

Tulva: Practical Asynchronous Threshold FHE

(Full version)

Valerio Cini¹, Antonio Guimarães², Russell W. F. Lai³, Michał Osadnik³, and Ivy K. Y. Woo³

¹ Bocconi University, Milan, Italy

² IMDEA Software Institute, Madrid, Spain

³ Aalto University, Espoo, Finland

Abstract. Fully homomorphic encryption (FHE) allows arbitrary computation over encrypted data, but its secret key presents a single point of failure which limits its applications. Threshold FHE patches this gap by distributing the secret key to K parties, requiring a threshold $t \leq K$ of them to cooperate for decryption. We focus on the asynchronous setting where a ciphertext can be decrypted by collecting decryption shares from any t parties who do not need to coordinate or even be online simultaneously. Asynchronous schemes are sought in applications such as privacy-preserving Ethereum Virtual Machines (EVM), processing of long-lived private data with quorum-based decryption and the thresholdisation of any cryptographic primitive in general.

We present **Tulva**, a practical threshold FHE scheme which equips a base FHE scheme with an asynchronous threshold decryption mechanism in a modular way. Our scheme simultaneously (1) is compatible with virtually any base FHE scheme and leaves its ciphertexts unchanged apart from an appended random tag, (2) supports a wide range of threshold profiles (t, K) and ring degrees ≥ 256 with 6–25 bits partial decryption per plaintext bit, and (3) achieves security in the presence of partial decryption queries on challenge ciphertexts based on the standard LWE assumption.

We provide an open-source implementation, benchmark its threshold decryption performance and evaluate the end-to-end costs for typical privacy-preserving EVM use cases. For 16-out-of-512 threshold and a publicly preprocessed ciphertext, computing secret-dependent partial decryption and public share combination requires only 3.7ms and 0.5ms, respectively, on commodity hardware. Compared with the state-of-the-art synchronous scheme [Zyskind et al., CCS’25], we replace their ciphertext-independent yet interactive preparation of secret-correlated randomness with a public, ciphertext-dependent 0.5s preprocessing step, eliminate interaction in the online phase while maintaining similar decryption time on the same machine, and support $32\times$ more total users.

This is the full version of a paper to appear in the proceedings of the 48th IEEE Symposium on Security and Privacy (S&P 2027) with the same title. This version includes additional proofs that are omitted in the proceedings version.

Table of Contents

1	Introduction	3
1.1	Threshold FHE	3
1.2	Our Contributions	4
1.3	Related Work	4
2	Technical Overview	5
2.1	Requirements on Underlying FHE	5
2.2	From FHE to Threshold FHE	6
2.3	Analysis	7
3	Preliminaries	9
4	Threshold FHE Construction	11
4.1	Requirements on Underlying FHE	11
4.2	Threshold FHE Definitions	13
4.3	Threshold FHE from FHE	14
5	Practical Instantiation	16
5.1	Implementation Details	17
5.2	Parameter Selection	18
5.3	Experiments	19
6	Examples of FHEVM Programs	20
A	Algorithms for FHEVM Programs	26
B	Additional Preliminaries	28
C	Proof of Lemma 4 (Quality of Subtractive Sets)	31
D	Proof of Theorem 1 (FHE Leakage Security)	35
E	Proof of Theorem 2 (Tulva Correctness)	37
F	Proofs for Theorem 3 (Tulva Security)	39
F.1	Theorem Statement and Proof	39
F.2	Proofs of Lemmas 12 to 15	41
G	A Utility Lemma and Intermediate Assumption	46
G.1	Intermediate lattice assumption	46
G.2	Utility Lemma	51
G.3	Theorem 4 and proofs	54
G.4	Theorem 5 and proofs	58

1 Introduction

Fully homomorphic encryption (FHE) has rapidly emerged in the past decade as a practically viable tool for computing over private data. However, in many real-world applications, storing the secret decryption key on a single device is undesirable. Threshold FHE distributes the single point of trust by splitting the secret key among K parties, such that any subset of at least t parties can jointly decrypt a ciphertext, while any coalition of fewer than t parties learns nothing about the underlying message.¹ The practical significance of threshold FHE is recognised: it constitutes a standalone category in NIST’s call for multiparty threshold schemes [BP26].

1.1 Threshold FHE

Threshold FHE can be roughly classified into two classes: asynchronous and synchronous/interactive. For an asynchronous scheme (e.g. [BGG⁺18, BS23, BFM⁺25]), any of the K shareholders can locally compute a partial decryption without coordinating with other parties. Anyone can recover the encrypted message by collecting t partial decryptions in any order and possibly across an extended time period. In contrast, the decrypting parties of a synchronous scheme (e.g. [BCD⁺25b, ZZLP25]) must be aware of the set of t participating parties in advance to produce valid partial decryptions. Some synchronous schemes are further interactive (e.g. [ZZLP25]), i.e. the partial decryptors must engage in at least two rounds of communication to decrypt the ciphertext, meaning that all decrypting parties must be simultaneously online during the decryption process.²

The above differences dictate where a threshold FHE can be deployed, and the non-interactivity of asynchronous schemes makes them particularly suited for decentralised, dynamic environments. A prominent application is FHEVM [Zam23a], a confidential smart-contract platform where encrypted account balances and contract states reside on a public ledger. A validator set homomorphically executes smart-contract logic over these ciphertexts, while a separate, rotating committee of decrypting parties is invoked to decrypt only specific outputs (e.g. a transaction-success bit or an auction’s clearing price). The rise of production FHEVM stacks from companies such as Zama [Zam23a] and Phenix [Fhe24] highlights the growing demand for this architecture. Because contracts trigger decryption requests unpredictably and the decryption committee rotates across epochs, the underlying decryption pattern is inherently asynchronous. Synchronous schemes, which require the decrypting parties to be agreed in advance and/or simultaneously online, are structurally incompatible. Beyond FHEVMs, asynchronous threshold FHE further enables generically thresholdising cryptographic primitives [BGG⁺18] without imposing constraints on set-agreement or communication that are inherent to synchronous alternatives.

Despite strong market demand, existing threshold FHE schemes struggle to meet all practical constraints simultaneously. Constructing schemes that simultaneously maximise the range of threshold profiles while minimising partial decryption size, time and communication rounds is challenging, even without asynchronous decryption. This forces a trade-off between asynchrony and efficiency. Among the best benchmarked schemes, Noah’s Ark [DDEK⁺23] is asynchronous only when $t = O(1)$: It requires interactive offline K -party preprocessing for $t \geq \omega(1)$, such that each decryption process consumes one preprocessing run; Zyskind et al. [ZZLP25] is restricted to the K -out-of- K setting and is highly interactive.

