s}^* \mid \cdots \mid \boldsymbol{\psi}_\ell \mathbf{s}^*]^\top$, and $\boldsymbol{\psi}_\theta = \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta)$ with $(\vec{v}_\theta)_{\theta=1}^{\ell}$ corresponding to level-1 challenge. Since we only need an upper bound, we may ignore the exclusion of the first and second challenge vectors and instead bound the fraction of all $(\mathbf{c}_i^{(3)})_{\theta=1}^{\ell} \in \mathcal{C}^\ell$ for which that condition holds. Conditioned on the BDLOP commitment being binding, $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*)$ is a weak opening, and we have $\mathbf{B}_1 \vec{\mathbf{r}}^* + [\mathbf{s}^* \mid \vec{\mathbf{g}}^{*\top} \mid \vec{\mathbf{m}}^{*\top}]^\top = [\mathbf{t}_1 \mid \vec{\mathbf{t}}_2 \mid \vec{\mathbf{t}}_3]^\top$, where $\mathbf{t}_1 \in R_q, \vec{\mathbf{t}}_2 \in R_q^\ell, \vec{\mathbf{t}}_3 \in R_q^{2\ell}$ are components

of the prover's first message shared by the transcripts t_1, t_2, t_3, and $\mathbf{B}_1 \in R_q^{(1+3\ell)\times m}$ is part of the public parameters. Moreover, let $\vec{\mathbf{y}}_\theta^* \in R_q^m$, for $\theta \in [\ell]$, be the unique values computed from $\vec{\mathbf{r}}^*$ and the tuple $(\mathbf{c}_\theta^{(3)}, \vec{\mathbf{z}}_\theta)$. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$ and $[\mathbf{m}_1^* \mid \cdots \mid \mathbf{m}_{2\ell}^*] := \vec{\mathbf{m}}^*$.

First, suppose that $\mathsf{NTT}(\mathbf{s}^*) \notin \{0, \pm 1\}^d$. By Step 5 of the verification procedure of Construction 3, for each $\theta \in [\ell]$, we have

$$\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta^* \rangle^3 + \langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_\theta^* \rangle - \mathbf{v}_\theta - \mathbf{c}_\theta^{(3)} \left(3\mathbf{s}^* \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta^* \rangle^2 - \langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_\theta^* \rangle + \mathbf{m}_{2\theta-1}^* \right) + \left(\mathbf{c}_\theta^{(3)} \right)^2 \left((3\mathbf{s}^* - 1) \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_\theta^* \rangle - \mathbf{m}_{2\theta}^* \right) - \left(\mathbf{c}_\theta^{(3)} \right)^3 (\mathbf{s}^*(\mathbf{s}^* - 1)(\mathbf{s}^* + 1)) = 0$$

where $\mathbf{v}_\theta$ are component of a vector $\vec{\mathbf{v}} \in R_q^\ell$ that is part of the prover's first message. Since $\mathsf{NTT}(\mathbf{s}^*) \notin \{0, \pm 1\}^d$, $\mathbf{s}^*(\mathbf{s}^* - 1)(\mathbf{s}^* + 1) \neq 0$. Therefore, there exists a prime ideal $(q, X^{d/h} - \zeta_k)$, with $k \in [h]$, over which the above expression is a non-zero polynomial of degree 3. Hence, by applying an argument similar to the one used for the commitment predicate of Construction 1 (see the proof of Lemma D.1), the probability that Step 5 holds for all $\theta \in [\ell]$ is at most $(3M)^\ell$.

It remains to consider the case $\mathsf{ct}(\vec{\mathbf{x}} + \vec{\mathbf{g}}^* - \vec{\mathbf{h}}) \neq 0^\ell$. By Step 3 of the verification procedure of Construction 3, we have

$$\begin{bmatrix} \langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{y}}_1^* \rangle \\ \vdots \\ \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{y}}_\ell^* \rangle \end{bmatrix} - \vec{\mathbf{v}}' = \begin{bmatrix} \mathbf{c}_1^{(3)} \\ \vdots \\ \mathbf{c}_\ell^{(3)} \end{bmatrix} \odot \left(\vec{\mathbf{g}}^* + \vec{\mathbf{x}} - \vec{\mathbf{h}} \right)$$

where $\vec{\mathbf{v}}'$ is a component of the prover's second message computed after receiving $(\vec{v}_\theta)_{\theta=1}^\ell$. Hence, if $\mathsf{ct}(\vec{\mathbf{x}} + \vec{\mathbf{g}}^* - \vec{\mathbf{h}}) \neq 0^\ell$, there exists $j \in [\ell]$ and a prime ideal $(q, X^{d/h} - \zeta_r)$, with $r \in [h]$, such that $\mathbf{c}_j^{(3)} \equiv (\mathbf{g}_j^* + \mathbf{x}_j^* - \mathbf{h}_j^*)^{-1} \cdot (\langle \boldsymbol{\psi}_j \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{j+1}, \vec{\mathbf{y}}_j^* \rangle - \mathbf{v}_j') \mod (X^{d/h} - \zeta_r)$. However, by Lemma 2.6, this happens with probability at most M. Therefore, by a union bound, we have $p_2^{\mathsf{com}} = (3M)^\ell + M$.

Finally, we analyze level 1. Let $(\bar{\mathbf{c}}, \vec{\mathbf{r}}^*, \mathbf{s}^*, \vec{\mathbf{g}}^*, \vec{\mathbf{m}}^*)$ be the witness extracted from the level 2 of a tree $t_1 \in \mathscr{T}_2$, and let $(\vec{v}_\theta^{(1)})_{\theta=1}^\ell \in (\mathbb{Z}_q^\nu)^\ell$ be the corresponding level-1 challenge vector. Still conditioning on the BDLOP commitment being binding, p_1^{com} upper-bounds the fraction of challenges $(\vec{v}_\theta^{(2)})_{\theta=1}^\ell \leftarrow\!\!\$\ (\mathbb{Z}_q^\nu)^\ell \setminus \{(\vec{v}_\theta^{(1)})_{\theta=1}^\ell\}$ for which there exists a tree $t_2 \in \mathscr{T}_2$ sharing the same prover's first message with t_1, such that t_2 contains valid transcripts but $A \cdot \mathsf{NTT}(\mathbf{s}^*) \neq \vec{u}$. Again, since we only need an upper bound, we ignore the exclusion of the first challenge vector and instead bound the fraction of all $(\vec{v}_\theta^{(2)})_{\theta=1}^\ell \leftarrow\!\!\$\ (\mathbb{Z}_q^\nu)^\ell$ for which the above failure conditions hold. This time, letting $\vec{\mathbf{x}} = [\boldsymbol{\psi}_1 \mathbf{s}^* \mid \cdots \mid \boldsymbol{\psi}_\ell \mathbf{s}^*]^\top \in R_q^\ell$, where $\boldsymbol{\psi}_\theta = \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta^{(2)})$, we have $\mathsf{ct}(\vec{\mathbf{x}}) = [\langle A \cdot \mathsf{NTT}(\mathbf{s}^*), \vec{v}_1^{(2)} \rangle \mid \cdots \mid \langle A \cdot \mathsf{NTT}(\mathbf{s}^*), \vec{v}_\ell^{(2)} \rangle]^\top \in \mathbb{Z}_q^\ell$. Let $\vec{\mathbf{h}} \in R_q^\ell$ be one of the components of the prover's second message computed after receiving $(\vec{v}_\theta^{(2)})_{\theta=1}^\ell$. Since t_2 contains valid transcripts, Step 3 of the verification procedure of Construction 3 implies that $\mathsf{ct}(\vec{\mathbf{g}}^* + \vec{\mathbf{x}} - \vec{\mathbf{h}}) = \mathsf{ct}(\vec{\mathbf{g}}^* + \vec{\mathbf{x}}) - [\langle \vec{u}, \vec{v}_1^{(2)} \rangle \mid \cdots \mid \langle \vec{u}, \vec{v}_\ell^{(2)} \rangle]^\top$. Since $A \cdot \mathsf{NTT}(\mathbf{s}^*) \neq \vec{u}$, $\mathsf{ct}(\vec{\mathbf{x}}) = [\langle \vec{u}, \vec{v}_1^{(2)} \rangle \mid \cdots \mid \langle \vec{u}, \vec{v}_\ell^{(2)} \rangle]^\top$ with probability $q^{-\ell}$. In this case, if the prover sampled $\vec{\mathbf{g}}^*$ as in Construction 3, it will be able to generate an accepting transcript. Now, conditioned on $\mathsf{ct}(\vec{\mathbf{x}}) \neq [\langle \vec{u}, \vec{v}_1^{(2)} \rangle \mid \cdots \mid \langle \vec{u}, \vec{v}_\ell^{(2)} \rangle]^\top$, it follows that $\mathsf{ct}(\vec{\mathbf{g}}^* + \vec{\mathbf{x}} - \vec{\mathbf{h}}) \neq 0^\ell$ with probability at least $1 - q^{-\ell}$ since the challenge vectors $\vec{v}_\theta^{(2)}$ are sampled independently of $\vec{\mathbf{g}}^*$. But then $\mathfrak{C}_2^{\mathsf{prop}}$ would fail, contradicting the requirement that $\mathfrak{C}_2^{\mathsf{prop}}$ be satisfied (see Definition 3.5). Hence, $p_1^{\mathsf{com}} = q^{-\ell}$.

Finally, if the BDLOP commitment is not binding, then, by an argument similar to the one used in the proof of Lemma D.1, we can recover an $\mathsf{MSIS}_{n,m,q,8d\beta}$'s solution with respect to the matrix $\mathbf{B}_0$. □

E.5 Proof of Theorem 6.5

Proof of Theorem 6.5. Let $\mathsf{Sim}_{\Pi_{\mathsf{tHE}}} = (\mathsf{SimKGen}, \mathsf{SimTDec})$ be a simulator for Π_{tHE}, which is guaranteed to exist by its simulation security, $\mathsf{Sim}_{\mathsf{vENS}} = (\mathsf{SimSetup}, \mathsf{SimProve}, \mathsf{SimRO})$ be the simulator for Construction 3 (see Simulator 3). Now, we are ready to describe the simulator.

Simulator 4: Simulator of threshold vENS

The simulator initializes the random tape of the adversary with a uniformly random string. In addition, it tracks and answers random oracle queries.

- $\mathsf{SimSetup}(1^\lambda, \mathtt{n}, \mathtt{t})$: Given $\lambda \in \mathbb{N}$, proceed as follows:
  1. Compute $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathsf{Sim}_{\mathsf{vENS}}.\mathsf{SimSetup}(1^\lambda)$.
  2. Compute $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}, \mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}) \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimKGen}(1^\lambda, \mathtt{n}, \mathtt{t})$.
  3. Initialize an empty state $\mathsf{st}_{\mathsf{Sim}}$ and store $\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}$.
  4. Return $\mathsf{pp} = (\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1)$ and $\mathsf{paux}_i = \mathsf{sk}_i$ to each prover P_i.

- $\mathsf{SimWitShare}(\mathsf{pp}, \mathbb{x})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, and $(A, \vec{u}) := \mathbb{x}$ do the following:
  1. Compute $\mathsf{ctx}_{\mathtt{s}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$ and $\{[\![\mathbf{s}]\!]_i\}_{i\in[\mathtt{n}]} \leftarrow \Pi_{\mathsf{SS}}.\mathsf{Share}(\mathbf{0}, (\mathtt{n}, \mathtt{t}))$.
  2. Set $\mathsf{aux}_w := (\mathsf{ctx}_{\mathtt{s}})$.
  3. For $i \in [\mathtt{n}]$, set $\mathsf{pwit}_i := [\![\mathbf{s}]\!]_i$ and send $(\mathsf{pwit}_i, \mathsf{aux}_w)$ to prover P_i.

- $\mathsf{SimCom}(\mathsf{pp}, \mathbb{x}, \mathcal{S}, \mathcal{V}, \{(\mathsf{paux}_k, \mathsf{pwit}_k)\}_{k\in\mathcal{V}}, \mathsf{aux}_w, \mathsf{st}_{\mathsf{Sim}})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(A, \vec{u}) := \mathbb{x}$, the total set of participating provers $\mathcal{S}$, a set of corrupted parties $\mathcal{V}$ with their initial inputs $(\{(\mathsf{paux}_k, \mathsf{pwit}_k)\}_{k\in\mathcal{V}}, \mathsf{aux}_w)$, and a simulator state $\mathsf{st}_{\mathsf{Sim}}$, proceed as follows:
  1. If $\mathcal{S} \not\subseteq [\mathtt{n}]$ or $\mathcal{V} \not\subset \mathcal{S}$ or $|\mathcal{V}| \geq \mathtt{t}$: abort.
  2. Let $\mathcal{H} \leftarrow \mathcal{S} \setminus \mathcal{V}$. (*this is the set of honest provers*)
  3. Run $\mathsf{Sim}_{\mathsf{vENS}}.\mathsf{SimProve}((\mathbf{B}_0, \mathbf{B}_1), \mathbb{x}; \mathfrak{s}_1\sqrt{|\mathcal{H}|}, \mathfrak{s}_2\sqrt{|\mathcal{H}|})$ with parameters corresponding to a proof generated by provers in $\mathcal{H}$ to obtain $(\vec{\mathbf{t}}^*, \pi := (\vec{\mathbf{t}}_2^*, \vec{\mathbf{t}}_3^*, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}_\theta^*, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta^*)_{\theta=1}^\ell))$, i.e., during the execution of $\mathsf{Sim}_{\mathsf{vENS}}.\mathsf{SimProve}$, use $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to sample each $\vec{\mathbf{z}}_\theta^*$.
  4. Parse $[(\vec{\mathbf{t}}_0^*)^\top \mid \mathbf{t}_1^*]^\top := \vec{\mathbf{t}}^*$.
  5. Let $j \leftarrow\!\!\$\ \mathcal{H}$. (*this will be the last honest prover to reply*).
  6. ▶ Upon receiving $\{[\vec{\mathbf{t}}_{0,k}^\top \mid \mathbf{t}_{1,k}]^\top\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
     (a) For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $[\vec{\mathbf{t}}_{0,i}^\top \mid \mathbf{t}_{1,i}]^\top \leftarrow\!\!\$\ R_q^{n+1}$ and return that as its output.
     (b) For P_j, compute $\vec{\mathbf{t}}_{0,j} \leftarrow \vec{\mathbf{t}}_0^* - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{t}}_{0,i}$ and $\mathbf{t}_{1,j} \leftarrow \mathbf{t}_1^* - \sum_{i\in\mathcal{S}\setminus\{j\}} \mathbf{t}_{1,i}$. Then, return $[\vec{\mathbf{t}}_{0,j}^\top \mid \mathbf{t}_{1,j}]^\top$ as its output.
  7. Compute $\vec{\mathbf{t}} \leftarrow [(\vec{\mathbf{t}}_0^* + \sum_{i\in\mathcal{V}} \vec{\mathbf{t}}_{0,i})^\top \mid \mathbf{t}_1^*]^\top$.
  8. Store $(\mathcal{S}, \mathcal{V}, \mathbb{x}, \vec{\mathbf{t}}, j, (\vec{\mathbf{t}}^*, \pi := (\vec{\mathbf{t}}_2^*, \vec{\mathbf{t}}_3^*, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}_\theta^*, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta^*)_{\theta=1}^\ell)))$ in $\mathsf{st}_{\mathsf{Sim}}$.