Beyond efficiency trade-offs, there is also a subtlety in how threshold FHE security is typically modeled. While the exact security definition of threshold FHE differs greatly from work to work (cf. Section 1.3), none of the definitions that we are aware of models attack scenarios where the adversary can learn partial decryptions of “challenge ciphertexts”, whose underlying plaintexts are (supposedly) unknown to the adversary.³ This is a strict [BKW26] limitation that is especially significant when the threshold FHE is used to thresholdise another cryptographic primitive. Taking threshold signatures as an example, the ability to learn partial decryptions of challenge ciphertexts corresponds to that of learning transcripts of

¹ For some schemes, the recovery threshold could be much larger than the corruption threshold.

² Some schemes have a synchronous/interactive offline phase and an asynchronous online phase.

³ This restriction is sometimes explicit (e.g. [BS23]), but is more often implicitly implied by restricting the adversary to corrupt exactly $t - 1$ users (e.g. [BGG⁺18, OT25, BFM⁺25]), so that any additional partial decryption leads to trivial decryption. In the context of proof, assuming the corruption of exactly $t - 1$ users seems to make a proof much easier, since any partial decryption would imply recovery of the plaintext, which can then be trivially simulated.

unfinished threshold signing protocols, the latter of which is captured in the strongest TS-UF-4 security definition of threshold signatures defined in [BCK⁺22].

1.2 Our Contributions

We present **Tulva**, a practical compiler that turns virtually any⁴ FHE into an asynchronous threshold FHE. The instantiation with TFHE, **tulva-tfhe**, efficiently supports a wide range of threshold profiles⁵, e.g. any $t \leq K \leq 128$ or any t with $\min\{t, K - t\} \leq 16$ for up to $K = 512$, with $\lambda = 128$ bits of estimated security. Our headline result is a 16-out-of-512 scheme whose threshold decryption process can be decomposed into a 0.5s public ciphertext preprocessing step, a 3.7ms partial decryption step, and a 0.5ms public share combination step, where the median timing measurements are obtained from an AWS `c7a.32xlarge` instance with AMD EPYC CPU.

Modular construction. **Tulva** consists of a decryption process which first performs a (public) key-switch operation on the FHE ciphertext, turning it into a threshold public-key encryption (PKE) ciphertext of a compatible threshold PKE scheme, and then performs the threshold decryption procedure of the threshold PKE. Such a modular construction is essentially agnostic to the underlying FHE scheme, hence enabling flexible instantiations depending on the use cases.

We highlight two advantages of key-switching to a dedicated threshold PKE scheme: 1. It enables cleaner decryption noise management and parameter selection: The FHE layer only needs to expose a small set of parameters (e.g. plaintext and ciphertext moduli, decryption noise budget) to the threshold layer, making the parameter selection process of the two layers mostly independent; 2. It allows us to inject additional (pseudo)randomness to ciphertexts to be decrypted. This is crucial for proving that **Tulva** remains secure even when the adversary learns partial decryptions of challenge ciphertexts, under the standard (module) learning with errors (LWE) assumption.

Security given partial decryptions of challenge ciphertexts. Proving this strong security property requires constructing a reduction which simulates partial decryptions that are jointly correlated with all key shares instead of only those of corrupt parties. While proof techniques for this scenario have recently been developed in the simpler context of threshold PKE [CLW25], the threshold FHE setting introduces more complications since many correlated FHE ciphertexts can be derived from a single one. The injection of additional randomness to ciphertexts allows us to extend the technique of [CLW25] to the threshold FHE setting.

Implementation, benchmark and use cases. Our TFHE-based instantiation **tulva-tfhe** consumes ciphertexts produced by the **tfhe-rs** [Zam23b] library directly: a thin compatibility layer repacks and key-switches them into the form expected by our threshold layer, whose performance-critical routines are written in C with AVX-512 optimisations and orchestrated from Python. We benchmark across ring degrees $\varphi \in \{256, 2048\}$ and threshold profiles up to $K = 512$ parties; the small ring minimises single-ciphertext latency, while the large ring packs $\varphi = 2048$ messages per batch for throughput. We further provide end-to-end benchmarks for four confidential smart contract use cases in an FHEVM (Section 6): a sealed-bid auction, a confidential ERC-20 transfer, a dark-pool DEX, and a median price oracle.

1.3 Related Work

There is a substantial literature on threshold FHE, including [AJL⁺12, BGG⁺18, BS23, PS24, OT25, BFM⁺25]. In the following, we focus on comparing with [DDEK⁺23, KLL⁺26, ZZLP25] which are the most practically relevant, of which [ZZLP25] is the only work that makes its implementation publicly available.

Pilvi [CLW25] is a threshold PKE scheme that uses subtractive-set elements as the evaluation points of Shamir sharing, and it is secure even when the adversary obtains partial decryptions of challenge ciphertexts. **Tulva** takes from Pilvi the subtractive-set evaluation points, the security model, and its proof techniques. Applying Pilvi directly to threshold FHE would mean secret-sharing the FHE secret

⁴ We require the FHE to satisfy a “linear decryption” property (Definition 4) which holds for virtually all practical FHE families except CKKS.

⁵ There is no hard limit and the costs scale gracefully.

key itself in the same way, as [KLL⁺26] does, which requires non-black-box changes to the FHE and its parameters. Instead, **Tulva** keeps the FHE and threshold layers separate: a public key switch moves an FHE ciphertext to a dedicated decryption key, and only that key is secret-shared, so the FHE is used as a black box and its ciphertexts and evaluation are unchanged apart from an appended random tag. On the proof side, these random tags are what let us adapt Pilvi’s proof techniques to homomorphically evaluated ciphertexts (see Section 2.3).

In terms of construction, both **Tulva** and Noah’s Ark [DDEK⁺23] secret-share a dedicated decryption key⁶, while the schemes of [ZZLP25, KLL⁺26] perform threshold decryption “in place”. All three schemes, **Tulva** and [DDEK⁺23, KLL⁺26], use noise flooding to mask leakages of the FHE decryption noise and the secret key share. The scheme of [ZZLP25] avoids noise flooding using a rounding protocol based on multiparty computation (MPC), at the cost of having online interactions.

Tulva and [BFM⁺25, KLL⁺26] consider the asynchronous model. Noah’s Ark [DDEK⁺23] is only asynchronous for $t = O(1)$. For $t \geq \omega(1)$, each threshold decryption consumes a ciphertext-independent K -party interactive offline preprocessing, although the online phase is asynchronous. The scheme of [ZZLP25] requires interactions among all K parties in both offline and online phases.

There are mainly two formal models of threshold FHE which define the notion either (1) as a standalone primitive (e.g. this work and [KLL⁺26]) or (2) as MPC protocols for key generation and threshold decryption associated to a base FHE scheme (e.g. [DDEK⁺23, ZZLP25]). Typically, security in the former model is described via security experiments for individual properties, and in the latter via ideal functionalities. All compared works use distinct security definitions, and we are not aware of any result that connects game-/simulation-/UC-based security definitions for threshold FHE, making them difficult to compare. This work inherits the strong security model for threshold PKE considered in [CLW25, BKW26] where the adversary can non-maximally corrupt $|\mathcal{C}| < t$ parties and make less than $t - |\mathcal{C}|$ partial decryption queries on each challenge ciphertext. The model in [KLL⁺26] is weaker [BKW26] in the sense that it forces $|\mathcal{C}| = t - 1$. Understanding the correspondence of this property in UC-based models [DDEK⁺23, ZZLP25] is beyond the scope of this work.

2 Technical Overview

We overview our construction and the security proof. Throughout, we work over a cyclotomic field $\mathcal{K} = \mathbb{Q}(\zeta)$ and its ring of integers $\mathcal{R} = \mathbb{Z}[\zeta]$ with norms measured in the coefficient embedding. The fractional part of $x \in \mathcal{K}$ wrt. the coefficient embedding is written as $\{x\} := x - \lfloor x \rfloor$.

2.1 Requirements on Underlying FHE

Our transformation assumes an FHE scheme **FHE** to satisfy a structural property and a security property.

Structural property. We assume that **FHE** has “linear decryption”⁷ (Definition 4), i.e. the following (typical) properties:

- The decryption key is a vector $\mathbf{sk}_{\text{FHE}} = \hat{\mathbf{s}} \in \mathcal{R}_q^{\hat{n}}$.
- There exists a deterministic transformation **Preproc** that turns a ciphertext into a tuple

$$(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}}) \in \mathcal{R}_q^{\hat{n}} \times \mathcal{R}_q.$$

In particular, we do *not* require the FHE scheme to operate over the same ring $\mathcal{R}_q$ (e.g. it can work over entirely different rings or moduli), as long as the above **Preproc** procedure exists.

- If a ciphertext ctxt_{FHE} is intended to encrypt a message $\mu \in \mathcal{R}_p$ (potentially after some homomorphic evaluations), then it holds that

$$\hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 = \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \hat{e} \bmod q \quad \text{with} \quad \|\hat{e}\|_{\infty} \leq B, \quad (1)$$

for some noise budget B depending on the FHE scheme. We call $\hat{e}$ the FHE decryption noise. This implies $\left\lfloor \frac{p}{q} \cdot (\hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0) \right\rfloor = \mu \bmod p$ when $q/2p > B + \left\{ \frac{q}{p} \right\}$.

⁶ A subtle difference is that in **Tulva** a ciphertext is key-switched to one under the decryption key while in Noah’s Ark [DDEK⁺23] it is bootstrapped.

⁷ The formal definition involves a “slack” factor ξ . We assume $\xi = 1$ in this overview for simplicity.

Virtually all lattice-based constructions of FHE have linear decryption. For most of them, **Preproc** can be implemented with well-known and relatively inexpensive procedures, such as Ring Switching [GHPS12] and Packing Key Switching [CGGI16, MS18, CDKS21]. In fact, some schemes (e.g. BFV [Bra12, FV12]) already often work with ciphertexts satisfying our requirements. An exception is CKKS [CKKS17], which, being an approximate scheme, would not immediately satisfy our correct decryption condition, requiring more complex procedures. We discuss the potential instantiations of **Preproc** and their compatibility with different FHE schemes in Section 5. For our proof-of-concept implementation, we consider the TFHE scheme [CGGI16], as provided by the **tfhe-rs** library [Zam23b], and implement **Preproc** using bootstrappings (to increase moduli) and packing key switching.

Security property. We assume that FHE has IND-CPA-with-leakage security (Definition 5). Informally, this means that FHE remains IND-CPA secure even when the adversary has access to a leakage oracle that, on input an honestly generated ciphertext of its choice, returns the corresponding FHE decryption noise $\hat{e}$ masked by additional noise. We show (Theorem 1) that any IND-CPA-secure FHE scheme with linear decryption satisfies this notion, provided the masking noise is super-polynomially large. We leave the case of polynomially large masking noise to future work.

Apart from these structural and security requirements, our transformation only requires the modulus-to-noise ratio q/B of the underlying FHE scheme to be sufficiently large (super-polynomial in our current analysis). The transformation is otherwise oblivious to the homomorphic evaluation procedures of the underlying FHE scheme. Hence, it preserves any optimisations implemented in the underlying FHE scheme.

2.2 From FHE to Threshold FHE

Next, we describe our generic compiler, **Tulva**, that turns any FHE scheme satisfying the requirements in Section 2.1 into a threshold FHE scheme. Conceptually, our transformation is to pair the FHE with an external linearly-homomorphic threshold PKE (tPKE) scheme which takes care of threshold decryption. The basic idea is to encrypt the FHE secret key under the tPKE, and homomorphically evaluate (the linear part) of the FHE’s decryption function on an FHE ciphertext, i.e. perform a key-switching operation to turn an FHE ciphertext into a tPKE ciphertext. The partial decryption procedure of the tPKE can then be applied. Looking ahead, we will instantiate the tPKE with the classic threshold Regev-PKE scheme. The construction involves two moduli $p \leq q$ for plaintexts and threshold layer ciphertexts, respectively. It is agnostic to the FHE layer modulus, which is left implicit.

Our modular framework offers two major advantages: (1) The parameters for the FHE and tPKE layers are almost decoupled: only the noise budget, the plaintext and ciphertext moduli and some dimensions of the former are exposed to the latter, allowing a simpler parameter selection process. (2) Our security proof can be decomposed into distinct parts concerning solely the FHE or the tPKE components, which allows us to borrow techniques for tPKE to argue strong threshold FHE security (see Section 2.3).