- $\mathsf{SimProve}(\mathsf{pp}, \mathbb{x}, \mathcal{S}, \mathcal{V}, \vec{\mathbf{t}}, \mathsf{st}_{\mathsf{Sim}})$: Given $(\mathsf{pk}, \mathbf{B}_0, \mathbf{B}_1) := \mathsf{pp}$, $(A, \vec{u}) := \mathbb{x}$, the total set of participating provers $\mathcal{S}$, a set of corrupted provers $\mathcal{V}$, a commitment $\vec{\mathbf{t}}$, and a simulator state $\mathsf{st}_{\mathsf{Sim}}$, proceed as follows:
  1. Use $(\mathcal{S}, \mathcal{V}, \mathbb{x}, \vec{\mathbf{t}})$ to recover the tuple $(\mathcal{S}, \mathcal{V}, \mathbb{x}, \vec{\mathbf{t}}, j, (\vec{\mathbf{t}}^*, \pi := (\vec{\mathbf{t}}_2^*, \vec{\mathbf{t}}_3^*, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}_\theta^*, \vec{v}_\theta)_{\theta=1}^\ell, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta^*)_{\theta=1}^\ell)))$ from $\mathsf{st}_{\mathsf{Sim}}$. If no such tuple exists, abort.
  2. Parse $[\vec{\mathbf{t}}_0 \mid \mathbf{t}_1] := \vec{\mathbf{t}}$ and $[(\vec{\mathbf{t}}_0^*)^\top \mid \mathbf{t}_1^*]^\top := \vec{\mathbf{t}}^*$. ($\mathbf{t}_1 = \mathbf{t}_1^*$ *by construction.*)
  3. For $k \in \mathcal{V}$, use the adversary's random tape to sample $\vec{\mathbf{r}}_k \leftarrow D_{R,\mathfrak{s}_1}^m$, and for $\theta \in [\ell]$, $\vec{\mathbf{y}}_{\theta,k} \leftarrow D_{R,\mathfrak{s}_2}^m$. Then, compute the adversary's aggregate contributions $\vec{\mathbf{r}}_\mathcal{V} \leftarrow \sum_{k\in\mathcal{V}} \vec{\mathbf{r}}_k$ and, for $\theta \in [\ell]$, $\vec{\mathbf{y}}_{\theta,\mathcal{V}} \leftarrow \sum_{k\in\mathcal{V}} \vec{\mathbf{y}}_{\theta,k}$. Finally, from these, compute $\vec{\mathbf{w}}_{\theta,\mathcal{V}} \leftarrow \mathbf{B}_0 \cdot \vec{\mathbf{y}}_{\theta,\mathcal{V}}$ and $\vec{\mathbf{z}}_{\theta,\mathcal{V}} \leftarrow \vec{\mathbf{y}}_{\theta,\mathcal{V}} + \mathbf{c}_\theta \cdot \vec{\mathbf{r}}_\mathcal{V}$.

4. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
5. For $\theta \in [\ell]$, compute $\boldsymbol{\psi}_\theta \leftarrow \mathsf{NTT}^{-1}(dA^\top \vec{v}_\theta)$.
6. Parse $[\mathbf{t}^*_{2+\ell} \mid \cdots \mid \mathbf{t}^*_{3\ell+1}]^\top := \vec{\mathbf{t}}^*_3$.
7. For $\theta \in [\ell]$, do the following:
   (a) Let $\tilde{\mathbf{B}}_\theta \leftarrow [\vec{\mathbf{b}}_1 \mid \vec{\mathbf{b}}_{\ell+2\theta} \mid \vec{\mathbf{b}}_{\ell+2\theta+1}]^\top$.
   (b) Let $\vec{\mathbf{f}}_\theta \leftarrow \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}^*_\theta - \mathbf{c}_\theta \cdot [\mathbf{t}^*_1 \mid \mathbf{t}^*_{\ell+2\theta} \mid \mathbf{t}^*_{\ell+2\theta+1}]^\top$.
   (c) Compute

$$\vec{\boldsymbol{\omega}}_{\theta,\mathcal{V}} \leftarrow \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \otimes (\vec{\mathbf{f}}_\theta + \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}) \otimes (\vec{\mathbf{f}}_\theta - \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}),$$
$$\Delta_{\vec{\mathbf{v}},\theta} \leftarrow \langle \vec{\mathbb{e}}_1, \vec{\mathbf{f}}_\theta \otimes \Big((\vec{\mathbf{f}}_\theta + \mathbf{c}_\theta \cdot \mathbf{1}^3) \otimes \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \otimes (\vec{\mathbf{f}}_\theta - \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}) \Big) \rangle$$
$$+ \langle \vec{\mathbb{e}}_1, \vec{\boldsymbol{\omega}}_{\theta,\mathcal{V}} \rangle + \langle \vec{\mathbf{e}}_2, \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \rangle + \mathbf{c}_\theta \cdot \langle \vec{\mathbf{e}}_3, \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \rangle.$$

   where $\vec{\mathbb{e}}_1$ is the first standard basis vector in R_q^{27}, and $\vec{\mathbf{e}}_2$ and $\vec{\mathbf{e}}_3$ are the second and third standard basis vectors in R_q^3.
8. Set $\vec{\Delta}_{\vec{\mathbf{v}}} := [\Delta_{\vec{\mathbf{v}},1} \mid \cdots \mid \Delta_{\vec{\mathbf{v}},\ell}]^\top$.
9. Set $\vec{\mathbf{v}} \leftarrow \vec{\mathbf{v}}^* + \vec{\Delta}_{\vec{\mathbf{v}}}$.
10. Retrieve $\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}$ from $\mathsf{st}_{\mathsf{Sim}}$.
11. Compute $\vec{\mathbf{t}}_2 \leftarrow \vec{\mathbf{t}}^*_2 + [\langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{r}}_\mathcal{V} \rangle \mid \cdots \mid \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{\ell+1}, \vec{\mathbf{r}}_\mathcal{V} \rangle]^\top$.
12. Parse $[\mathbf{t}_{\mathbf{g}_1} \mid \cdots \mid \mathbf{t}_{\mathbf{g}_\ell}]^\top := \vec{\mathbf{t}}_2$
13. ▶ If $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}^*_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}^*_\theta + \vec{\mathbf{w}}_{\theta,\mathcal{V}})_{\theta=1}^\ell) \neq \perp$, abort. Else, set $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}^*_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}^*_\theta + \vec{\mathbf{w}}_{\theta,\mathcal{V}})_{\theta=1}^\ell) \leftarrow (\vec{v}_\theta)_{\theta=1}^\ell$.
14. Compute $\vec{\mathbf{v}}'_\mathcal{V} \leftarrow \begin{bmatrix} \langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \vec{\mathbf{y}}_{1,\mathcal{V}} \rangle \\ \vdots \\ \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \vec{\mathbf{y}}_{\ell,\mathcal{V}} \rangle \end{bmatrix}$
15. ▶ If $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}^*_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}^*_\theta + \vec{\mathbf{w}}_{\theta,\mathcal{V}})_{\theta=1}^\ell, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**} + \vec{\mathbf{v}}'_\mathcal{V}) \neq \perp$, abort. Else, set $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}^*_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}^*_\theta + \vec{\mathbf{w}}_{\theta,\mathcal{V}})_{\theta=1}^\ell, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**} + \vec{\mathbf{v}}'_\mathcal{V}) \leftarrow (\mathbf{c}_\theta)_{\theta=1}^\ell$.
16. ▶ Upon receiving $\{\big(\mathsf{ctx}_{\vec{\mathbf{r}}_k}, (\mathbf{t}_{\mathbf{g}_{\theta,k}}, \vec{\mathbf{w}}_{\theta,k}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,k}}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,k} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,k} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,k} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_k \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_k \rangle})_{\theta=1}^\ell\big)\}_{k \in \mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
   (a) For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, compute $\mathsf{ctx}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$ and then for each $\theta \in [\ell]$, do the following:
      – Sample $\vec{\mathbf{w}}_{\theta,i} \leftarrow\!\!\$\ R_q^n$ and $\mathbf{t}_{\mathbf{g}_{\theta,i}} \leftarrow\!\!\$\ R_q$.
      – Compute $\mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$.
      – Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i} \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
      – Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,i} \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
      – Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,i} \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
      – Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_i \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
      – Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_i \rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.

      Output $\big(\mathsf{ctx}_{\vec{\mathbf{r}}_i}, (\mathbf{t}_{\mathbf{g}_{\theta,i}}, \vec{\mathbf{w}}_{\theta,i}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,i}}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,i} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,i} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_i \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_i \rangle})_{\theta=1}^\ell\big)$ as P_i's answer.
   (b) For P_j, proceed like the other honest provers. However, instead of sampling $\vec{\mathbf{w}}_{\theta,j}$ and $\mathbf{t}_{\mathbf{g}_{\theta,j}}$ uniformly at random, compute $\vec{\mathbf{w}}_{\theta,j} \leftarrow \vec{\mathbf{w}}^*_\theta - \sum_{i \in \mathcal{H} \setminus \{j\}} \vec{\mathbf{w}}_{\theta,i}$ and $\mathbf{t}_{\mathbf{g}_{\theta,j}} \leftarrow \mathbf{t}_{\mathbf{g}_\theta} - \sum_{i \in \mathcal{S} \setminus \{j\}} \mathbf{t}_{\mathbf{g}_{\theta,i}}$, with $\theta \in [\ell]$. Return $\big(\mathsf{ctx}_{\vec{\mathbf{r}}_j}, (\mathbf{t}_{\mathbf{g}_{\theta,j}}, \vec{\mathbf{w}}_{\theta,j}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,j}}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,j} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,j} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,j} \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_j \rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_j \rangle})_{\theta=1}^\ell\big)$ as its output.

17. ▶ Upon receiving $\{(\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,k}}, \mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,k}}, \mathsf{ds}_{\mathbf{v}_{\theta,k}})_{\theta=1}^{\ell}\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
    (a) Parse $[\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell]^\top := \vec{\mathbf{v}}$.
    (Aggregate the ciphertexts)
    (b) Compute $\mathsf{ctx}_{\vec{\mathbf{r}}} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\vec{\mathbf{r}}_k}$.
    (c) For $\theta \in [\ell]$, do the following:
        – Compute $\mathsf{ctx}_{\vec{\mathbf{y}}_\theta} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,k}}$.
        – Compute $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}_\theta\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}_{\theta,k}\rangle}$.
        – Compute $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{y}}_\theta\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{y}}_{\theta,k}\rangle}$.
        – Compute $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{y}}_\theta\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{y}}_{\theta,k}\rangle}$.
        – Compute $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{r}}\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{r}}_k\rangle}$.
        – Compute $\mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{r}}\rangle} \leftarrow \sum_{k\in\mathcal{S}} \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{r}}_k\rangle}$.
    (Derive the ciphertexts used to finalize the ternary proof)
    (d) For $\theta \in [\ell]$, do the following:
        – Compute $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{r}}\rangle} + \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{y}}_\theta\rangle} - 3\cdot\mathsf{ctx}_{\mathbf{s}}\cdot(\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}_\theta\rangle})^2$.
        – Compute $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta+1},\vec{\mathbf{r}}\rangle} + (3\cdot(\mathsf{ctx}_{\mathbf{s}})^2 - 1)\cdot\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}_\theta\rangle}$.
        – Compute $\mathsf{ctx}_{\mathbf{v}_\theta} \leftarrow \mathsf{ctx}_{\langle\vec{\mathbf{b}}_{\ell+2\theta},\vec{\mathbf{y}}_\theta\rangle} + (\mathsf{ctx}_{\langle\vec{\mathbf{b}}_1,\vec{\mathbf{y}}_\theta\rangle})^3$.
    (Simulate the decryption shares for honest provers)
    (e) For each prover P_i with $i \in \mathcal{H}$, and for each $\theta \in [\ell]$, do the following:
        – Compute $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}}, \mathbf{t}^*_{\ell+2\theta}, \mathcal{S}, \mathcal{V}, i)$.
        – Compute $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}}, \mathbf{t}^*_{\ell+2\theta+1}, \mathcal{S}, \mathcal{V}, i)$.
        – Compute $\mathsf{ds}_{\mathbf{v}_{\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{v}_\theta}, \mathbf{v}_\theta, \mathcal{S}, \mathcal{V}, i)$.
    Output $(\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,i}}, \mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,i}}, \mathsf{ds}_{\mathbf{v}_{\theta,i}})_{\theta=1}^{\ell}$ as P_i's answer.

18. ▶ Upon receiving $(\vec{\mathbf{h}}_k, \vec{\mathbf{v}}'_k)_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
    (a) From $\vec{\mathbf{h}}^*$ (recovered from $\mathsf{st}_{\mathsf{Sim}}$) and the corrupted contributions $(\vec{\mathbf{h}}_k)_{k\in\mathcal{V}}$, form $\mathcal{T}_{\mathsf{ct}} := \{\mathsf{ct}(\vec{\mathbf{h}}^*)\} \cup \{\lambda_k^{-1}\cdot\mathsf{ct}(\vec{\mathbf{h}}_k)\}_{k\in\mathcal{V}} \subset \mathbb{Z}_q^\ell$, where $\{\lambda_k\}_{k\in\mathcal{V}} \subset \mathbb{Z}_q^\times$ are the Lagrange coefficients of the corrupted provers with respect to $\mathcal{S}$; the factor λ_k^{-1} recovers each corrupted prover's underlying constant-term share, so that $\mathcal{T}_{\mathsf{ct}}$ is a Shamir sharing with secret $\mathsf{ct}(\vec{\mathbf{h}}^*)$.
    (b) If $|\mathcal{T}_{\mathsf{ct}}| < \mathtt{t}$, pad the set as follows:
        – Sample $\mathtt{t} - |\mathcal{T}_{\mathsf{ct}}|$ fresh uniform vectors $\vec{s}_i \leftarrow\!\!\$\ \mathbb{Z}_q^d$.
        – For each, compute $\vec{u}_i \leftarrow A\vec{s}_i \in \mathbb{Z}_q^\nu$ and add the vector $(\langle\vec{u}_i,\vec{v}_1\rangle, \ldots, \langle\vec{u}_i,\vec{v}_\ell\rangle)^\top \in \mathbb{Z}_q^\ell$ to $\mathcal{T}_{\mathsf{ct}}$.
    (c) Interpret $\mathcal{T}_{\mathsf{ct}}$ as a Shamir sharing with secret $\mathsf{ct}(\vec{\mathbf{h}}^*)$ (its value at evaluation point 0), and let ϕ be its Lagrange interpolant. Then, for each P_i, with $i \in \mathcal{H}$, compute the constant term $\mathsf{ct}(\vec{\mathbf{h}}_i) \leftarrow \lambda_i\cdot\phi(i)$, which corresponds to the evaluation $\phi(i)$ scaled by P_i's Lagrange coefficient $\lambda_i \in \mathbb{Z}_q^\times$ (with respect to $\mathcal{S}$), rather than $\phi(i)$ itself.
    (d) For all provers P_i, with $i \in \mathcal{H}\setminus\{j\}$, do the following:
        – Sample $\vec{\mathbf{v}}'_i \leftarrow\!\!\$\ R_q^\ell$ and $\overline{\mathsf{ct}}(\vec{\mathbf{h}}_i) \leftarrow\!\!\$\ \{\vec{\mathbf{a}} \in R_q^\ell : \mathsf{ct}(\vec{\mathbf{a}}) = 0^\ell\}$.
        – Set $\vec{\mathbf{h}}_i \leftarrow \mathsf{ct}(\vec{\mathbf{h}}_i) + \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i)$, *where* $\mathsf{ct}(\vec{\mathbf{h}}_i)$ *was computed in Step 18c.*
        – Output $(\vec{\mathbf{h}}_i, \vec{\mathbf{v}}'_i)$ as its contribution.
    (e) For P_j, do the following:
        – Compute $\overline{\mathsf{ct}}(\vec{\mathbf{h}}_j) \leftarrow \overline{\mathsf{ct}}(\vec{\mathbf{h}}^*) - \sum_{i\in\mathcal{S}\setminus\{j\}} \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i)$.
        – Set $\vec{\mathbf{h}}_j \leftarrow \mathsf{ct}(\vec{\mathbf{h}}_j) + \overline{\mathsf{ct}}(\vec{\mathbf{h}}_j)$, *where* $\mathsf{ct}(\vec{\mathbf{h}}_j)$ *was computed in Step 18c.*
        – Compute $\vec{\mathbf{v}}'_j \leftarrow \vec{\mathbf{v}}^{**} - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{v}}'_i$
        – Output $(\vec{\mathbf{h}}_j, \vec{\mathbf{v}}'_j)$ as its contribution.