Key Generation. A trusted dealer first generates the FHE key pair $(\text{pk}_{\text{FHE}}, \text{sk}_{\text{FHE}})$, where $\text{sk}_{\text{FHE}} = \hat{\mathbf{s}}$. It then generates a key-switching key $(\mathbf{A}, \mathbf{b})$ which allows anyone to turn an FHE ciphertext into a tPKE ciphertext, as follows: Sample the matrices $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ and $\mathbf{R} \leftarrow \mathcal{R}_q^{t \times n}$ uniformly at random. Compute $\mathbf{r}^T = \mathbf{v}_0^T \cdot \mathbf{R} \bmod q$, where $\mathbf{v}_0 = (1, 0, \dots, 0)^T$ denotes the t -dimensional first unit vector, i.e. $\mathbf{r}^T$ is the first row of $\mathbf{R}$. Sample a short error vector $\mathbf{e} \leftarrow \mathcal{R}^m$. Compute

$$\mathbf{b} := \mathbf{r}^T \cdot \mathbf{A} + \mathbf{e}^T + (\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \mathbf{G} \bmod q, \quad (2)$$

where $\mathbf{G}$ is an appropriate gadget matrix [MP12]. The public key is $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$. The secret key sk_k for the k -th party is the k -th Shamir’s secret-share of the vector $\mathbf{r}$, which can be expressed as the k -th row of $\mathbf{V} \cdot \mathbf{R} \bmod q$, where $\mathbf{V} \in \mathcal{R}^{K \times t}$ is the Vandermonde matrix with the evaluation points as entries.⁸ The norm growth issue can be handled by carefully chosen evaluation points, similar to other recent works [CLW25, KLL⁺26] (see Section 2.3 “optimisations”).

Remark 1 (Distributed key generation). Removing the trusted-dealer assumption requires generating the underlying FHE key material, our key-switching key, and the secret shares in a distributed manner.

⁸ Our security proof will not directly rely on the security of secret-sharing, but via analysing the structure of the Vandermonde matrix.

Generic secure MPC can in principle implement KGen, but an efficient distributed key generation protocol must address both distributed key generation for the underlying FHE scheme and joint sampling of the Gaussian error in the key-switching key. We leave its design and evaluation to future work.

Encryption and Homomorphic Evaluation. The encryption and homomorphic evaluation algorithms are almost exactly those of FHE, except that each ciphertext $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$ is accompanied by an independently uniformly random tag $\mathbf{t} \leftarrow \mathcal{R}_q^h$, which is generated during encryption and refreshed every time a homomorphic operation is performed. This seemingly useless tag injects additional randomness to a ciphertext and is crucial for proving the strong security property in Section 2.3. In practice, $\mathbf{t}$ could e.g. be derived by hashing ctxt_{FHE} with a hash function (fixed in the public parameters) that can reasonably be modelled as a random oracle (cf. Remarks 3 and 6). We note again that fresh FHE ciphertexts ctxt_{FHE} and those undergoing homomorphic evaluations may be over rings different from $\mathcal{R}_q$, as long as they can be transformed to take the form $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \in \mathcal{R}_q^{\hat{n}} \times \mathcal{R}_q$ by the procedure Preproc.

Partial Decryption. For a ciphertext $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$, any party k who wants to contribute its partial decryption first runs the deterministic procedure Preproc to obtain $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$, then performs (part of) a key-switching operation to obtain⁹

$$\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \bmod q, \quad (3)$$

where $\mathbf{G}^{-1}(\cdot)$ is the “gadget-decomposition” operation [MP12]. Each party k then releases the partial decryption

$$\text{pd}_k = \mathbf{s}_k^{\text{T}} \cdot \mathbf{c}_0 + e_k \bmod q$$

where $e_k \leftarrow \mathcal{R}$ is a fresh masking noise.

Remark 2. In applications where ciphertexts are processed by a dedicated entity, e.g. a server, before they are partial-decrypted, Preproc can be run only once by this entity, instead of multiple times by individual partial decryptors.

Combining. After collecting t partial decryptions $(\text{pd}_k)_{k \in T}$ from a set $T \subseteq [K]$ of parties, the ciphertext can be fully decrypted by a public share combination procedure. First, the combiner completes the key-switching operation, i.e. computes $\mathbf{c}_1 := \mathbf{b}^{\text{T}} \cdot \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \bmod q$. Next, it combines the set of partial decryptions $(\text{pd}_k)_{k \in T}$ into

$$\text{pd} := \mathbf{v}_0^{\text{T}} \cdot \mathbf{V}_T^{-1} \cdot (\text{pd}_k)_{k \in T} \bmod q,$$

where $\mathbf{V}_T$ is the submatrix of $\mathbf{V}$ corresponding to the set T of parties. Equivalently, $\mathbf{v}_0^{\text{T}} \cdot \mathbf{V}_T^{-1}$ are the recovery coefficients of the secret-sharing. Then, it removes the encryption in the threshold layer by computing

$$\hat{\mu} := \lfloor \hat{\mathbf{c}}_1 - (\mathbf{c}_1 - \text{pd}) \rfloor \bmod q \quad \text{then} \quad \mu := \left\lfloor \frac{p}{q} \cdot \hat{\mu} \right\rfloor \bmod p.$$

2.3 Analysis

Main parameter constraint. In our transformation, each partial decryptor k adds a moderate amount of noise e_k to the partial decryption pd_k . On the one hand, for security, this noise e_k must be large enough to mask out the FHE decryption noise $\hat{e}$. This ultimately allows us to argue the security of the resulting threshold FHE scheme based on the IND-CPA-with-Leakage security of the underlying FHE scheme. On the other hand, for correctness, the modulus-to-noise ratio q/B of the underlying FHE scheme must be large enough to accommodate the extra masking noise e_k in addition to the existing FHE decryption noise $\hat{e}$.

⁹ In practice, the computation of $\mathbf{c}_0$ can alternatively be done by a designated party, e.g. whoever is performing the last homomorphic evaluation step on the ciphertext.

Security model. Following recent works which studied and refined tPKE security [CLW25, BKW26], we define an analogously refined IND-CPA security model for threshold FHE (Definition 7). For a threshold t , our model supports corrupting a subset of arbitrary $\leq t - 1$ many users, and allows an adversary $\mathcal{A}$ to access limited partial decryptions even for ciphertexts that it is unauthorised to decrypt, which captures security under thresholdisation. In more detail, our IND-CPA model is as follows:

- The adversary $\mathcal{A}$ statically corrupts a subset $\mathcal{C}$ of size $\leq t - 1$ (instead of exactly $t - 1$) users.
- $\mathcal{A}$ is given the public key and the secret keys of all corrupt users.
- $\mathcal{A}$ has access to a challenge oracle ChalO , an encryption oracle EncO , an evaluation oracle EvalO , and a partial decryption oracle ParDecO , which do the following:
 - ChalO encrypts one of the two challenge messages μ_0, μ_1 chosen by $\mathcal{A}$.
 - EncO encrypts any message μ chosen by $\mathcal{A}$.
 - EvalO allows $\mathcal{A}$ to instruct any homomorphic evaluation on any set of ciphertexts that have been honestly generated (and possibly evaluated) by the experiment.
 - ParDecO adaptively answers partial decryption queries of any user on any honestly generated (and possibly evaluated) ciphertext, with a condition: for each ciphertext whose underlying plaintext is different in games 0 and 1, the number of partial decryption queries is less than $t - |\mathcal{C}|$.