19. ▶ Upon receiving $\{(\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,k}})_{\theta=1}^{\ell}\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, do the following:
    (a) For each $\theta \in [\ell]$, compute $\mathsf{ctx}_{\vec{\mathbf{z}}_\theta} \leftarrow \mathsf{ctx}_{\vec{\mathbf{y}}_\theta} + \mathbf{c}_\theta \cdot \mathsf{ctx}_{\vec{\mathbf{r}}}$.
    (b) For each honest prover P_i (with $i \in \mathcal{H}$), compute $\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\vec{\mathbf{z}}_\theta}, \vec{\mathbf{z}}_\theta^* + \vec{\mathbf{z}}_{\theta,\mathcal{V}}, \mathcal{S}, \mathcal{V}, i)$ for each $\theta \in [\ell]$, and output $(\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}})_{\theta=1}^{\ell}$ as P_i's output.

- $\mathsf{SimRO}(k, \mathsf{qry})$:Given a query qry if $H_k(\mathsf{qry}) = \perp$, then proceed as follows: If $k = 1$, sample $(\vec{v}_\theta)_{\theta=1}^{\ell} \leftarrow\!\$\ (\mathbb{Z}_q^\nu)^\ell$ and set $\mathsf{output} := (\vec{v}_\theta)_{\theta=1}^{\ell}$. If $k = 2$, sample $(\mathbf{c}_\theta)_{\theta=1}^{\ell} \leftarrow\!\$\ \mathcal{C}^\ell$ and set $\mathsf{output} := (\mathbf{c}_\theta)_{\theta=1}^{\ell}$. Then set $H_k(\mathsf{qry}) \leftarrow \mathsf{output}$. Return $H_k(\mathsf{qry})$.

Let $(\mathsf{D}, \mathcal{A})$ be a semi-honest ppt adversary in the weak $(\mathtt{t}\text{-}1)$-ZK experiment that performs at most Q_H random oracle queries and Q_P proof queries. We now proceed via a sequence of hybrids that interpolate between the weak $(\mathtt{t}\text{-}1)$-ZK experiment 0 and the weak $(\mathtt{t}\text{-}1)$-ZK experiment 1 (see Definition 4.1):

- Hyb 0: This is the weak $(\mathtt{t}\text{-}1)$-ZK experiment 0 where Commit and Prove are instantiated as in Construction 4. Let E_0 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid.

- Hyb 1: Similar to Hyb 0, except the challenger samples a uniformly random string, uses it to initialize $(\mathsf{D}, \mathcal{A})$'s random tape, and stores this string in memory. Let E_1 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. This is purely syntactic, so $\Pr[\mathsf{E}_0] = \Pr[\mathsf{E}_1]$.

- Hyb 2: Similar to Hyb 1, except during Setup, the challenger samples $\mathbf{Q} \leftarrow\!\$\ \mathrm{GL}_n(R_q)$ and $\bar{\mathbf{B}}_0 \leftarrow\!\$\ R_q^{n\times m-n}$. Then, it sets $\mathbf{B}_0 \leftarrow \mathbf{Q} \cdot [\mathbf{I}_n \mid \bar{\mathbf{B}}_0]$, and keeps $\mathbf{Q}$ in memory. Let E_2 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Lemma 2.1, it follows that $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \le O(hq^{-d/h}) \le \mathsf{negl}(\lambda)$.

- Hyb 3: Similar to Hyb 2, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, where $\mathbf{s} := \mathbb{w}$, the challenger executes $\Pi_{\mathsf{tCP}}.\mathsf{WitnessShare}$ and stores $\mathbb{w}$ in memory. Next, at the start of the execution of the Prove algorithm, it proceeds as follows:

  1. Given $\mathcal{S}$, the set of provers, and $\mathcal{V}$, the set of corrupted provers, compute $\mathcal{H} \leftarrow \mathcal{S} \setminus \mathcal{V}$ and store it in memory.
  2. Sample $(\mathbf{c}_\theta)_{\theta=1}^{\ell} \leftarrow \mathcal{C}^\ell$, and $(\vec{v}_\theta)_{\theta=1}^{\ell} \leftarrow\!\$\ (\mathbb{Z}_q^\nu)^\ell$.
  3. For each $k \in \mathcal{V}$, use $(\mathsf{D}, \mathcal{A})$'s randomness to sample $\vec{\mathbf{r}}_k \leftarrow D_{R,\mathfrak{s}_1}^m$, $\vec{\mathbf{y}}_{\theta,k} \leftarrow D_{R,\mathfrak{s}_2}^m$, and $\mathbf{g}_{\theta,k} \in \{\mathbf{a} \in R_q : \mathsf{ct}(\mathbf{a}) = 0\}$ for each $\theta \in [\ell]$. Then, for each $\theta \in [\ell]$, compute $\vec{\mathbf{z}}_{\theta,\mathcal{V}} \leftarrow \sum_{k\in\mathcal{V}} (\vec{\mathbf{y}}_{\theta,k} + \mathbf{c}_\theta \cdot \vec{\mathbf{r}}_k)$.
  4. For each $k \in \mathcal{H}$, sample $\vec{\mathbf{r}}_k \leftarrow D_{R,\mathfrak{s}_1}^m$, $\vec{\mathbf{y}}_{\theta,k} \leftarrow D_{R,\mathfrak{s}_2}^m$, and $\mathbf{g}_{\theta,k} \leftarrow\!\$\ \{\mathbf{a} \in R_q : \mathsf{ct}(\mathbf{a}) = 0\}$ for each $\theta \in [\ell]$, and compute $\vec{\mathbf{z}}_{\theta,\mathcal{H}} \leftarrow \sum_{k\in\mathcal{H}} (\vec{\mathbf{y}}_{\theta,k} + \mathbf{c}_\theta \cdot \vec{\mathbf{r}}_k)$ for each $\theta \in [\ell]$.
  5. Parse $[\vec{\mathbf{b}}_1 \mid \cdots \mid \vec{\mathbf{b}}_{3\ell+1}]^\top := \mathbf{B}_1$, where $\vec{\mathbf{b}}_i \in R_q^m$.
  6. Compute $\vec{\mathbf{t}}_2 \leftarrow [\vec{\mathbf{b}}_2 \mid \cdots \mid \vec{\mathbf{b}}_{\ell+1}]^\top \cdot \sum_{i\in\mathcal{S}} \vec{\mathbf{r}}_i + [\sum_{i\in\mathcal{S}} \mathbf{g}_{1,i} \mid \cdots \mid \sum_{i\in\mathcal{S}} \mathbf{g}_{\ell,i}]^\top$.
  7. For each $\theta \in [\ell]$, compute $\mathbf{t}_{\ell+2\theta}, \mathbf{t}_{\ell+2\theta+1}, \mathbf{v}_\theta \in R_q$ as described in Steps 2c, 2d, and 2e of Construction 3, but with $\vec{\mathbf{r}} = \sum_{k\in\mathcal{S}} \vec{\mathbf{r}}_k$ and $\vec{\mathbf{y}}_\theta = \sum_{k\in\mathcal{S}} \vec{\mathbf{y}}_{\theta,k}$.
  8. Set $\vec{\mathbf{t}}_3 := [\mathbf{t}_{\ell+2} \mid \cdots \mid \mathbf{t}_{3\ell+1}]^\top$ and $\vec{\mathbf{v}} := [\mathbf{v}_1 \mid \cdots \mid \mathbf{v}_\ell]^\top$.
  9. Compute $\vec{\mathbf{h}}, \vec{\mathbf{v}}' \in R_q^\ell$ as described in Step 7 of Construction 3, but with $\mathbf{g}_\theta = \sum_{k\in\mathcal{S}} \mathbf{g}_{\theta,k}$ and $\vec{\mathbf{y}}_\theta = \sum_{k\in\mathcal{S}} \vec{\mathbf{y}}_{\theta,k}$ for $\theta \in [\ell]$.
  10. If $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}) = \perp$, program $H_1(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}) \leftarrow (\vec{v}_\theta)_{\theta=1}^{\ell}$. Else, abort.

11. If $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}') = \bot$, program $H_2(\mathbb{x}, \vec{\mathbf{t}}, \vec{\mathbf{t}}_2, \vec{\mathbf{t}}_3, \vec{\mathbf{v}}, (\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}, \vec{\mathbf{v}}') \leftarrow (\mathbf{c}_\theta)_{\theta=1}^{\ell}$. Else, abort.

Let E_3 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By an argument similar to the one developed in Hyb 3 of the proof of Theorem 5.5, applied to the tuple $(\vec{\mathbf{w}}_\theta)_{\theta=1}^{\ell}$, it follows that $|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq 2Q_P(Q_H + Q_P) \cdot 2^{-\ell dn}$.

- Hyb 4: Similar to Hyb 3, except decryption shares generated by honest provers are simulated. The steps are quite similar to those developed in Hyb 4 of the proof of Theorem 5.5. Specifically, during Setup, the challenger replaces Π_{tHE} execution with $\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}$, i.e., it computes $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[\mathtt{n}]}, \mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}) \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimKGen}(1^\lambda, \mathtt{n}, \mathtt{t})$ and stores $\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}$ in memory. Next, during the execution of the Prove protocol, the challenger proceeds as follows:

  1. For $\theta \in [\ell]$, let $\mathbf{t}_{\ell+2\theta}$, $\mathbf{t}_{\ell+2\theta+1}$, $\mathbf{v}_\theta$, $\vec{\mathbf{z}}_{\theta,\mathcal{V}}$, $\vec{\mathbf{z}}_{\theta,\mathcal{H}}$ be the values computed in Hyb 3, and let $\vec{\mathbf{z}}_\theta \leftarrow \vec{\mathbf{z}}_{\theta,\mathcal{V}} + \vec{\mathbf{z}}_{\theta,\mathcal{H}}$.
  2. For each $\theta \in [\ell]$, let $\tilde{\mathbf{B}}_\theta \in R_q^{3\times m}$ be the matrix defined in Step 7a of Simulator 4, and compute $\vec{\mathbf{f}}_\theta \leftarrow \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{H}} - \mathbf{c}_\theta \cdot [\mathbf{t}_1 \mid \mathbf{t}_{\ell+2\theta} \mid \mathbf{t}_{\ell+2\theta+1}]^\top$, where $\mathbf{t}_1$ is the value computed during the execution of the Commit protocol. Let $\vec{\mathbb{e}}_1$ denote the first standard basis vector in R_q^{27}, and let $\vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3$ denote the second and third standard basis vectors in R_q^3. Using these, compute

$$\begin{aligned}
\mathbf{v}^*_\theta &\leftarrow \langle \vec{\mathbb{e}}_1, \vec{\mathbf{f}}_\theta \otimes (\vec{\mathbf{f}}_\theta + \mathbf{c}_\theta \cdot \mathbf{1}^3) \otimes (\vec{\mathbf{f}}_\theta - \mathbf{c}_\theta \cdot \mathbf{1}^3) \rangle + \langle \vec{\mathbf{e}}_2, \vec{\mathbf{f}}_\theta \rangle + \mathbf{c}_\theta \cdot \langle \vec{\mathbf{e}}_3, \vec{\mathbf{f}}_\theta \rangle, \\
\vec{\boldsymbol{\omega}}_{\theta,\mathcal{V}} &\leftarrow \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \otimes (\vec{\mathbf{f}}_\theta + \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}) \otimes (\vec{\mathbf{f}}_\theta - \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}), \\
\Delta_{\vec{\mathbf{v}},\theta} &\leftarrow \langle \vec{\mathbb{e}}_1, \vec{\mathbf{f}}_\theta \otimes \Big((\vec{\mathbf{f}}_\theta + \mathbf{c}_\theta \cdot \mathbf{1}^3) \otimes \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \otimes (\vec{\mathbf{f}}_\theta - \mathbf{c}_\theta \cdot \mathbf{1}^3 + \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}}) \Big) \rangle \\
&\quad + \langle \vec{\mathbb{e}}_1, \vec{\boldsymbol{\omega}}_{\theta,\mathcal{V}} \rangle + \langle \vec{\mathbf{e}}_2, \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \rangle + \mathbf{c}_\theta \cdot \langle \vec{\mathbf{e}}_3, \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}_{\theta,\mathcal{V}} \rangle.
\end{aligned}$$

     Finally, set $\vec{\mathbf{v}}^* := [\mathbf{v}^*_1 \mid \cdots \mid \mathbf{v}^*_\ell]$ and $\vec{\Delta}_{\vec{\mathbf{v}}} := [\Delta_{\vec{\mathbf{v}},1} \mid \cdots \mid \Delta_{\vec{\mathbf{v}},\ell}]^\top$, and output $\vec{\mathbf{v}} \leftarrow \vec{\mathbf{v}}^* + \vec{\Delta}_{\vec{\mathbf{v}}}$.
  3. For each $\theta \in [\ell]$, let $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}}$, $\mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}}$, $\mathsf{ctx}_{\mathbf{v}_\theta}$, and $\mathsf{ctx}_{\vec{\mathbf{z}}_\theta}$ be the ciphertexts corresponding to $\mathbf{t}_{\ell+2\theta}$, $\mathbf{t}_{\ell+2\theta+1}$, $\mathbf{v}_\theta$, and $\vec{\mathbf{z}}_\theta$, respectively, computed during the execution of the Prove protocol. Upon receiving $\{(\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,k}}, \mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,k}}, \mathsf{ds}_{\mathbf{v}_{\theta,k}}, \mathsf{ds}_{\vec{\mathbf{z}}_{\theta,k}})_{\theta=1}^{\ell}\}_{k\in\mathcal{V}}$ from the provers in $\mathcal{V}$, compute, for each honest prover $\mathsf{P}_i \in \mathcal{H}$, the decryption shares
     - $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta}}, \mathbf{t}_{\ell+2\theta}, \mathcal{S}, \mathcal{V}, i)$,
     - $\mathsf{ds}_{\mathbf{t}_{\ell+2\theta+1,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{t}_{\ell+2\theta+1}}, \mathbf{t}_{\ell+2\theta+1}, \mathcal{S}, \mathcal{V}, i)$,
     - $\mathsf{ds}_{\mathbf{v}_{\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\mathbf{v}_\theta}, \mathbf{v}_\theta, \mathcal{S}, \mathcal{V}, i)$,
     - $\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\vec{\mathbf{z}}_\theta}, \vec{\mathbf{z}}_{\theta,\mathcal{H}} + \vec{\mathbf{z}}_{\theta,\mathcal{V}}, \mathcal{S}, \mathcal{V}, i)$,

     and return these as its output.

  Let E_4 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Remark D.2, the way $\vec{\mathbf{v}}$ is computed in Step 2 of this hybrid is identical to the way it is computed in Construction 4. Consequently, by inspection and by the correctness of Π_{tHE}, the values $\mathbf{t}_{\ell+2\theta}$, $\mathbf{t}_{\ell+2\theta+1}$, $\mathbf{v}_\theta$, and $\vec{\mathbf{z}}_\theta$ obtained by provers in Hyb 4 are indistinguishable, for each $\theta \in [\ell]$, from those obtained in Hyb 3. The only remaining difference between the two hybrids therefore lies in the generation of (1) the public and secret encryption keys of all parties and (2) the decryption shares of the honest parties. Hence, by the simulation security of Π_{tHE}, it follows that $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \leq \mathsf{negl}(\lambda)$.

- Hyb 5: Similar to Hyb 4, except ciphertexts produced by honest provers are replaced with ciphertexts of zero. Specifically, during the execution of the Prove protocol, upon receiving

$\{\big(\mathsf{ctx}_{\vec{\mathbf{r}}_k}, (\mathbf{t}_{\mathbf{g}_{\theta,k}}, \vec{\mathbf{w}}_{\theta,k}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,k}}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,k}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,k}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,k}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_k\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_k\rangle})_{\theta=1}^{\ell}\big)\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, for each prover $\mathsf{P}_i \in \mathcal{H}$, the challenger computes $\mathsf{ctx}_{\vec{\mathbf{r}}_i} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$ and then for each $\theta \in [\ell]$, do the following:

- Compute $\mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,i}} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0}^m)$.
- Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i}\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
- Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,i}\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
- Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,i}\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
- Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_i\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.
- Compute $\mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_i\rangle} \leftarrow \Pi_{\mathsf{tHE}}.\mathsf{Enc}(\mathsf{pk}, \mathbf{0})$.

and return

$$\big(\mathsf{ctx}_{\vec{\mathbf{r}}_i}, (\mathbf{t}_{\mathbf{g}_{\theta,i}}, \vec{\mathbf{w}}_{\theta,i}, \mathsf{ctx}_{\vec{\mathbf{y}}_{\theta,i}}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,i}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{y}}_{\theta,i}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,i}\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_i\rangle}, \mathsf{ctx}_{\langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_i\rangle})_{\theta=1}^{\ell}\big)$$

as P_i's answer.

Let E_5 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Since the correct values of $\mathbf{t}_{\ell+2\theta}$, $\mathbf{t}_{\ell+2\theta+1}$, $\mathbf{v}_\theta$, and $\vec{\mathbf{z}}_\theta$ (for $\theta \in [\ell]$) are used to simulate the decryption shares, Hyb 4 and Hyb 5 differ only if $(\mathsf{D}, \mathcal{A})$ can distinguish encryptions of $\mathbf{0}^j$, for $1 \le j \le m$, from encryptions of other messages. Hence, by the semantic security of Π_{tHE}, $|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \le \mathsf{negl}(\lambda)$.

- Hyb 6: Similar to Hyb 5, except during the execution of Setup, the challenger samples $\mathbf{V} \leftarrow\!\!\$\ \mathrm{GL}_{\bar{n}}(R_q)$ and $\mathbf{A} \leftarrow\!\!\$\ R_q^{\bar{n}\times(m-\bar{n})}$, and then computes $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \leftarrow \mathbf{V} \cdot [\mathbf{I}_{\bar{n}} \mid \mathbf{A}]$ and keeps $\mathbf{V}$ in memory. By Lemma 2.1, we have $|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \le O(hq^{-d/h}) \le \mathsf{negl}(\lambda)$.

- Hyb 7: Similar to Hyb 6, except the vectors $\vec{\mathbf{w}}_{\theta,i}$ for each $\theta \in [\ell]$, together with $\vec{\mathbf{v}}'_i$, produced by all but one of the honest provers are replaced with random vectors over R_q. Specifically, the challenger samples $j \leftarrow\!\!\$\ \mathcal{H}$ and keeps it in memory; this index corresponds to the last honest prover to reply. Then, during the execution of the Prove protocol, the challenger proceeds as follows:
  - For each $\mathsf{P}_i \in \mathcal{H} \setminus \{j\}$, sample $\vec{\mathbf{w}}_{\theta,i} \leftarrow\!\!\$\ R_q^n$ for each $\theta \in [\ell]$ and sample $\vec{\mathbf{v}}'_i \leftarrow\!\!\$\ R_q^\ell$. Then return these as P_i's output.
  - For P_j, compute $\vec{\mathbf{w}}_{\theta,j} \leftarrow \mathbf{B}_0 \cdot \big(\sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_{\theta,i}\big) - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{w}}_{\theta,i}$ for each $\theta \in [\ell]$ and compute $\vec{\mathbf{v}}'_j \leftarrow \begin{bmatrix} \langle \boldsymbol{\psi}_1 \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_2, \sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_{1,i}\rangle \\ \vdots \\ \langle \boldsymbol{\psi}_\ell \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{1+\ell}, \sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_{\ell,i}\rangle \end{bmatrix} - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{v}}'_i$. Then return these as P_j's output.

  Let E_7 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Hyb 7 differs from Hyb 6 in how the vectors $\vec{\mathbf{w}}_{\theta,i}$ and $\vec{\mathbf{v}}'_i$ are handled. For each $\mathsf{P}_i \in \mathcal{H} \setminus \{j\}$, we replace $\vec{\mathbf{w}}_{\theta,i}$ and $\vec{\mathbf{v}}'_i$ with a uniformly random sample, and for the remaining honest prover P_j, we set its contribution so that the aggregate $\sum_{i\in\mathcal{H}} \vec{\mathbf{w}}_{\theta,i}$ and $\sum_{i\in\mathcal{H}} \vec{\mathbf{v}}'_i$ is indistinguishable from that in Hyb 6.

  Recall that, from Hyb 6, we have $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top = \mathbf{V} \cdot [\mathbf{I}_{\bar{n}} \mid \mathbf{A}]$. Consequently, still in Hyb 6, the contribution $\vec{\mathbf{w}}_{\theta,i}$ and $\vec{\mathbf{v}}'_i$ of each $\mathsf{P}_i \in \mathcal{H} \setminus \{j\}$ can be computed from the matrix $[\mathbf{B}_0^\top \mid \mathbf{B}_1^\top]^\top \cdot \mathbf{Y}_i = \mathbf{V} \cdot [\mathbf{I}_{\bar{n}} \mid \mathbf{A}] \cdot \mathbf{Y}_i$, where $\mathbf{Y}_i = [\vec{\mathbf{y}}_{1,i} \mid \cdots \mid \vec{\mathbf{y}}_{\ell,i}]$. Treating $\mathbf{Y}_i$ as the secret, the ℓ-Hint-MLWE$^{1,\mathfrak{s}_2,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_2}$ assumption implies that this matrix is computationally indistinguishable from random, and hence so

are $\vec{\mathbf{w}}_{\theta,i}$ and $\vec{\mathbf{v}}'_i$ themselves, which is precisely their distribution in Hyb 7. Since $(\mathsf{D}, \mathcal{A})$ performs at most Q_P proof queries, and at most $\mathtt{n}-2$ contributions are replaced per query, then applying a proof strategy similar to the one used in Hyb 6 of the proof of Theorem 5.5 yields $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]| \leq \mathtt{n} \cdot Q_P \cdot \epsilon_{\ell\text{-Hint-MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_2}}$, where $\epsilon_{\ell\text{-Hint-MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_2}}$ is the advantage of a ppt distinguisher against the $\ell\text{-Hint-MLWE}^{1,\mathfrak{s}_2,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_2}$ assumption. Since the assumption holds by hypothesis, we conclude that $|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_7]| \leq \mathsf{negl}(\lambda)$.

- Hyb 8: Similar to Hyb 7, except during the execution of Commit, the contributions of all but one honest prover are replaced with random vectors over R_q^{n+1}. Specifically, the challenger proceeds as follows:
  - For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $[\vec{\mathbf{t}}_{0,i}^{\top} \mid \mathbf{t}_{1,i}]^{\top} \leftarrow\!\$\ R_q^{n+1}$ and return it as P_i's output.
  - Compute $\vec{\mathbf{t}}'_0 \leftarrow \mathbf{B}_0 \cdot (\sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i)$ and $\mathbf{t}'_1 \leftarrow \langle \vec{\mathbf{b}}_1, \sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i \rangle + \sum_{i\in\mathcal{H}} \lambda_i [\![\mathbf{s}]\!]_i$.
  - For P_j, compute $\vec{\mathbf{t}}_{0,j} \leftarrow \vec{\mathbf{t}}'_0 - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{t}}_{0,i}$ and $\mathbf{t}_{1,j} \leftarrow \mathbf{t}'_1 - \sum_{i\in\mathcal{H}\setminus\{j\}} \mathbf{t}_{1,i}$. Then, return $[\vec{\mathbf{t}}_{0,j}^{\top} \mid \mathbf{t}_{1,j}]^{\top}$ as its output.

  Next, during the execution of Prove, the contributions of all honest provers except P_j to the computation of $\vec{\mathbf{t}}_2$ are replaced with random components in R_q. Specifically, the challenger proceeds as follows:
  - For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $\mathbf{t}_{\mathbf{g}_{\theta,i}} \leftarrow\!\$\ R_q$ for $\theta \in [\ell]$.
  - For P_j, compute $\mathbf{t}_{\mathbf{g}_{\theta,j}} \leftarrow \langle \vec{\mathbf{b}}_{\theta+1}, \sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i \rangle + \sum_{i\in\mathcal{H}} \mathbf{g}_{\theta,i} - \sum_{i\in\mathcal{H}\setminus\{j\}} \mathbf{t}_{\mathbf{g}_{\theta,i}}$ for $\theta \in [\ell]$.

  Let E_8 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By an argument similar to the one developed in Hyb 6 of the proof of Theorem 5.5, but applied to $\vec{\mathbf{r}}_i$, we have $|\Pr[\mathsf{E}_7] - \Pr[\mathsf{E}_8]| \leq \mathtt{n} \cdot Q_P \cdot \epsilon_{\mathsf{Hint-MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}}$, where $\epsilon_{\mathsf{Hint-MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}}$ is the advantage of a ppt distinguisher against the $\mathsf{Hint-MLWE}^{1,\mathfrak{s}_1,\{1\}}_{m-\bar{n},\bar{n},q,\mathfrak{s}_1}$ assumption. Since the assumption holds by hypothesis, we conclude that $|\Pr[\mathsf{E}_7] - \Pr[\mathsf{E}_8]| \leq \mathsf{negl}(\lambda)$.

- Hyb 9 : Similar to Hyb 8, except that during the execution of Commit and Prove, the challenger replaces $\sum_{i\in\mathcal{H}} \vec{\mathbf{r}}_i$ and $\sum_{i\in\mathcal{H}} \vec{\mathbf{y}}_{\theta,i}$ with $\vec{\mathbf{r}}_{\mathcal{H}} \leftarrow D^m_{R,\mathfrak{s}_1\sqrt{|\mathcal{H}|}}$ and $\vec{\mathbf{y}}_{\theta,\mathcal{H}} \leftarrow D^m_{R,\mathfrak{s}_2\sqrt{|\mathcal{H}|}}$, respectively, for each $\theta \in [\ell]$.

  Let E_9 be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By Lemma 2.4, $|\Pr[\mathsf{E}_8] - \Pr[\mathsf{E}_9]| \leq \mathsf{negl}(\lambda)$.

- Hyb 10: Similar to Hyb 9, except that during the execution of Prove, the non-constant terms of the contributions of all but one of the honest provers in the computation of $\vec{\mathbf{h}}$ are replaced with random vectors over R_q^{ℓ}. Specifically, the challenger proceeds as follows:
  - Let $V = [\vec{v}_1 \mid \cdots \mid \vec{v}_\ell] \in \mathbb{Z}_q^{\nu\times\ell}$.
  - For each prover P_i with $i \in \mathcal{H} \setminus \{j\}$, sample $\overline{\mathsf{ct}}(\vec{\mathbf{h}}_i) \leftarrow\!\$\ \{\vec{\mathbf{a}} \in R_q^{\ell} : \mathsf{ct}(\vec{\mathbf{a}}) = 0^{\ell}\}$, compute $\mathsf{ct}(\vec{\mathbf{h}}_i) \leftarrow V^{\top} A(\lambda_i \mathsf{NTT}([\![\mathbf{s}]\!]_i))$, and set $\vec{\mathbf{h}}_i \leftarrow \mathsf{ct}(\vec{\mathbf{h}}_i) + \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i)$.
  - For P_j, compute $\mathsf{ct}(\vec{\mathbf{h}}_j) \leftarrow V^{\top} A(\lambda_j \mathsf{NTT}([\![\mathbf{s}]\!]_j))$ and

$$\overline{\mathsf{ct}}(\vec{\mathbf{h}}_j) \leftarrow \sum_{i\in\mathcal{H}} \begin{bmatrix} \mathbf{g}_{1,i} \\ \vdots \\ \mathbf{g}_{\ell,i} \end{bmatrix} + \overline{\mathsf{ct}} \left((\sum_{i\in\mathcal{H}} \lambda_i [\![\mathbf{s}]\!]_i) \begin{bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_\ell \end{bmatrix} \right) - \sum_{i\in\mathcal{H}\setminus\{j\}} \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i)$$

    , and set $\vec{\mathbf{h}}_j \leftarrow \mathsf{ct}(\vec{\mathbf{h}}_j) + \overline{\mathsf{ct}}(\vec{\mathbf{h}}_j)$.

Let E_{10} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. Note that in Hyb 9 we have $\mathsf{ct}(\vec{\mathbf{g}}_i) = 0^\ell$ for each honest prover P_i with $i \in \mathcal{H}$, where $\vec{\mathbf{g}}_i = [\mathbf{g}_{1,i} \mid \cdots \mid \mathbf{g}_{\ell,i}]^\top$. Hence $\mathsf{ct}(\vec{\mathbf{h}}_i) = \mathsf{ct}(\lambda_i [\![\mathbf{s}]\!]_i \cdot [\boldsymbol{\psi}_1 \mid \cdots \mid \boldsymbol{\psi}_\ell]^\top)$, where $\boldsymbol{\psi}_\theta = \mathsf{NTT}^{-1}(d\, \vec{v}_\theta^\top A)$ for $\theta \in [\ell]$. Focusing on one of the components, we have $\mathsf{ct}(\lambda_i [\![\mathbf{s}]\!]_i \boldsymbol{\psi}_\theta) = \lambda_i \langle \mathsf{NTT}([\![\mathbf{s}]\!]_i), \vec{v}_\theta^\top A \rangle$ by Lemma 2.2. Therefore, the constant terms in Hyb 9 and Hyb 10 are identical. Next, consider the non-constant terms. In Hyb 9, for each $i \in \mathcal{H} \setminus \{j\}$, we have $\overline{\mathsf{ct}}(\mathbf{h}_{\theta,i}) = \overline{\mathsf{ct}}(\lambda_i [\![\mathbf{s}]\!]_i \boldsymbol{\psi}_\theta) + \mathbf{g}_{\theta,i}$, where all the non-constant terms of $\mathbf{g}_{\theta,i}$ are uniformly random over R_q. In addition, recall that in Hyb 8 the commitment $\mathbf{t}_{\mathbf{g}_{\theta,i}}$ to $\mathbf{g}_{\theta,i}$ was replaced by a uniformly random value over R_q, and is therefore independent of $\mathbf{g}_{\theta,i}$. Hence, by a one-time-pad argument, $\overline{\mathsf{ct}}(\mathbf{h}_{\theta,i})$ is uniform over R_q in Hyb 9, which is precisely its distribution in Hyb 10. Finally, the contribution of P_j is adjusted so that verification holds. We conclude that $\Pr[\mathsf{E}_9] = \Pr[\mathsf{E}_{10}]$.

- Hyb 11: Similar to Hyb 10, except that during the execution of Prove, the last prover P_j replaces $\sum_{i \in \mathcal{H}} \mathbf{g}_{\theta,i}$ with $\tilde{\mathbf{g}}_\theta$, where $\tilde{\mathbf{g}}_\theta \leftarrow\!\$\, \{\mathbf{a} \in R_q : \mathsf{ct}(\mathbf{a}) = 0\}$. Since this is just a replacement of the sum of uniform variables over a finite group by a uniform variable, it follows that $\Pr[\mathsf{E}_{10}] = \Pr[\mathsf{E}_{11}]$.