Relative to prior works [BGG⁺18, BS23, OT25, BFM⁺25, KLL⁺26], the main distinction lies in ParDecO , where we model security even when giving $< t - |\mathcal{C}|$ (instead of 0) partial decryptions for each “challenge ciphertext”¹⁰ to $\mathcal{A}$.

Security analysis. We show (Theorem 3) that under appropriate parameter constraints, if the underlying FHE has linear decryption and is IND-CPA-with-Leakage secure, and if the standard LWE assumption holds, then our threshold FHE transformation satisfies the above IND-CPA security. Since IND-CPA-with-Leakage is implied by plain IND-CPA up to appropriate masking noise (Theorem 1), overall, our threshold FHE can be based on any IND-CPA secure FHE with linear decryption and the LWE assumption (Corollary 1).

Our security proof goes via a hybrid argument consisting of two main parts, corresponding to the “threshold layer” and the “FHE layer” of the construction.

Threshold Layer. The first part ($\text{Hyb}_{b,0} \approx_c \text{Hyb}_{b,1}$ via Lemma 12 and $\text{Hyb}_{b,1} \equiv \text{Hyb}_{b,2}$ via Lemma 13) unwraps the threshold layer. We hop from the real experiment, where the challenge oracle ChalO encrypts message μ_b , to a hybrid experiment with the following main changes:

- The vector $\mathbf{b}$ in the public key which encodes the FHE secret key $\hat{\mathbf{s}}$ (ref. Eq. (2)) is swapped to uniformly random. In exchange, the correlation between $\mathbf{b}$ and all partial decryptions pd_k ’s (which arises naturally from decryption correctness) is embedded in the pd_k ’s.
- For ciphertexts whose underlying plaintexts differ in games 0 and 1 (i.e. the “challenges”), their partial decryptions are swapped to uniformly random.

The proof ideas for this part borrow those in [CLW25] established for tPKE: The first change is by carefully manipulating the underlying linear correlations and invoking LWE along the way; the second change relies on the algebraic structure of the Vandermonde matrix, together with the leverage that only $\leq t - |\mathcal{C}|$ partial decryptions of each “challenge” are handed to the adversary.

Relative to [CLW25], however, we need to handle a new complication: During partial decryption, the component $(\mathbf{A}, \mathbf{b})$ is multiplied with $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \bmod q$ (Eq. (3)), where the top part $\mathbf{G}^{-1}(\hat{\mathbf{c}}_0)$ is (the encoding of) an arbitrary honestly generated (and possibly evaluated) FHE ciphertext. Since we do not assume any specific FHE scheme, the experiment has no guarantee on the distribution of $\hat{\mathbf{c}}_0$. Yet, it needs to adaptively answer partial decryption queries on these – possibly adversarially influenced – ciphertexts. This is in contrast to the tPKE setting of [CLW25] where $\mathbf{x}$ is simply the encryption randomness which the security experiment has full control over.

¹⁰ In the simpler case of threshold PKE, one may easily classify ciphertexts from ChalO as “challenges”, which an adversary is unauthorised to decrypt, and those from EncO as “non-challenges”, see e.g. [DLN⁺21, CLW25, BKW26]. For threshold FHE, because of the freedom of homomorphic evaluation, alternative classification is needed. In our model, any (possibly evaluated) ciphertext is considered “challenge” if the underlying plaintext is different in games 0 and 1, else it is “non-challenge”. This aspect is consistent with the model of [BS23].

We tackle the above issue as follows. First, we utilise the random tag $\mathbf{t}$ appended to the threshold FHE ciphertext, which serves as a rerandomisation (in the form of $\mathbf{G}^{-1}(\mathbf{t})$), allowing us to invoke standard LWE w.r.t. $\mathbf{G}^{-1}(\mathbf{t})$ when analysing the modified partial decryptions. Second, to handle the adaptivity issue, we rely on standard LWE and a super-polynomial masking error, with which we argue the distribution of partial decryptions in an information-theoretic way. The formal security proof, which we provide in the appendices, involves reestablishing the counterparts of all lemmas in [CLW25] with the above modifications incorporated.

FHE Layer. The second part handles partial decryptions for the ciphertexts which an adversary is authorised to fully decrypt and recover the plaintexts. For this, we further rewrite these partial decryptions, this time relying on the linear decryption (Eq. (1)) of the underlying FHE ($\text{Hyb}_{b,2} \equiv \text{Hyb}_{b,3}$ via Lemma 14), such that they can be simulated given the leakages from the IND-CPA-with-Leakage experiment of the FHE ($\text{Hyb}_{0,3} \approx_c \text{Hyb}_{1,3}$ via Lemma 15).

Optimisations. As in [BFM⁺25, CLW25, KLL⁺26], to support t -out-of- K decryption, our transformation utilises ξ -subtractive sets [AL21, KLNO24b] of cardinality K , where $\xi = \xi(t) : \mathbb{N} \rightarrow \mathcal{R}$ is a slack function. In brief, if $\Xi \subset \mathcal{R}$ is a ξ -subtractive set, then for any t -subset $T \subseteq \Xi$ which defines a Vandermonde matrix $\mathbf{V}_T$, it holds that $\xi(t) \cdot \mathbf{V}_T^{-1}$ is integral, i.e. has entries in $\mathcal{R}$. This allows choosing the modulus q such that $\xi(t) \cdot \mathbf{V}_T^{-1}$ has low norm relative to q . Since the norm of $\xi(t) \cdot \mathbf{V}_T^{-1}$ is empirically only moderate-sized and especially small with small thresholds, this means the modulus q can also be kept practically small, leading to the concrete efficiency of our threshold FHE. To further compress partial decryption size, we also adopt the standard bit-dropping technique, where the erroneous low-order bits are dropped.

Finally, in our formal security proof, we use an intermediate computational assumption called Leaky LWE to modularise proof steps and ease presentation. In more detail, [LSW25] and [HKM⁺26] gave reductions from standard LWE to the semi-adaptive and fully-adaptive versions of Leaky LWE, respectively, both in the polynomial modulus-to-noise ratio regime. The security proof of **Tulva** (Appendices G.3 and G.4) assumes fully-adaptive Leaky LWE (implied by LWE) and uses proof techniques adapted from the semi-adaptive reduction of [LSW25] in a non-black-box way.