- Hyb 12: Similar to Hyb 11, except that during the execution of Commit, the challenger sets $\mathbf{t}'_1 \leftarrow \langle \vec{\mathbf{b}}_1, \vec{\mathbf{r}}_{\mathcal{H}} \rangle + \mathbf{s}$ and, for the last honest prover P_j, sets $\mathbf{t}_{1,j} \leftarrow \mathbf{t}'_1 - \sum_{i \in \mathcal{S} \setminus \{j\}} \mathbf{t}_{1,i}$. Next, during the execution of Prove, for each $\theta \in [\ell]$, it sets:
  - $\mathbf{t}_{\ell+2\theta} \leftarrow \langle \vec{\mathbf{b}}_{\ell+2\theta}, \vec{\mathbf{r}}_{\mathcal{H}} \rangle + \langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{y}}_{\theta,\mathcal{H}} \rangle - 3\mathbf{s} \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,\mathcal{H}} \rangle^2$;
  - $\mathbf{t}_{\ell+2\theta+1} \leftarrow \langle \vec{\mathbf{b}}_{\ell+2\theta+1}, \vec{\mathbf{r}}_{\mathcal{H}} \rangle + (3\mathbf{s}^2 - 1) \langle \vec{\mathbf{b}}_1, \vec{\mathbf{y}}_{\theta,\mathcal{H}} \rangle$; and
  - for the last honest prover P_j,
    - $\mathbf{t}_{\mathbf{g}_{\theta,j}} \leftarrow \langle \vec{\mathbf{b}}_{\theta+1}, \vec{\mathbf{r}}_{\mathcal{H}} \rangle + \tilde{\mathbf{g}}_\theta - \sum_{i \in \mathcal{S} \setminus \{j\}} \mathbf{t}_{\mathbf{g}_{\theta,i}} + \langle \boldsymbol{\psi}_\theta \vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{\theta+1}, \vec{\mathbf{r}}_{\mathcal{V}} \rangle$
    - and $\mathbf{h}_{\theta,j} \leftarrow \mathsf{ct}(\lambda_j [\![\mathbf{s}]\!]_j \boldsymbol{\psi}_\theta) + \overline{\mathsf{ct}}(\mathbf{s} \cdot \boldsymbol{\psi}_\theta) + \tilde{\mathbf{g}}_\theta - \sum_{i \in \mathcal{S} \setminus \{j\}} \overline{\mathsf{ct}}(\mathbf{h}_{\theta,i})$, where $\mathbf{h}_{\theta,j}$ denotes the θ-th component of $\vec{\mathbf{h}}_j$.

  Let E_{12} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By an argument similar to the one developed in Hyb 10 of Theorem 5.5, it follows by the $\mathsf{Hint\text{-}MLWE}^{\ell, \mathfrak{s}_2 \sqrt{|\mathcal{H}|}, \mathcal{C}}_{m-\bar{n}, \bar{n}, q, \mathfrak{s}_1 \sqrt{|\mathcal{H}|}}$ assumption that the values $\mathbf{t}_{\ell+2\theta}$ and $\mathbf{t}_{\ell+2\theta+1}$ are computationally indistinguishable from those in Hyb 11, and that the contributions of P_j to $\mathbf{t}_1$ during Commit, and to $\mathbf{t}_{\mathbf{g}_\theta}$ and $\mathbf{h}_\theta$ during Prove, are also indistinguishable from those in Hyb 11. Hence, $|\Pr[\mathsf{E}_{11}] - \Pr[\mathsf{E}_{12}]| \le 2 \cdot Q_P \cdot \epsilon_{\mathsf{Hint\text{-}MLWE}^{\ell, \mathfrak{s}_2 \sqrt{|\mathcal{H}|}, \mathcal{C}}_{m-\bar{n}, \bar{n}, q, \mathfrak{s}_1 \sqrt{|\mathcal{H}|}}}$, where $\epsilon_{\mathsf{Hint\text{-}MLWE}^{\ell, \mathfrak{s}_2 \sqrt{|\mathcal{H}|}, \mathcal{C}}_{m-\bar{n}, \bar{n}, q, \mathfrak{s}_1 \sqrt{|\mathcal{H}|}}}$ denotes the maximum advantage of any ppt distinguisher against the $\mathsf{Hint\text{-}MLWE}^{\ell, \mathfrak{s}_2 \sqrt{|\mathcal{H}|}, \mathcal{C}}_{m-\bar{n}, \bar{n}, q, \mathfrak{s}_1 \sqrt{|\mathcal{H}|}}$ assumption. Since the assumption holds by hypothesis, $|\Pr[\mathsf{E}_{11}] - \Pr[\mathsf{E}_{12}]| \le \mathsf{negl}(\lambda)$.

- Hyb 13: Similar to Hyb 12, except during the execution of Prove, given the statement $(A, \vec{u}) := \mathbb{x}$, the challenger proceeds as follows:
  1. Let $\vec{\mathbf{h}} \leftarrow \mathbf{s}[\boldsymbol{\psi}_1 \mid \cdots \mid \boldsymbol{\psi}_\ell]^\top + [\tilde{\mathbf{g}}_1 \mid \cdots \mid \tilde{\mathbf{g}}_\ell]^\top$.
  2. Upon receiving $(\vec{\mathbf{h}}_k, \vec{\mathbf{v}}'_k)_{k \in \mathcal{V}}$ from provers in $\mathcal{V}$, it does the following:
     (a) Form the set $\mathcal{T}_{\mathsf{ct}} := \{\mathsf{ct}(\vec{\mathbf{h}})\} \cup \{\lambda_k^{-1} \cdot \mathsf{ct}(\vec{\mathbf{h}}_k)\}_{k \in \mathcal{V}} \subset \mathbb{Z}_q^\ell$, where $\{\lambda_k\}_{k \in \mathcal{V}} \subset \mathbb{Z}_q^\times$ are the Lagrange coefficients of the corrupted provers with respect to $\mathcal{S}$; the factor λ_k^{-1} recovers each corrupted prover's underlying constant-term share, so that $\mathcal{T}_{\mathsf{ct}}$ is a Shamir sharing with secret $\mathsf{ct}(\vec{\mathbf{h}})$.
     (b) If $|\mathcal{T}_{\mathsf{ct}}| < \mathtt{t}$, pad the set as follows:

– Pick $\mathtt{t} - |\mathcal{T}_{\mathsf{ct}}|$ distinct honest provers' witness shares $\{[\![\mathbf{s}]\!]_i\}_{i\in\mathcal{H}}$.
– For each, compute $\vec{u}_i \leftarrow A \cdot \mathsf{NTT}([\![\mathbf{s}]\!]_i) \in \mathbb{Z}_q^{\nu}$ and add the vector $(\langle\vec{u}_i, \vec{v}_1\rangle, \ldots, \langle\vec{u}_i, \vec{v}_\ell\rangle)^\top \in \mathbb{Z}_q^{\ell}$ to $\mathcal{T}_{\mathsf{ct}}$.

(c) Interpret $\mathcal{T}_{\mathsf{ct}}$ as a Shamir sharing with secret $\mathsf{ct}(\vec{\mathbf{h}})$ (its value at evaluation point 0), and let ϕ be its Lagrange interpolant. Then, for each P_i with $i \in \mathcal{H}$, compute the constant term $\mathsf{ct}(\vec{\mathbf{h}}_i) \leftarrow \lambda_i \cdot \phi(i)$, which corresponds to the evaluation $\phi(i)$ scaled by P_i's Lagrange coefficient $\lambda_i \in \mathbb{Z}_q^{\times}$ (with respect to $\mathcal{S}$).

(d) For P_j, compute $\overline{\mathsf{ct}}(\vec{\mathbf{h}}_j) \leftarrow \overline{\mathsf{ct}}(\vec{\mathbf{h}}) - \sum_{i\in\mathcal{S}\setminus\{j\}} \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i)$.

Let E_{13} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. The difference between Hyb 13 and Hyb 12 is purely syntactic. Indeed, by the correctness of Shamir's secret sharing and using the fact that NTT is a ring homomorphism, $\mathsf{ct}(\vec{\mathbf{h}}) = \sum_{i\in\mathcal{S}} \mathsf{ct}(\vec{\mathbf{h}}_i) = V^\top A \left(\sum_{i\in\mathcal{S}} \lambda_i \mathsf{NTT}([\![\mathbf{s}]\!]_i)\right) = V^\top A \cdot \mathsf{NTT}(\mathbf{s}) = V^\top \vec{u}$. Therefore, in Step 2c, the Lagrange interpolant computed is exactly the unique interpolant passing through every prover's constant-term share. In addition, from Hyb 12, we have $\overline{\mathsf{ct}}(\vec{\mathbf{h}}) = \sum_{i\in\mathcal{S}} \overline{\mathsf{ct}}(\vec{\mathbf{h}}_i) = [\tilde{\mathbf{g}}_1 \mid \cdots \mid \tilde{\mathbf{g}}_\ell]^\top + \overline{\mathsf{ct}}(\mathbf{s}[\boldsymbol{\psi}_1 \mid \ldots \boldsymbol{\psi}_\ell]^\top)$. Hence, $\Pr[\mathsf{E}_{12}] = \Pr[\mathsf{E}_{13}]$.

- Hyb 14: Similar to Hyb 13, except during the execution of Prove, the challenger proceeds as follows: during the construction of the set $\mathcal{T}_{\mathsf{ct}}$, if $(\mathsf{D}, \mathcal{A})$ corrupted fewer than $\mathtt{t} - 1$ provers, the challenger pads $\mathcal{T}_{\mathsf{ct}}$ by sampling $\mathtt{t} - |\mathcal{T}_{\mathsf{ct}}|$ vectors $\vec{s}_i \leftarrow\!\!\$\ \mathbb{Z}_q^d$ and adding $V^\top A\vec{s}_i$ to $\mathcal{T}_{\mathsf{ct}}$.

  Let E_{14} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. First, note that if $(\mathsf{D}, \mathcal{A})$ corrupted exactly $\mathtt{t} - 1$ provers, then the constant term of the honest provers' contribution to $\vec{\mathbf{h}}$ corresponds to what $(\mathsf{D}, \mathcal{A})$ would have computed on its own: since $\mathsf{ct}(\vec{\mathbf{h}}) = V^\top \vec{u}$, where $\vec{u} \in \mathbb{Z}_q^{\nu}$ is part of the statement and corresponds to the evaluation point 0, the adversary can use this together with its $\mathtt{t} - 1$ shares to recover the unique interpolant passing through the honest provers' constant-term shares. If it corrupts fewer than $\mathtt{t} - 1$ provers, then by the security of Shamir secret sharing the unique interpolant is completely hidden from $(\mathsf{D}, \mathcal{A})$; therefore, padding $\mathcal{T}_{\mathsf{ct}}$ with shares generated as images of random vectors in $\mathbb{Z}_q^d$ under the linear transformation $V^\top A$ does not change $(\mathsf{D}, \mathcal{A})$'s view, and we conclude that $|\Pr[\mathsf{E}_{13}] - \Pr[\mathsf{E}_{14}]| \leq \mathsf{negl}(\lambda)$.

- Hyb 15: Similar to Hyb 14. However, upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger runs $\mathsf{vENS.Prove}$ (see Construction 3) with $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$ and standard deviations $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to obtain $(\vec{\mathbf{t}}^*, \pi := (\vec{\mathbf{t}}_2^*, \vec{\mathbf{t}}_3^*, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}_\theta^*, \vec{v}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta, \vec{\mathbf{z}}_\theta^*)_{\theta=1}^{\ell}))$.

  During the execution of Commit protocol, upon receiving $\{[\vec{\mathbf{t}}_{0,k}^\top \mid \mathbf{t}_{1,k}]^\top\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, the challenger parses $[(\vec{\mathbf{t}}_0^*)^\top \mid \mathbf{t}_1^*]^\top := \vec{\mathbf{t}}^*$ and proceeds as follows:

  – For all provers P_i, with $i \in \mathcal{H} \setminus \{j\}$, sample $[\vec{\mathbf{t}}_{0,i}^\top \mid \mathbf{t}_{1,i}]^\top \leftarrow\!\!\$\ R_q^{n+1}$ and return that as its output.
  – For P_j, compute $\vec{\mathbf{t}}_{0,j} \leftarrow \vec{\mathbf{t}}_0^* - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{t}}_{0,i}$ and $\mathbf{t}_{1,j} \leftarrow \mathbf{t}_1^* - \sum_{i\in\mathcal{S}\setminus\{j\}} \mathbf{t}_{1,i}$. Then, return $[\vec{\mathbf{t}}_{0,j}^\top \mid \mathbf{t}_{1,j}]^\top$ as its output.

  Then, during the execution of the Prove protocol, the challenger proceeds as follows:

  – During the computation of $\vec{\mathbf{w}}_\theta$, with $\theta \in [\ell]$, set the contribution of P_j to be $\vec{\mathbf{w}}_{\theta,j} \leftarrow \vec{\mathbf{w}}_\theta^* - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{w}}_{\theta,i}$.
  – Parse $[\mathbf{t}_{\mathbf{g}_1}^* \mid \cdots \mid \mathbf{t}_{\mathbf{g}_\ell}^*]^\top := \vec{\mathbf{t}}_2^*$.
  – For $\theta \in [\ell]$, during the computation of $\mathbf{t}_{\mathbf{g}_\theta}$, set the contribution of P_j to be $\mathbf{t}_{\mathbf{g}_{\theta,j}} \leftarrow \mathbf{t}_{\mathbf{g}_\theta}^* - \sum_{i\in\mathcal{S}\setminus\{j\}} \mathbf{t}_{\mathbf{g}_{\theta,i}} + \langle\boldsymbol{\psi}_\theta\vec{\mathbf{b}}_1 + \vec{\mathbf{b}}_{\theta+1}, \vec{\mathbf{r}}_{\mathcal{V}}\rangle$.

- Parse $[\mathbf{t}^*_{2+\ell} \mid \cdots \mid \mathbf{t}^*_{3\ell+1}]^\top := \vec{\mathbf{t}}^*_3$
- For $\theta \in [\ell]$, set $\mathbf{t}_{\ell+2\theta} \leftarrow \mathbf{t}^*_{\ell+2\theta}$ and $\mathbf{t}_{\ell+2\theta+1} \leftarrow \mathbf{t}^*_{\ell+2\theta+1}$ and use those values to simulate honest provers' decryption shares.
- For each $\theta \in [\ell]$, let $\tilde{\mathbf{B}}_\theta \in R_q^{3\times m}$ be the matrix defined in Step 7a of Simulator 4, and compute $\vec{\mathbf{f}}_\theta \leftarrow \tilde{\mathbf{B}}_\theta \vec{\mathbf{z}}^*_\theta - \mathbf{c}_\theta \cdot [\mathbf{t}^*_1 \mid \mathbf{t}^*_{\ell+2\theta} \mid \mathbf{t}^*_{\ell+2\theta+1}]^\top$. Using $\vec{\mathbf{f}}_\theta$, compute $\vec{\mathbf{v}}$ as in Hyb 4 and use that to simulate honest provers' decryption shares.
- Set $\vec{\mathbf{h}} \leftarrow \vec{\mathbf{h}}^*$, and during the computation of $\vec{\mathbf{v}}'$, set the contribution of P_j to be $\vec{\mathbf{v}}'_j \leftarrow \vec{\mathbf{v}}^{**} - \sum_{i\in\mathcal{H}\setminus\{j\}} \vec{\mathbf{v}}'_i$.
- Upon receiving $\{(\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,k}})_{\theta=1}^{\ell}\}_{k\in\mathcal{V}}$ from provers in $\mathcal{V}$, compute and output $\mathsf{ds}_{\vec{\mathbf{z}}_{\theta,i}} \leftarrow \mathsf{Sim}_{\Pi_{\mathsf{tHE}}}.\mathsf{SimTDec}($
  $\mathsf{st}_{\mathsf{Sim}_{\Pi_{\mathsf{tHE}}}}, \mathsf{ctx}_{\vec{\mathbf{z}}_\theta}, \vec{\mathbf{z}}_{\theta,\mathcal{V}} + \vec{\mathbf{z}}^*_\theta, \mathcal{S}, \mathcal{V}, i)$ for each $\mathsf{P}_i \in \mathcal{H}$ and $\theta \in [\ell]$.

Let E_{15} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. This is just a syntactical change. Therefore $\Pr[\mathsf{E}_{14}] = \Pr[\mathsf{E}_{15}]$.

- Hyb 16 : Similar to Hyb 15. However, upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger only uses $\mathbb{w}$ to run $\Pi_{\mathsf{tCP}}.\mathsf{WitnessShare}$. Next, it runs $\mathsf{Sim}_{\mathsf{vENS}}.\mathsf{SimProve}$ with $\mathsf{pp} := (\mathbf{B}_0, \mathbf{B}_1)$ and standard deviations $\mathfrak{s}_1\sqrt{|\mathcal{H}|}$ and $\mathfrak{s}_2\sqrt{|\mathcal{H}|}$ to obtain $(\vec{\mathbf{t}}^*, \pi := (\vec{\mathbf{t}}^*_2, \vec{\mathbf{t}}^*_3, \vec{\mathbf{v}}^*, (\vec{\mathbf{w}}^*_\theta, \vec{v}_\theta)_{\theta=1}^{\ell}, \vec{\mathbf{h}}^*, \vec{\mathbf{v}}^{**}, (\mathbf{c}_\theta, \vec{\mathbf{z}}^*_\theta)_{\theta=1}^{\ell}))$.

  Let E_{16} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. By the zero-knowledge property of vENS, it follows that $|\Pr[\mathsf{E}_{15}] - \Pr[\mathsf{E}_{16}]| \le \mathsf{negl}(\lambda)$.

- Hyb 17 : Similar to Hyb 16, except upon receiving $(\mathbb{x}, \mathbb{w})$ from D, the challenger discards $\mathbb{w}$ and executes $\mathsf{SimWitShare}$. Let E_{17} be the event that $(\mathsf{D}, \mathcal{A})$ outputs 1 in this hybrid. From Hyb 16, the executions of Commit and Prove are independent of the witness; therefore, the only difference is in the execution of the WitnessShare algorithm, and by the security of Shamir secret sharing and the semantic security of Π_{tHE}, it follows that $|\Pr[\mathsf{E}_{16}] - \Pr[\mathsf{E}_{17}]| \le \mathsf{negl}(\lambda)$.

  Finally, note that Hyb 17 is identical to the weak $(\mathtt{t}\text{-}1)$-ZK experiment 1 (see Definition 4.1) with Simulator 4.

□

E.6 Proof of Theorem 6.6

Proof of Theorem 6.6. This proof is similar to the proof of Theorem 5.7. Given that a threshold proof is identical to a single-prover proof except for the $\sqrt{\mathtt{t}}$ factor, then under the weak-unique response of vENS, proofs issued by an adversary without interacting with the simulator are fresh, i.e, they do not appear in $\mathcal{Q}_{\mathbb{x}}$. Now, using an argument similar to the one developed in Theorem 5.7, we can build an extractor for the threshold setting that uses the extractor of vENS, which exists by Theorem 6.4 and Simulator 4. □

F Proofs of Section 7

Theorem F.1 ([BL26], adapted)**.** *Let $n, k, q \in \mathbb{N}_{>0}$ such that q is an unramified prime in R. Let $\mathbb{I}$ be the set of ideals in R_q. Let $\mathcal{D}$ be a distribution over R_q^n and let $\vec{\mathbf{x}} \sim \mathcal{D}$. For any $\mathbf{A} \leftarrow\!\!\$\ R_q^{k\times n}$ and $\vec{\mathbf{u}} \leftarrow\!\!\$\ R_q^k$,*

$$\Delta\left((\mathbf{A}, \mathbf{A}\vec{\mathbf{x}}), (\mathbf{A}, \vec{\mathbf{u}})\right) \le \frac{1}{2}\sqrt{\sum_{\mathcal{I}\in\mathbb{I}\setminus\{R_q\}} \mathcal{N}(\mathcal{I})^k \cdot \mathsf{Coll}(\mathcal{D} \bmod \mathcal{I})}$$

Lemma F.1. *Let $n, q, \ell, h \in \mathbb{N}_{>0}$ such that $\ell < h \leq d$ are powers of two and q is an unramified prime with inertia degree $r > 0$ in R. Let $g = d/r$, $\xi \geq 0$, and $\mathbb{I}$ be the set of ideals in R_q. For any $\mathcal{I} \in \mathbb{I}$,*

$$\mathsf{Coll}(U(S^n_{\xi,\mathsf{R}_{h,\ell}}) \bmod \mathcal{I}) \leq (2\xi+1)^{-n\left\lceil \frac{rh\mathfrak{d}_{\mathcal{I}}}{\ell d}\right\rceil}$$

where $\mathfrak{d}_{\mathcal{I}} \leq g$ is determined by $\mathcal{I}$.

Proof. Suppose $\mathcal{I} \in \mathbb{I}$. From Lemma A.1, $\mathcal{I} = (\Phi_{\mathcal{I}}(X))$ where $\Phi_{\mathcal{I}}$ is a polynomial of degree $r \cdot \mathfrak{d}_{\mathcal{I}}$, with $\mathfrak{d}_{\mathcal{I}} \leq g$. In addition,

$$\begin{aligned}
\mathsf{Coll}(U(S^n_{\xi,\mathsf{R}_{h,\ell}}) \bmod \mathcal{I}) &= \Pr_{\vec{\mathbf{c}},\vec{\mathbf{c}}' \sim U(S^n_{\xi,\mathsf{R}_{h,\ell}})}[\vec{\mathbf{c}} \equiv \vec{\mathbf{c}}' \bmod \Phi_{\mathcal{I}}(X)] \\
&\leq \max_{\vec{\mathbf{x}}} \Pr\left[\vec{\mathbf{c}} \bmod \Phi_{\mathcal{I}}(X) \equiv \vec{\mathbf{x}} \mid \vec{\mathbf{c}} \leftarrow\$ S^n_{\xi,\mathsf{R}_{h,\ell}}\right] \\
&= (2\xi+1)^{-n\left\lceil \frac{rh\mathfrak{d}_{\mathcal{I}}}{\ell d}\right\rceil}
\end{aligned}$$

where the maximum in the first inequality is taken over all vectors in R_q^n whose components are polynomials in $\mathbb{Z}_q[X]/(\Phi_{\mathcal{I}}(X))$. The last equality follows from the following facts: (1) by writing $\mathbf{c} = \mathbf{c}_0 + \mathbf{c}_1 \in S_{\xi,\mathsf{R}_{h,\ell}}$, where $\deg(\mathbf{c}_0) < r \cdot \mathfrak{d}_{\mathcal{I}}$ and $\deg(\mathbf{c}_1) \geq r \cdot \mathfrak{d}_{\mathcal{I}}$, then $\mathbf{c} \to \mathbf{c} \bmod \Phi_{\mathcal{I}}(X)$ defines a bijection that maps $\mathbf{c}_0 \to \mathbf{c}_0 + (\mathbf{c}_1 \bmod \Phi_{\mathcal{I}}(X))$ and (2) $\mathbf{c}_0$ contains at most $\lceil \frac{rh\mathfrak{d}_{\mathcal{I}}}{\ell d} \rceil$ non-zero coefficients. □

Theorem F.2. *Let $n, k, q, \ell, h \in \mathbb{N}_{>0}$ such that $\ell < h \leq d$ are powers of two and q is an unramified prime with inertia degree $r > 0$ in R. Let $g = d/r$, and $\xi \geq 0$. For any $\mathbf{A} \leftarrow\$ R_q^{k\times n}$, $\vec{\mathbf{u}} \leftarrow\$ R_q^k$, and $\vec{\mathbf{x}} \leftarrow\$ S^n_{\xi,\mathsf{R}_{h,\ell}}$, we have*

$$\Delta\left((\mathbf{A}, \mathbf{A}\vec{\mathbf{x}}), (\mathbf{A}, \vec{\mathbf{u}})\right) \leq \frac{1}{2}\sqrt{((q^k \cdot (2\xi+1)^{-\frac{nh}{\ell d}})^r + 1)^g - 1}$$

Proof. From Theorem F.1, it is sufficient to upper bound $\sum_{\mathcal{I}\in\mathbb{I}\setminus\{R_q\}} \mathcal{N}(\mathcal{I})^k \cdot \mathsf{Coll}(U(S^n_{\xi,\mathsf{R}_{h,\ell}}) \bmod \mathcal{I})$:

$$\begin{aligned}
\sum_{\mathcal{I}\in\mathbb{I}\setminus\{R_q\}} \mathcal{N}(\mathcal{I})^k \cdot \mathsf{Coll}(U(S^n_{\xi,\mathsf{R}_{h,\ell}}) \bmod \mathcal{I}) &= -1 + \sum_{\mathcal{I}\in\mathbb{I}} \mathcal{N}(\mathcal{I})^k \cdot \mathsf{Coll}(U(S^n_{\xi,\mathsf{R}_{h,\ell}}) \bmod \mathcal{I}) \\
&\leq -1 + \sum_{G\subseteq[g]} q^{kr|G|} \cdot (2\xi+1)^{-n\frac{rh|G|}{\ell d}} \\
&= -1 + \sum_{i=0}^{g} \binom{g}{i} (q^k \cdot (2\xi+1)^{-\frac{nh}{\ell d}})^{ri} \\
&= -1 + ((q^k \cdot (2\xi+1)^{-\frac{nh}{\ell d}})^r + 1)^g
\end{aligned}$$

where the inequality follows from Lemmas A.1 and F.1. □

Corollary F.2.1. *Let $\lambda, n, k, q, \ell, h \in \mathbb{N}_{>0}$ such that $\ell < h \leq d$ are powers of two and q is an unramified prime with inertia degree $r > 0$ in R. Let $g = d/r$, and $\xi \geq 0$. In addition, suppose $(\frac{nhr}{\ell d})\log(2\xi+1) \geq kr\log q + \omega(\log\lambda) + \log g$. Then, for any $\mathbf{A} \leftarrow\$ R_q^{k\times n}$, $\vec{\mathbf{u}} \leftarrow\$ R_q^k$, and $\vec{\mathbf{x}} \leftarrow\$ S^n_{\xi,\mathsf{R}_{h,\ell}}$, $\Delta\left((\mathbf{A}, \mathbf{A}\vec{\mathbf{x}}), (\mathbf{A}, \vec{\mathbf{u}})\right) \leq \mathsf{negl}(\lambda)$.*

Proof of Theorem 7.1. Let $\tau : [d] \to \mathbb{Z}_{2d}^\times$ be a bijection. Given $\vec{\mathbf{s}} \leftarrow\$ S^n_{\xi,\mathsf{R}_{h,\ell}}$, let $\vec{r} \in \mathbb{R}^d$ with $r_j = \sqrt{\gamma^2 + \alpha^2 \sum_{i\in[n]} |\sigma_{\tau(j)}(\mathbf{s}_i)|^2}$. Consider the continuous Gaussian distribution $D_{\mathsf{diag}(\vec{r})}$ over $K^m_{\mathbb{R}}$. Note that $s_1(\mathsf{diag}(\vec{r})) = \|\vec{r}\|_\infty \leq \sqrt{\gamma^2 + \alpha^2(\sum_{i\in[n]} \|\sigma(\mathbf{s}_i)\|^2_\infty)} \leq \sqrt{\gamma^2 + \alpha^2 nB^2} = \beta$. Hence, $D_{\mathsf{diag}(\vec{r})} \in \Psi_{\leq\beta}$.

Define an $\widehat{\mathsf{MLWE}}_{n,m,q,\Psi_{\leq\beta}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ instance with respect to $\vec{\mathbf{s}}$ as $(\mathbf{A},\vec{\mathbf{b}})$ where $\vec{\mathbf{b}} = q^{-1}\mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}} \bmod R$ with $\mathbf{A} \leftarrow\!\$\ R_q^{m\times n}$ and $\vec{\mathbf{e}} \leftarrow (D_{\mathsf{diag}(\vec{r})})^m$. Let $\mathcal{A}$ be a ppt adversary that distinguishes $(\mathbf{A},\vec{\mathbf{b}})$ from $(\mathbf{A},\vec{\mathbf{u}})$ with $\vec{\mathbf{u}} \leftarrow\!\$\ \mathbb{T}^m$. We proceed via a sequence of hybrids:

- Hyb 0: $\mathcal{A}$ receives an instance $(\mathbf{A},\vec{\mathbf{b}})$ of $\widehat{\mathsf{MLWE}}_{n,m,q,\Psi_{\leq\beta}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ as defined above. Let E_0 be the event that $\mathcal{A}$ outputs 1 in this hybrid.
- Hyb 1: Similar to Hyb 0. However, $\vec{\mathbf{b}}$ is computed as follows: Sample $\mathbf{N} \leftarrow D_{q^{-1}R,\alpha}^{m\times n}$ and $\vec{\mathbf{e}}' \leftarrow D_\gamma^m$. Then, set $\vec{\mathbf{b}} \leftarrow q^{-1}\mathbf{A}\vec{\mathbf{s}} - \mathbf{N}\vec{\mathbf{s}} + \vec{\mathbf{e}}' \bmod R$.

  Let E_1 be the event that $\mathcal{A}$ outputs 1 in this hybrid, and $\mathfrak{a} = \max_{j\in[d]} \sqrt{\sum_{i\in[n]} |\sigma_{\tau(j)}(\mathbf{s}_i)|^2}$. We have $\left(1/\alpha^2 + \mathfrak{a}^2/\gamma^2\right) \leq 2/\alpha^2$ $\leq (\eta_\varepsilon((q^{-1}R)^n))^{-2}$, and by Lemma A.2, it follows that $\vec{\mathbf{s}}^\top\mathbf{N}^\top + (\vec{\mathbf{e}}')^\top$ is within $2m\varepsilon$ statistical distance of a vector sampled from $(D_{\mathsf{diag}(\vec{r})})^m$, which is the distribution of $\vec{\mathbf{e}}$ in Hyb 0. Therefore, $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq 2m\varepsilon \leq \mathsf{negl}(\lambda)$.
- Hyb 2: Similar to Hyb 1. However, $\mathbf{A}$ is computed as follows: Sample $\mathbf{B} \leftarrow\!\$\ R_q^{m\times k}, \mathbf{C} \leftarrow\!\$\ R_q^{k\times n}$. Then, compute $\mathbf{A} \leftarrow q(q^{-1}\mathbf{B}\mathbf{C} + \mathbf{N} \bmod R)$.

  Let E_2 be the event that $\mathcal{A}$ outputs 1 in this hybrid. Notice that $(\mathbf{C}^\top, q^{-1}\mathbf{A}^\top, \mathbf{N}\vec{\mathbf{s}})$ is an instance of the $\mathsf{ext\text{-}MLWE}^m{}_{k,n,q,D_{q^{-1}R,\alpha},S^n_{\xi,\mathsf{R}_{h,\ell}}}(U(R_q))$ problem. Indeed, if $\mathbf{A} \leftarrow\!\$\ R_q^{m\times n}$ like in Hyb 1, then $q^{-1}\mathbf{A} \sim U((q^{-1}R)/R)$ since $R_q \cong (q^{-1}R)/R$ as an additive group via the multiplication by q^{-1}, and if $\mathbf{A}^\top = q(q^{-1}\mathbf{C}^\top\mathbf{B}^\top + \mathbf{N}^\top \bmod R)$ like in Hyb 2, then $q^{-1}\mathbf{A}^\top$ is distributed exactly as an output of an $\mathsf{ext\text{-}MLWE}^m{}_{k,n,q,D_{q^{-1}R,\alpha},S^n_{\xi,\mathsf{R}_{h,\ell}}}(U(R_q))$ sampler with $\mathbf{B}^\top$ as the secret. Hence, $|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \mathsf{negl}(\lambda)$.
- Hyb 3: Similar to Hyb 2. However, $\vec{\mathbf{b}}$ is computed as follows: Sample $\vec{\mathbf{s}}' \leftarrow\!\$\ R_q^k$ and set $\vec{\mathbf{b}} \leftarrow q^{-1}\mathbf{B}\vec{\mathbf{s}}' + \vec{\mathbf{e}}' \bmod R$.

  Let E_3 be the event that $\mathcal{A}$ outputs 1 in this hybrid. In Hyb 2, we have $\vec{\mathbf{b}} = q^{-1}\mathbf{A}\vec{\mathbf{s}} - \mathbf{N}\vec{\mathbf{s}} + \vec{\mathbf{e}}' \bmod R = q^{-1}\mathbf{B}\mathbf{C}\vec{\mathbf{s}} + \vec{\mathbf{e}}' \bmod R$. Given that $(\frac{nhr}{\ell d})\log(2\xi+1) \geq kr\log q + \omega(\log\lambda) + \log g$, it follows that $\Delta((\mathbf{C},\mathbf{C}\vec{\mathbf{s}}),(\mathbf{C},\vec{\mathbf{s}}')) \leq \mathsf{negl}(\lambda)$ by Corollary F.2.1. Therefore, $|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq \mathsf{negl}(\lambda)$.
- Hyb 4: Similar to Hyb 3. However, $\vec{\mathbf{b}} \leftarrow\!\$\ \mathbb{T}^m$.