3 Preliminaries

Let $\lambda \in \mathbb{N}$ and $\delta \in \mathbb{N}$ denote the computational and statistical security parameters respectively. For any $n \in \mathbb{N}$, we write $\text{poly}(n)$ and $\text{negl}(n)$ for the sets of all polynomials and negligible functions in n , respectively. For $k, n \in \mathbb{N}$, $k \leq n$, we write $[n]$ for $\{1, \dots, n\}$ and $[k, n]$ for $\{k, \dots, n\}$. If S is a set, we write $x \leftarrow S$ for sampling a uniformly random element from S . If $\mathcal{D}$ is a distribution over S , denoted as $\mathcal{D} \sim S$, we write $x \leftarrow \mathcal{D}$ for sampling a random element from S according to the distribution $\mathcal{D}$. We use bold capital and lower-case letters, e.g. $\mathbf{A}$ and $\mathbf{b}$, to denote matrices and vectors, respectively.

Let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field of conductor $\mathfrak{f}$ and degree $\varphi = \varphi(\mathfrak{f})$ and $\mathcal{R} = \mathbb{Z}[\zeta]$ its ring of integers. For $q \in \mathbb{N}$, define the quotient ring $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. We denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ the sets of units in $\mathcal{R}$ and $\mathcal{R}_q$ respectively.

For any $x \in \mathbb{R}$, we write $\lfloor x \rfloor$ for the nearest integer to x with an arbitrary fixed tie-breaking rule. The fractional part of x is defined as $\{x\} := x - \lfloor x \rfloor$. This notation is extended naturally to vectors of real numbers. For $x \in \mathbb{Z}$ and $z \in \mathbb{N}$, we write $\lfloor x \rfloor_{2^z} := 2^z \cdot \lfloor x/2^z \rfloor$ for x rounded to the nearest multiple of 2^z . For $x \in \mathbb{Z}_q$, $\lfloor x \rfloor_{2^z}$ is computed on the representative of x in $[-q/2, q/2)$, i.e. the one with the smallest absolute value, and the result is read again in $\mathbb{Z}_q$. The notation extends coefficient-wise to ring elements and vectors. We abuse the notation of modular arithmetic so that it expresses two meanings depending on the context: e.g. $a = b \bmod q$ means a and b are congruent modulo q but $a \bmod q = b$ means b is the smallest (in absolute value) of all integers congruent to a modulo q . Relatedly, $\frac{p}{q} \cdot (x \bmod q)$ means multiplying the smallest representative of $x \bmod q$ by p/q . Note that $\left\{ \frac{q}{p} \right\} = \frac{q \bmod p}{p}$.

Embeddings and norms. We write $\psi : \mathcal{K} \rightarrow \mathbb{Q}^\varphi$ and $\sigma = (\sigma_i)_{i \in \mathbb{Z}_f^\times} : \mathcal{K} \rightarrow \mathbb{C}^\varphi$ for the coefficient (wrt. the standard power basis) and canonical embeddings of $\mathcal{K}$, respectively. For $x \in \mathcal{K}$, its ℓ_p -norm is measured via its coefficient embedding, i.e. $\|x\|_p := \|\psi(x)\|_p$, and $p = \infty$ when omitted. If $\mathcal{K}$ is a power-of-two field, then it is known that $\|\psi(x)\| \leq \|\sigma(x)\|$ and $\|\sigma(x)\|_2 = \sqrt{\varphi} \cdot \|\psi(x)\|_2$. The field expansion factor of $\mathcal{K}$ w.r.t. the coefficient ℓ_p -norm is defined as $\mathfrak{c}_p := \max_{a, b \in \mathcal{K}} \|a \cdot b\|_p / (\|a\|_p \cdot \|b\|_p)$.

For a vector $\mathbf{x} \in \mathcal{K}^m$, the embedding is extended component-wise, e.g. $\psi(\mathbf{x}) = (\psi(x_i))_{i \in [m]}$, and $\|\mathbf{x}\|_p := \|\psi(\mathbf{x})\|_p$, and $p = \infty$ when omitted. For a matrix $\mathbf{M} \in \mathcal{K}^{n \times m}$, we write $\Psi(\mathbf{M}) \in \mathbb{Q}^{\varphi n \times \varphi m}$ for the matrix of the $\mathbb{Q}$ -linear map $\mathbf{x} \mapsto \mathbf{M}\mathbf{x}$ from $\mathcal{K}^m$ to $\mathcal{K}^n$ in the coefficient embedding, i.e. $\psi(\mathbf{M}\mathbf{x}) = \Psi(\mathbf{M}) \cdot \psi(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{K}^m$.

For a matrix $\mathbf{M} = (\mathbf{m}_1, \dots, \mathbf{m}_m) \in \mathcal{K}^{n \times m}$, its ℓ_p -induced-norm is $\|\mathbf{M}\|_{p,p} := \sup_{\mathbf{x} \in \mathcal{K}^m} \|\mathbf{M} \cdot \mathbf{x}\|_p / \|\mathbf{x}\|_p$. Consequently, $\|\mathbf{M}\mathbf{x}\|_p \leq \|\mathbf{M}\|_{p,p} \cdot \|\mathbf{x}\|_p$ and $\|\mathbf{M}\mathbf{N}\|_{p,p} \leq \|\mathbf{M}\|_{p,p} \cdot \|\mathbf{N}\|_{p,p}$. Similarly, by default we write $\|\mathbf{M}\|$ for $\|\mathbf{M}\|_{\infty,\infty} = \|\Psi(\mathbf{M})\|_{\infty,\infty}$.

For a real or complex matrix $\mathbf{B}$, we write $s_{\max}(\mathbf{B})$ for its largest singular value, i.e. the square root of the largest eigenvalue of $\mathbf{B}^* \mathbf{B}$, where $\mathbf{B}^*$ is the conjugate transpose of $\mathbf{B}$, and we define its ℓ_2 -induced norm as $\|\mathbf{B}\|_{2,2} := s_{\max}(\mathbf{B})$. By [LSW25, Lemma 2], for any $\mathbf{M} \in \mathcal{K}^{n \times m}$, it holds that $\|\mathbf{M}\|_{2,2} = \|\Psi(\mathbf{M})\|_{2,2} = s_{\max}(\Psi(\mathbf{M}))$. We note that matrices over $\mathcal{K}$ are not identified as complex matrices: a complex matrix arises from $\mathbf{M}$ only as $\sigma_i(\mathbf{M})$ for some $i \in \mathbb{Z}_f^\times$, obtained by applying the i -th component σ_i of the canonical embedding σ entrywise. In particular, $\|\mathbf{M}\|_{2,2}$ is in general not $s_{\max}(\sigma_i(\mathbf{M}))$ for a fixed i (cf. Lemma 2).