  Let E_4 be the event that $\mathcal{A}$ outputs 1 in this hybrid. Notice that $(\mathbf{B},\vec{\mathbf{b}})$ is an instance of the $\widehat{\mathsf{MLWE}}_{k,m,q,D_\gamma}(U(R_q))$ problem. Indeed, in Hyb 3, $(\mathbf{B},\vec{\mathbf{b}})$ is distributed exactly as the output of an $\widehat{\mathsf{MLWE}}_{k,m,q,D_\gamma}(U(R_q))$ sampler, and in Hyb 4, $(\mathbf{B},\vec{\mathbf{b}}) \sim U(R_q^{m\times k} \times \mathbb{T}^m)$. Therefore, $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \leq \mathsf{negl}(\lambda)$.
- Hyb 5: Similar to Hyb 4. However, $\mathbf{A} \leftarrow\!\$\ R_q^{m\times n}$.

  Let E_5 be the event that $\mathcal{A}$ outputs 1 in this hybrid. Notice that $(\mathbf{C}^\top, q^{-1}\mathbf{A}^\top, \mathbf{0})$ is an instance of the $\widehat{\mathsf{ext\text{-}MLWE}}^m{}_{k,n,q,\psi,\{0^d\}}(U(R_q))$ problem. (We do not need to provide hints since $(\mathbf{C}^\top, q^{-1}\mathbf{A}^\top)$ is sufficient to interpolate between Hyb 4 and Hyb 5). By an argument similar to the one used in Hyb 2, it follows that $|\Pr[\mathsf{E}_4] - \Pr[\mathsf{E}_5]| \leq \mathsf{negl}(\lambda)$.

This completes the proof. □

Remark F.1 (Discarding $\widehat{\mathsf{ext\text{-}MLWE}}$). By Lemma A.5, if $S_{\xi,\mathsf{R}_{h,\ell}} \setminus \{0\} \subseteq R_q^\times$, then the hardness of $\widehat{\mathsf{MLWE}}_{n,m,q,\Psi_{\leq\beta}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ can solely be based on the hardness of $\widehat{\mathsf{MLWE}}$ with secrets sampled from $U(R_q)$.

G BGV with automorphisms

G.1 Homomorphic Multiplication for Modified BGV

([BGV12, LPR13, BV11]) This modification of the BGV encryption scheme works over a ring, $R_q = \mathbb{Z}_q[X]/(X^d+1)$ where d is a power of 2. It also is parameterized by a secret distribution, χ, and plaintext modulus, p, such that $\gcd(p,q)=1$. Specifically, our modification uses an additive error during encryption whereas [BGV12] did not include new additive error in the encryption and instead relied on the left-over hash lemma for IND-CPA. The security of this scheme is a known fact and is used in the literature.

Construction G.1 (BGV based on MLWE)**.** For any $\lambda \in \mathbb{N}$, let $q = q(\lambda) > p = p(\lambda)$ such that $\gcd(q,p)=1$. Let $n = n(\lambda)$, $m > n$, and $d = d(\lambda)$ be lattice parameters. Let $\chi_{\mathbf{s}} = \chi_{\mathbf{s}}(\lambda), \chi_{\mathbf{r}} = \chi_{\mathbf{r}}(\lambda), \chi_{\mathbf{e}} = \chi_{\mathbf{e}}(\lambda), \chi_{\mathbf{e}'} = \chi_{\mathbf{e}'}(\lambda)$, and $\chi_{\mathbf{e}^*} = \chi_{\mathbf{e}^*}(\lambda)$ be distributions over R.

- $\mathsf{KGen}(1^\lambda)$: Given $\lambda \in \mathbb{N}$, sample $\mathbf{A} \leftarrow\$ R_q^{m\times n}$. Next, sample vectors $\vec{\mathbf{s}} \leftarrow \chi_{\mathbf{s}}^n$, $\vec{\mathbf{e}} \leftarrow \chi_{\mathbf{e}}^m$ and compute $\vec{\mathbf{t}} \leftarrow \mathbf{A}\vec{\mathbf{s}} + p\cdot\vec{\mathbf{e}}$. Set $\mathsf{sk} := \vec{\mathbf{s}}$ and $\mathsf{pk} := (\mathbf{A}, \vec{\mathbf{t}})$. Further, encrypt the product of each pair of elements of the secret key multiplied by powers of 2: $\mathsf{pk}_{\mathsf{ctx},i,j,k} = 2^k \vec{\mathbf{s}}_i \vec{\mathbf{s}}_j$ for $i,j \in [n], k \in \log q$ and include these ciphertexts in the public key.
- $\mathsf{Encrypt}(\mathsf{pk} = \vec{\mathbf{t}}, \mathbf{m}) \to (\mathsf{ctx} = (\vec{\mathbf{a}}, \mathbf{b}))$: Sample randomness, $\vec{\mathbf{r}} \leftarrow\$ \chi_{\mathbf{r}}^m, (\mathbf{e}', \vec{\mathbf{e}}^*) \leftarrow\$ \chi_{\mathbf{e}'}^{n+1}$. Let $\vec{\mathbf{a}} = \vec{\mathbf{r}}^\top \mathbf{A} + p\vec{\mathbf{e}}^*$ and $\mathbf{b} = \langle \vec{\mathbf{r}}, \vec{\mathbf{t}} \rangle + \mathbf{m} + p\mathbf{e}'$.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ctx})$: Using $\mathsf{sk} = \vec{\mathbf{s}}$, compute: $\mathbf{m} = \mathbf{b} - \langle \vec{\mathbf{a}}, \vec{\mathbf{s}} \rangle \bmod p$
- $\mathsf{Add}(\mathsf{ctx}_1 = (\vec{\mathbf{a}}_1, \mathbf{b}_1), \mathsf{ctx}_2 = (\vec{\mathbf{a}}_2, \mathbf{b}_2))$: Compute and output $\mathsf{ctx} = (\vec{\mathbf{a}}_1 + \vec{\mathbf{a}}_2, \mathbf{b}_1 + \mathbf{b}_2)$.
- $\mathsf{Multiplication}(\mathsf{ctx}_1 = (\vec{\mathbf{a}}_1, \mathbf{b}_1), \mathsf{ctx}_2 = (\vec{\mathbf{a}}_2, \mathbf{b}_2))$: Compute the tensor of the two ciphertexts $\mathsf{ctx} = \mathsf{ctx}_1 \otimes \mathsf{ctx}_2$ and use the extra ciphertexts in the public key to re-linearlize as described in the proof of Theorem G.3.

Theorem G.1 (Correctness of BGV fresh encryptions)**.** *Let* $B_{\mathsf{Dec}} = dpnB_{\mathsf{KG,sec}}B_{\mathsf{Enc,err}} + dpmB_{\mathsf{Enc,sec}}B_{\mathsf{KG,err}} + pB_{\mathsf{Enc,err}} + p$ *where* $B_{\mathsf{KG,sec}} = |\chi_{\mathbf{s}}|_\infty$, $B_{\mathsf{KG,err}} = |\chi_{\mathbf{e}}|_\infty$, $B_{\mathsf{Enc,sec}} = |\chi_{\mathbf{r}}|_\infty$, *and* $B_{\mathsf{Enc,err}} = |\chi_{\mathbf{e}'}|_\infty$. *If* $B_{\mathsf{Dec}} < \lfloor q/2 \rfloor$, *then the scheme in Construction G.1 is correct.*

Theorem G.2 (Correctness of BGV encryptions after addition)**.** *Let* $B^+_{\mathsf{Dec}} = 2(B_{\mathsf{Dec}})$ *where* B_{Dec} *is defined in Theorem G.1. If* $B^+_{\mathsf{Dec}} < \lfloor q/2 \rfloor$, *then the scheme in Construction G.1 is correct under homomorphic addition.*

Theorem G.3 (Correctness of BGV encryptions after multiplication)**.** *Let* $B^*_{\mathsf{Dec}} = (B_{\mathsf{Dec}})^2 + n^2 \log q \cdot (B_{\mathsf{Dec}})$ *where* B_{Dec} *is the value defined in Theorem G.1. If* $B^*_{\mathsf{Dec}} < \lfloor q/2 \rfloor$, *then the scheme in Construction G.1 is correct under homomorphic multiplication.*

Proofs of Theorems G.1 to G.3 can be found in Appendix G.3.

G.2 Threshold Decryption for Modified BGV

Definition G.1 (Threshold decryption functions for modified BGV)**.**

- $\mathsf{KGen}(1^\lambda, \mathtt{n}, \mathtt{t}) \to (\mathsf{pk}, \{\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathtt{n}\})$*: Run the non-threshold version of* KGen *as shown in Construction G.1.*
- $\mathsf{TDec}(\mathsf{sk}_i, \mathsf{ctx}) \to \mathsf{ds}_i$: *Let* $\mathsf{ctx} = (\vec{\mathbf{a}}, \mathbf{b})$. *Sample randomness,* $\mathbf{e}^\dagger \leftarrow\$ \chi_{\mathbf{e}^\dagger}$. *Let* $\mathsf{ds}_i = \lambda_i \mathsf{sk}_i \vec{\mathbf{a}} + p\mathbf{e}^\dagger$.
- $\mathsf{Comb}(\mathsf{pk}, \mathsf{ctx}, D) \to \tilde{m}$: *Let* $\mathsf{ctx} = (\vec{\mathbf{a}}, \mathbf{b})$. *Let* $\tilde{m} = \mathbf{b} - \sum_{\mathsf{ds}_i \in D} \mathsf{ds}_i$

Theorem G.4 (Correctness of threshold BGV decryption)**.** *Let B_{Dec} be as described in Theorem G.1. Let $B_{\mathsf{TDec}} = B_{\mathsf{Dec}} + \sqrt{t}B_{\mathsf{TEnc,err}}$ where $B_{\mathsf{TEnc,err}} = |\chi_{\mathbf{e}^\dagger}|_\infty$. If $B_{\mathsf{TDec}} < \lfloor q/2 \rfloor$, then the scheme in Construction G.1 with the threshold functions in Definition G.1 is correct.*

Theorem G.5 (Simulation security of BGV encryptions)**.** *The modified BGV scheme in Construction G.1 with the threshold functions in Definition G.1 is simulation secure with respect to Definition C.1.*

G.3 Proofs of BGV encryption

Proof of Theorem G.1 We first prove hardness of MLWE using a discrete Gaussian. Theorem 7.1 proves that MLWE is hard over the continuous error distribution D_α. In this section, we will prove that this implies that the hardness of the MLWE assumption with the discrete error distribution $D_{R_q,\alpha\sqrt{2}}$.

Theorem G.6 (Hardness of $\widehat{\mathsf{MLWE}}_{n,m,q,D_{R_q,\alpha\sqrt{2}}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$)**.** *If $\widehat{\mathsf{MLWE}}_{n,m,q,D_\alpha}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ is hard, then $\widehat{\mathsf{MLWE}}_{n,m,q,D_{R_q,\alpha\sqrt{2}}}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$ is hard. Thus, given a solver for the MLWE problem with error sampled from $D_{R_q,\alpha\sqrt{2}}$ a reduction can solve the MLWE problem with continuous error with probability $2\varepsilon \Pr_{D_{R_q,\alpha}}[\mathcal{A}]$ where $\alpha \geq \eta_\varepsilon(R)$ and $\Pr_{D_{R_q,\alpha}}[\mathcal{A}]$ is the advantage of the adversary when the error is sampled from $D_{R_q,\alpha}$ as in $\widehat{\mathsf{MLWE}}_{n,m,q,D_\alpha}(U(S_{\xi,\mathsf{R}_{h,\ell}}))$.*

Proof of Theorem G.6. We do this via a similar reduction to [GKV10] *i.e.,* we construct a transformation, $T : R_q^{m\times n} \times \mathbb{T}^m \to R_q^{m\times n} \times R_q^m$, such that if the $\mathbb{T}^m$ portion of input to the transformation is related to the $R_q^{m\times n}$ portion via $\mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}}$, then the output is related to $\mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{e}}$ where $\vec{\mathbf{e}}$ is sampled from a discrete distribution and thus the result is in R_q^m. If the $\mathbb{T}^m$ portion of the input is uniform, then the resulting output is uniform as well (Unrelated to $\mathbf{A} \in R_q^n$). In the case where $\vec{\mathbf{y}}$ is structured, our transformation T gets $(\mathbf{A}, \vec{\mathbf{y}}) \in R_q^{m\times n} \times \mathbb{T}^m$ as input, and computes $\vec{\mathbf{w}} \leftarrow D_{R-\vec{\mathbf{y}},s_2}$. From Lemma A.3, we have that $\vec{\mathbf{y}}+\vec{\mathbf{w}}$ is within 8ε statistical distance of $D_{R,s}$ where $\mathfrak{s} = \sqrt{\mathfrak{s}_1^2 + \mathfrak{s}_2^2}$ and $\mathfrak{s}_1$ is the statistical distance from which $\vec{\mathbf{y}}$ is sampled from. We then use Lemma A.4 to show that when we mod by q, the resulting value is within 2ε of sampling from R_q^m. In the case where $\vec{\mathbf{y}}$ is uniform over $\mathbb{T}^m$, our output is uniform over R_q^m via Lemma A.4. □

Proof of Theorem G.1 We see that decryption is: $\mathsf{Dec}(\mathsf{sk} = \vec{\mathbf{s}}, \mathsf{ctx} = (\vec{\mathbf{a}}, \mathbf{b})) : \mathbf{b} - \vec{\mathbf{a}}\vec{\mathbf{s}}$. We also see that if this operation is performed in $R = \mathbb{Z}[X]/(X^d + 1)$, then $\mathbf{b} - \vec{\mathbf{a}}\vec{\mathbf{s}} = \mu \pmod p$. But, to preserve security, we must take the ciphertext $\bmod q$ and thus we must ensure that $|\mathbf{b} - \vec{\mathbf{a}}\vec{\mathbf{s}}|$ does not exceed $q/2 - 1$ otherwise it would "overflow" the modulus, q and thus would not decrypt to $\mu \bmod p$. We then see that this value is $\vec{\mathbf{r}}^T\vec{\mathbf{t}} + \mu + p\mathbf{e}' - (\vec{\mathbf{r}}^T\mathbf{A}\vec{\mathbf{s}} + pe^*\vec{\mathbf{s}}) = \vec{\mathbf{r}}^T\mathbf{A}\vec{\mathbf{s}} + \vec{\mathbf{r}}^T\vec{\mathbf{e}} + \mu + p\mathbf{e}' - \vec{\mathbf{r}}^T\mathbf{A}\vec{\mathbf{s}} - p\mathbf{e}'\vec{\mathbf{s}} = p\vec{\mathbf{r}}^T\vec{\mathbf{e}} + \mu + p\mathbf{e}' - p\vec{\mathbf{e}}^*\vec{\mathbf{s}}$. Because $\vec{\mathbf{e}}^*\vec{\mathbf{s}}$ is the product of two vectors of polynomials and we are bounding the ∞ norm, we include the expansion factor of R in the computation. The most a polynomial can "expand" during multiplication (using the coefficient embedding) is by a factor of d (the degree of the polynomial). Thus, we arrive at the final bound on $p\vec{\mathbf{r}}^T\vec{\mathbf{e}} + \mu + p\mathbf{e}' - p\vec{\mathbf{e}}^*\vec{\mathbf{s}}$ being $dpnB_{\mathsf{KG,sec}}B_{\mathsf{Enc,err}} + dpmB_{\mathsf{Enc,sec}}B_{\mathsf{KG,err}} + pB_{\mathsf{Enc,err}} + p$.

Proof of Theorem G.2. This can be seen by adding the coefficients of the polynomial representations of the ciphertexts (see proof of multiplication correctness).

Proof of Theorem G.3. The decryption circuit for BGV can be thought of as a polynomial, $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}})$ where $(\vec{\mathbf{a}}, \mathbf{b})$ is a ciphertext. This function takes a secret key, $\vec{\mathbf{x}}$, as input. Specifically, this polynomial is a degree n polynomial formed as $f_0 + f_1\vec{\mathbf{x}}_1 + \cdots + f_n\vec{\mathbf{x}}_n^n$ where $f_0 = -\mathbf{b}$ and $f_1, \cdots, f_n$ are the elements in the $\vec{\mathbf{a}}$ vector. When this function is evaluated on the correct secret key and then "rounded" (i.e. taken modulo the plaintext modulus, p) then the resulting value is the message μ.

Thus, we can think of a ciphertext as the coefficients of this polynomial and thus to add ciphertexts we simply add the coefficients. For example, adding $\mathsf{ctx}_1 = (\vec{\mathbf{a}}_1, b_1)$ and $\mathsf{ctx}_2 = (\vec{\mathbf{a}}_2, b_2)$ (which are encryptions of $\mathbf{m}_1$ and $\mathbf{m}_2$) results in the final polynomial: $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}}) = f_{(\vec{\mathbf{a}}_1,b_1)}(\vec{\mathbf{x}}) + f_{(\vec{\mathbf{a}}_2,b_2)}(\vec{\mathbf{x}})$ where the coefficients of $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}})$ are an encryption of $\mathbf{m}_1 * \mathbf{m}_2$.

To multiply ciphertexts, we expand the multiplication of polynomials: $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}}) = f_{(\vec{\mathbf{a}}_1,b_1)}(\vec{\mathbf{x}}) f_{(\vec{\mathbf{a}}_2,b_2)}(\vec{\mathbf{x}}) = h_0 + \sum_{i\in[n]} h_i \vec{\mathbf{x}}_i + \sum_{i\in[n]}\sum_{j\in[n]} h_{i,j}\vec{\mathbf{x}}_i\vec{\mathbf{x}}_j$ for some computable coefficients, h_i, $h_{i,j}$ which becomes quadratic in the elements of the secret key. To *re-linearize*, we need to have encryptions of each product, $\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j$. We'll denote and encryption of $\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j$ as $(\vec{\mathbf{a}}_{ij}, b_{ij})$. This relies on our encryption scheme having circular security. Thus, we can rewrite this product polynomial as: $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}}) = h_0 + \sum_{i\in[n]} h_i\vec{\mathbf{x}}_i + \sum_{i\in[n]}\sum_{j\in[n]} h_{i,j} f_{\vec{\mathbf{a}}_{ij},b_{ij}}(\vec{\mathbf{x}})$ which now has $n+1$ coefficients just like fresh encryptions.

Unfortunately, the coefficients $h_{i,j}$ include $\vec{\mathbf{a}}_1$ and $\vec{\mathbf{a}}_2$ from the ciphertexts being multiplied and thus may have large norm. To avoid this, we instead publish $\log(q)$ encryptions of each $\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j$, resulting in $(\vec{\mathbf{a}}_{i,j,k}, b_{i,j,k})_{(i,j)\in[d],k\in\log(q)}$ where $(\vec{\mathbf{a}}_{i,j,k}, b_{i,j,k})$ is an encryption of $2^k\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j$. When we do homomorphic multiplication, we can then decompose $h_{i,j}$ into $h_{i,j} = \sum_{k\in\log(q)} h_{i,j,k}2^k$ where each $h_{i,j,k} \in \{0,1\}^d$ (polynomials with binary coefficients of length d) and then rewrite the "re-linearized terms" in the homomorphically evaluated polynomial as $\sum_{i\in[n]}\sum_{j\in[n]} h_{i,j,k} f_{\vec{\mathbf{a}}_{i,j,k},b_{i,j,k}}(\vec{\mathbf{x}})$. Doing the same binary decomposition for each h_i (resulting in $\vec{\mathbf{a}}_{i,k}, b_{i,k}$ as encryptions of each $2^k\vec{\mathbf{s}}_i$) we can keep the increase in the error small. This results in a polynomial: $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{x}}) = h_0 + \sum_{i\in[n]}\sum_{k\in[\log q]} h_{i,k} f_{a_{i,k},b_{i,k}}(\vec{\mathbf{x}}) + \sum_{i\in[n]}\sum_{j\in[n]}\sum_{k\in[\log q]} h_{i,j,k} f_{a_{ijk},b_{ijk}}(\vec{\mathbf{x}})$.

To bound the error increase of this multiplication, we will follow a similar technique to [BV11] (Section 4.1 of the full version). To simplify the analysis without adding much overhead, we will instead consider the polynomial $f_{\vec{\mathbf{a}},\mathbf{b}}(\vec{\mathbf{x}}) = \sum_{i\in[n]\cup\{0\}}\sum_{j\in[n]\cup\{0\}}\sum_{k\in[\log q]} h_{i,j,k} f_{a_{i,j,k},b_{i,j,k}}(\vec{\mathbf{x}})$ where each $f_{a_{i,j,k},b_{i,j,k}}(\vec{\mathbf{x}})$ for $j = 0$ is an encryption of $2^k\vec{\mathbf{s}}_i$ and for each $i = 0$, this is an encryption of 2^k. We then want to bound $|\vec{\mathbf{b}} - \vec{\mathbf{a}}\vec{\mathbf{s}}|_\infty$ which is the evaluation of $f_{(\vec{\mathbf{a}},\mathbf{b})}$ at $\vec{\mathbf{s}}$. Expanding the $f_{a_{i,j,k},b_{i,j,k}}(\vec{\mathbf{x}})$ values, we see that $f_{(\vec{\mathbf{a}},\mathbf{b})}(\vec{\mathbf{s}}) = \sum_{i\in[n]\cup\{0\}}\sum_{j\in[n]\cup\{0\}}\sum_{k\in[\log q]} h_{i,j,k}(b_{i,j,k} - \vec{\mathbf{a}}_{i,j,k}\vec{\mathbf{s}})$ which equals $\sum_{i\in[n]\cup\{0\}}\sum_{j\in[n]\cup\{0\}}\sum_{k\in[\log q]} h_{i,j,k}(2^k\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j + p\vec{\mathbf{r}}_{i,j,k}^T\vec{\mathbf{e}}_{i,j,k} + pe'_{i,j,k} - p\vec{\mathbf{e}}^*_{i,j,k}\vec{\mathbf{s}})$. We can split this into two sums:

$$\sum_{(i,j)\in[n]\cup\{0\},k\in[\log q]} h_{i,j,k}(2^k\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j) + \sum_{(i,j)\in[n]\cup\{0\},k\in[\log q]} h_{i,j,k}p\vec{\mathbf{r}}^T\vec{\mathbf{e}} + pe'_{i,j,k} - p\vec{\mathbf{e}}^*_{i,j,k}\vec{\mathbf{s}}) \tag{5}$$

We then can see that the first term of Equation 5 is the decryption of the tensor of the two ciphertexts which is equal to: $f_{(\vec{\mathbf{a}}_1,b_1)}(\vec{\mathbf{x}}) * f_{(\vec{\mathbf{a}}_2,b_2)}(\vec{\mathbf{x}}) = (b_1 - \vec{\mathbf{a}}_1)(b_2 - \vec{\mathbf{a}}_2) = (\mathbf{m}_1 + p\vec{\mathbf{r}}_1^T\vec{\mathbf{e}}_1 + pe'_1 - p\vec{\mathbf{e}}^*_1\vec{\mathbf{s}}) * (\mathbf{m}_2 + p\vec{\mathbf{r}}_2^T\vec{\mathbf{e}}_2 + pe'_2 - p\vec{\mathbf{e}}^*_2\vec{\mathbf{s}})$ the infinity norm of which is bounded by $d * (B_{\mathsf{Dec}})^2$ where B_{Dec} is defined in Theorem G.1.

The extra additive error incurred from re-linearization (the second term of Equation 5) is $n^2 \log q * (B_{\mathsf{Dec}})$. This results in the final bound on the noise of the multiplication of two ciphertexts, $B^*_{\mathsf{Dec}} = d * (B_{\mathsf{Dec}})^2 + n^2 \log q * (B_{\mathsf{Dec}})$. □

Proof of Theorem G.5 We define a simulator for use in Fig. 3.

- $\mathsf{ds}_i \leftarrow \mathsf{SimTDec}(\mathsf{st}_{\mathsf{Sim}}, \hat{\mathsf{ctx}}, \hat{\mathbf{m}}, \mathcal{S}, \mathcal{V}, i)$: Let $\mathsf{ctx} = (\vec{\mathbf{a}}, \mathbf{b})$. Outputs $\{\mathsf{ds}_i\}_{i\in\mathcal{H}}$ such that $\hat{\mathbf{m}} = \mathbf{b} - \sum_{i\in\mathcal{H}} - \sum_{i\in\mathcal{V}}$.

If the scheme is semantically secure, we have that the distribution of $\mathsf{ctx} - (\vec{\mathbf{a}}, \mathbf{b})$ is indistinguishable from uniform. Since $\{\mathsf{ds}_i\}_{i\in\mathcal{V}}$ are generated honestly and independently of $\mathbf{b}$ (each ds_i is computing using the now uniform and independent $\vec{\mathbf{a}}$) we have that $\mathbf{b} - \sum_{i\in\mathcal{V}}$ is distributed uniformly. We then see that setting ds_i to this value in the case of a single honest decryptor is computationally indistinguishable from the real world when $\mathsf{MLWE}_{n,m,q,B_{\mathsf{TEnc,err}}}$ is hard.

G.4 Correctness

To ensure we can compute automorphisms on the encrypted secrets of the proofs, we stabilize the secret elements of the BGV encryption scheme. In [ALS20], automorphisms of R_q are defined as $\sigma_i(x) \mapsto x^i \in \mathsf{Aut}(R_q) \cong \mathbb{Z}_{2d}^*$. In their proof, they use ℓ repeated compositions of the automorphism $\sigma = \sigma_{2h/\ell+1}$ where R_q factors into $\phi(2\ell)$ sub-rings and ℓ is set so that $q^{-\ell} \approx 2^{-100}$. These compositions are: $\sigma^i \forall i \in [\ell]$. We can see that for an element $c \in R_q$, where $c = c_0 + c_1 x + \ldots + c_{d-1}x^{d-1}$, if the ring splits fully (*i.e.*,, $h = d$), then we have that any coefficient of the form $c_{i\ell}x^{i\ell}$ for all $i \in [h/\ell]$ is stable. For example, if $q \approx 2^{32}$, then we set $\ell = 4$ to achieve $q^{-\ell} < 2^{-100}$. If the ring does not split fully, we find that the fixed coefficients take the form $c_{(d/h)\ell i}x^{(d/h)\ell i}$. Thus, we can modify our BGV encryption from Construction G.1 to stabilize secrets under this automorphism:

Definition G.2 (Automorphism-friendly BGV key generation)**.** • $\mathsf{KGen}(\mathsf{pp}) \to (\mathsf{pk}, \mathsf{sk})$*: Sample vectors,* $\vec{\mathbf{s}} \leftarrow\$\ S^n_{\xi,\mathsf{R}_{h,\ell}}$ *(as defined in Section 7),* $\vec{\mathbf{e}} \leftarrow\$\ \chi^m_{\mathbf{e}}$*, and compute* $\vec{\mathbf{t}} = \mathbf{A}\vec{\mathbf{s}} + p\vec{\mathbf{e}}$*. Set* $\mathsf{sk} = \vec{\mathbf{s}}$ *and* $\mathsf{pk} = \vec{\mathbf{t}}$*. Further, encrypt the product of each pair of elements of the secret key multiplied by powers of* 2 $2^k\vec{\mathbf{s}}_i\vec{\mathbf{s}}_j$ *for* $i, j \in [n], k \in \log q$ *and publish the ciphertexts.*

The rest of the functions remain as-is from Construction G.1 in Section 2.6. From Construction G.1, we can then see that a ciphertext is: $\vec{\mathbf{a}}, \mathbf{b}$ where $\vec{\mathbf{a}} = \vec{\mathbf{r}}^\top\mathbf{A} + p\vec{\mathbf{e}}^*$ and $\mathbf{b} = \vec{\mathbf{r}}^\top\vec{\mathbf{t}} + \mathbf{m} + p\mathbf{e}' = \vec{\mathbf{r}}^\top\mathbf{A}\vec{\mathbf{s}} + \mathbf{m} + p\mathbf{e}'$. We can see that for two vectors of polynomials, $\vec{\mathbf{a}}, \vec{\mathbf{b}} \in R_q^n$, the automorphism applied to their inner product behaves nicely: $\sigma(\vec{\mathbf{a}}\vec{\mathbf{b}}) = \sigma\left(\sum_i \vec{\mathbf{a}}_i\vec{\mathbf{b}}_i\right) = \sum_i \sigma(\vec{\mathbf{a}}_i\vec{\mathbf{b}}_i) = \sum_i \sigma(\vec{\mathbf{a}}_i)\sigma(\vec{\mathbf{b}}_i) = \sigma(\vec{\mathbf{a}})\sigma(\vec{\mathbf{b}})$ since σ is a ring homomorphism, and thus, matrix multiplication also commutes with σ. Applying an automorphism σ to our ciphertexts yields:$\sigma(\vec{\mathbf{a}}) = \sigma(\vec{\mathbf{r}}^\top)\sigma(\mathbf{A}) + \sigma(p\vec{\mathbf{e}}^*)$ and $\sigma(\mathbf{b}) = \sigma(\vec{\mathbf{r}}^\top)\sigma(\mathbf{A})\sigma(\vec{\mathbf{s}}) + \sigma(\mathbf{m}) + \sigma(p\mathbf{e}')$ and since our secrets are fixed by the automorphism, $\sigma(\vec{\mathbf{s}}) = \vec{\mathbf{s}}$ and thus:

$$\sigma(\mathbf{b}) = \sigma(\vec{\mathbf{r}}^\top)\sigma(\mathbf{A})\vec{\mathbf{s}} + \sigma(\mathbf{m}) + \sigma(p\mathbf{e}')$$

We can now check decryption, where we see that because $p \in \mathbb{Z}_q^*$, it is fixed by the automorphism:

$$\begin{aligned}\mathbf{m}' &= \sigma(\mathbf{b}) - \sigma(\vec{\mathbf{a}})^\top\vec{\mathbf{s}} \pmod p \\ &= \sigma(\vec{\mathbf{r}}^\top)\sigma(\mathbf{A})\vec{\mathbf{s}} + \sigma(\mathbf{m}) + \sigma(p\mathbf{e}') - \sigma(\vec{\mathbf{r}}^\top)\sigma(\mathbf{A})\vec{\mathbf{s}} - \sigma(p\vec{\mathbf{e}}^*)\vec{\mathbf{s}} \pmod p \\ &= \sigma(\mathbf{m}) - p\sigma(\vec{\mathbf{e}}^*)\vec{\mathbf{s}} + p\sigma(\mathbf{e}') \pmod p = \sigma(\mathbf{m})\end{aligned}$$

Applying this automorphism does not increase the error because we see that this only occurs when two indices collide. We see that $\sigma(\mathbf{e}) = \sum \mathbf{e}_i x^{i(2h/\ell+1)}$ and thus these coefficients will collide when $(2h/\ell+1)i \equiv (2h/\ell+1)j \pmod d$. Since $2h/\ell$ is a power of 2, $(2h/\ell+1)$ has an inverse, so this only occurs when: $i \equiv j \pmod d$, which only occurs for $i = j$ and thus no distinct coefficients collide which means the infinity norm stays the same after the automorphism is applied.

G.5 BGV with Automorphisms proof

We can now use Theorem 7.1 to prove that our BGV-style FHE scheme with secrets fixed by automorphisms is IND-CPA. We do this in Appendix G.3, proving Theorem G.7.

Theorem G.7 (IND-CPA of Construction G.1 with modified keygen from Definition G.2)**.** *If*

$$\widehat{\mathsf{MLWE}}_{n,m,q,D_{R_q,\alpha\sqrt{2}}}\left(U(S_{\xi,\mathsf{R}_{h,\ell}})\right)$$

is hard, then the BGV-style FHE scheme from Construction G.1 with modified keygen from Definition G.2 is IND-CPA.