Subtractive sets. We rely on the notion of subtractive sets, proposed by [AL21], further studied by e.g. [KLNO24b], and used by e.g. [CLW25] for constructing threshold PKE.

Definition 1 (Subtractive Sets). For a set $\Xi = \{\mu_1, \dots, \mu_K\} \subseteq \mathcal{R}$, any $t \leq K$, and any subset $T \subseteq \{0\} \cup \Xi$, let $\mathbf{V}_T$ (with implicit dependency on t) denote the (row-style) $T \times t$ Vandermonde matrix, i.e. its rows are indexed by T and row $\mu \in T$ is given by $(1, \mu, \dots, \mu^{t-1})$.

Let $\xi = \xi(t) : [K] \rightarrow \mathcal{R}$. The set Ξ is said to be ξ -subtractive [AL21] if for any $t \in [K]$ and any size- t subset $T \subseteq \Xi$, it holds that $\xi \cdot \mathbf{V}_T^{-1} \in \mathcal{R}^{t \times t}$.

Consider Ξ with $0 \notin \Xi$. Denote by $\mathbf{v}_0^\top := \mathbf{V}_{\{0\}} = (1, 0, \dots, 0)$ the first unit vector. We measure the quality of Ξ by the following “quality measurements”:

$$\begin{aligned} \rho(t) &:= \max_{T \subseteq \Xi} \|\xi \cdot \mathbf{v}_0^\top \cdot \mathbf{V}_T^{-1}\|, & \gamma(t) &:= \max_{T^* \subseteq \Xi, |T^*|=t} \|\xi \cdot \mathbf{V}_{\Xi} \cdot \mathbf{V}_{\{0\} \cup T^*}^{-1}\|_{2,2}, \\ \theta(t) &:= \max_{T^* \subseteq \Xi, |T^*|=t} \|\xi \cdot \mathbf{V}_{\Xi \setminus T^*} \cdot \mathbf{V}_{\{0\} \cup T^*}^{-1} \cdot \mathbf{v}_0\|, \end{aligned} \quad (4)$$

where the top row of $\mathbf{V}_{\{0\} \cup T^*}$ corresponds to the index 0. Here ρ and θ use the default coefficient ℓ_∞ -norm, whereas γ uses the ℓ_2 -induced operator norm $\|\cdot\|_{2,2}$.

The Vandermonde matrix $\mathbf{V}_\Xi$ defined above is equivalent to the share-generating matrix of the t -out-of- K Shamir’s secret sharing with evaluation points in Ξ . The subtractive property ensures that multiplying $\mathbf{V}_T^{-1}$ by ξ clears its denominators for every size- t subset $T \subseteq \Xi$.

The quality measurements provide norm bounds for various scaled interpolation maps and associated coefficient vectors. These bounds control noise growth and the parameters required by our later analysis. The quantity $\rho(t)$ bounds the norm of the ξ -scaled interpolation coefficients used to reconstruct the value at zero from any t shares. It therefore controls the noise growth when **Combine** combines partial decryptions, and hence correctness (Theorem 2). The quantity $\gamma(t)$ bounds the growth in size when interpolating from the value at zero and the evaluations at $t - 1$ points in Ξ to the evaluations at all points in Ξ , scaled by ξ . It appears in the conditions of the security theorem (Theorem 3). Finally, fix the evaluations at any $t - 1$ points in Ξ . Each remaining evaluation, scaled by ξ , can be written as an interpolation coefficient times the value at zero, plus a term determined by the fixed evaluations. The quantity $\theta(t)$ bounds the sizes of these interpolation coefficients. It determines how wide the flooding noise χ must be to hide the FHE decryption noise (cf. Corollary 1). We always have $\theta(t) \leq \gamma(t)$. Upper bounds on these quantities for power-of-two fields are given in Lemma 4 in Appendix B.

Discrete Gaussians. We denote by $D_{\mathcal{R}, \chi}^m$ the (centred) discrete Gaussian distribution over $\mathcal{R}^m$ with parameter χ , i.e. the distribution over $\mathcal{R}^m$ where for all $\mathbf{x} \in \mathcal{R}^m$, the probability mass function $D_{\mathcal{R}, \chi}^m(\mathbf{x}) \propto \rho_\chi(\mathbf{x})$ where $\rho_\chi(\mathbf{x}) := e^{-\pi \|\mathbf{x}\|_2^2 / \chi^2}$ is the Gaussian function of width χ , extended to a set S by $\rho_\chi(S) := \sum_{\mathbf{x} \in S} \rho_\chi(\mathbf{x})$. More generally, for a positive definite $\Sigma \in \mathbb{R}^{\varphi m \times \varphi m}$ we denote by $D_{\mathcal{R}, \sqrt{\Sigma}}^m$ the discrete Gaussian over $\mathcal{R}^m$ with covariance Σ , i.e. the distribution with probability mass function proportional to $e^{-\pi \cdot \psi(\mathbf{x})^\top \Sigma^{-1} \psi(\mathbf{x})}$, which for $\Sigma = \chi^2 \cdot \mathbf{I}_{\varphi m}$ is $D_{\mathcal{R}, \chi}^m$. We recall the smoothing parameter [MR04]: for a full-rank lattice Λ and $\epsilon > 0$, $\eta_\epsilon(\Lambda)$ is the smallest $\chi > 0$ such that $\rho_{1/\chi}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$, where Λ^* is the

dual lattice. Throughout, $\eta_\epsilon(\mathcal{R}^m)$ denotes the smoothing parameter of $\mathcal{R}^m$ identified with $\mathbb{Z}^{\varphi m}$ via the coefficient embedding.

Lattice assumption. We recall the LWE assumption with the matrix $\mathbf{A}$ sampled from a specified, not necessarily uniform distribution $\mathcal{D}_{\mathbf{A}}$. This formulation accommodates both instances used in our security proof: uniform matrices and matrices whose columns are gadget decompositions of uniformly sampled vectors. Their hardness follows from the usual LWE assumption with uniform matrices, using [BLMR13, Theorem 4.3] for the latter.

Definition 2 (LWE). Let $\text{par} = (\mathcal{R}, n, m, t, q, \chi, \mathcal{D}_{\mathbf{A}})$ be parametrised by λ . The (decision) LWE_{par} assumption states that for any PPT adversary $\mathcal{A}$, it holds that $\text{Adv}_{\text{par}, \mathcal{A}}^{\text{LWE}}(\lambda) :=$

$$\left| \Pr \left[\mathcal{A}(\mathbf{A}, \mathbf{B}) = 1 \mid \begin{array}{l} \mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}} \\ \mathbf{R} \leftarrow \mathcal{R}_q^{t \times n}; \mathbf{E} \leftarrow D_{\mathcal{R}, \chi}^{t \times m} \\ \mathbf{B} := \mathbf{R} \cdot \mathbf{A} + \mathbf{E} \bmod q \end{array} \right] - \Pr \left[\mathcal{A}(\mathbf{A}, \mathbf{B}) = 1 \mid \begin{array}{l} \mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}} \\ \mathbf{B} \leftarrow \mathcal{R}_q^{t \times m} \end{array} \right] \right| \leq \text{negl}(\lambda).$$

The parameter t is suppressed if $t = 1$. The parameter $\mathcal{D}_{\mathbf{A}}$ is suppressed if it is the uniform distribution over $\mathcal{R}_q^{n \times m}$. We write $m = \text{poly}(\lambda)$ for the version where $\mathcal{A}$ is given unbounded polynomially many LWE samples, in which case $\mathcal{D}_{\mathbf{A}}$ is the distribution per sample which is over $\mathcal{R}_q^n$.

Gadget Matrix. For any $\beta, q \in \mathbb{N}$ with $\beta \leq q$, we write $\mathbf{g}_{\beta, q}^T := (\beta^0, \beta^1, \dots, \beta^{\lceil \log_{\beta} q \rceil - 1})$ for the gadget vector and $\mathbf{G}_{\beta, q} := \mathbf{I}_n \otimes \mathbf{g}_{\beta, q}^T$ for the gadget matrix respectively [MP12]. We denote by $\mathbf{G}_{\beta, q}^{-1}(\cdot)$ the corresponding balanced β -ary decomposition operation, i.e. it holds $\mathbf{G}_{\beta, q} \cdot \mathbf{G}_{\beta, q}^{-1}(\mathbf{x}) = \mathbf{x}$ and $\|\mathbf{G}_{\beta, q}^{-1}(\mathbf{x})\| \leq \beta/2$ for any $\mathbf{x} \in \mathcal{R}_q^n$.

Fully-Homomorphic Encryption. A fully-homomorphic encryption (FHE) for a message space $\mathcal{M}$ consists of PPT algorithms (Setup, KGen, Enc, Eval, Dec) with the syntax:

$\text{pp} \leftarrow \text{Setup}(1^\lambda)$: The setup algorithm generates the public parameters pp .
 $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{pp})$: The key generation algorithm, on input the public parameters pp , generates the public and secret keys (pk, sk) .
 $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu)$: The encryption algorithm encrypts a message $\mu \in \mathcal{M}$ w.r.t. the public key pk .
 $\text{ctxt}' \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_i)_{i \in I})$: The evaluation algorithm takes the public key pk , a function f , and a set of ciphertexts $(\text{ctxt}_i)_{i \in I}$ indexed by I , and returns an evaluated ciphertext ctxt' .
 $\mu' \leftarrow \text{Dec}(\text{sk}, \text{ctxt})$: The decryption algorithm, on input a secret key sk and ciphertext ctxt , outputs some μ' .

Because of the homomorphic nature of FHE, defining the correctness requirement is not straightforward and multiple proposals have appeared in the literature (e.g. [BCF⁺25, BJSW25, ABMP26]). We state a version that follows the spirit of that in [ABMP26], which formalises correctness as an experiment so as to capture the adaptiveness of an adversary in choosing ciphertext evaluations, except that we further allow the flexibility to decrypt unevaluated ciphertexts.

Definition 3 (FHE Correctness). An FHE is said to have correctness error ε , if for any PPT $\mathcal{A}$, it holds that

$$\Pr[\text{Dec}(\text{sk}, \text{ctxt}) = \mu \mid (\text{sk}, \text{ctxt}, \mu) \leftarrow \text{Correct}_{\text{FHE}, \mathcal{A}}(1^\lambda)] \geq 1 - \varepsilon(\lambda),$$

where $\text{Correct}_{\text{FHE}, \mathcal{A}}$ is defined in Fig. 1. Furthermore, we say that FHE is correct if ε is negligible in λ .

4 Threshold FHE Construction

4.1 Requirements on Underlying FHE

Definition 4 (FHE with Linear Decryption). Let $\xi \in \mathcal{R}$ be coprime to q . An FHE scheme FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε , if the following hold:

$\text{Correct}_{\text{FHE}, \mathcal{A}}(1^\lambda)$	$\text{Correct}_{\text{thFHE}, \mathcal{A}}(1^\lambda)$	$\text{EncO}(\text{id}, \mu \in \mathcal{M})$
$L_{\text{Query}} := \emptyset$	$L_{\text{Query}} := \emptyset$	$\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu)$
$\text{pp} \leftarrow \text{Setup}(1^\lambda)$	$\text{pp} \leftarrow \text{Setup}(1^\lambda)$	$L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu)$
$(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{pp})$	$(\text{pk}, (\text{sk}_k)_{k \in [K]}) \leftarrow \text{KGen}(\text{pp})$	$\text{EvalO}(\text{id}, f, (\text{id}_i)_{i \in I})$
$\text{id} \leftarrow \mathcal{A}^{\text{EncO}, \text{EvalO}}(\text{pp}, \text{pk}, \text{sk})$	$(T, \text{id}) \leftarrow \mathcal{A}^{\text{EncO}, \text{EvalO}}(\text{pp}, \text{pk}, (\text{sk}_k)_{k \in [K]})$	for $i \in I$:
parse $(\text{ctxt}, \mu) \leftarrow L_{\text{Query}}[\text{id}]$	parse $(\text{ctxt}, \mu) \leftarrow L_{\text{Query}}[\text{id}]$	assert $L_{\text{Query}}[\text{id}_i] \neq \emptyset$
return $(\text{sk}, \text{ctxt}, \mu)$	$\text{pd}_k \leftarrow \text{ParDec}(\text{pk}, \text{sk}_k, \text{ctxt}) \quad \forall k \in T$	parse $(\text{ctxt}_i, \mu_i) \leftarrow L_{\text{Query}}[\text{id}_i]$
	if $\mu \neq \text{Combine}(T, (\text{pd}_k)_{k \in T}, \text{ctxt})$	$\text{ctxt}' \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_i)_{i \in I})$
	then return 1	$L_{\text{Query}}[\text{id}] := (\text{ctxt}', f((\mu_i)_{i \in I}))$
	else return 0	

Fig. 1: Correctness experiment for FHE and threshold FHE (Definitions 3 and 6).

1. The secret key space is $\mathcal{R}_q^{\hat{n}}$, i.e. a secret key is a vector $\text{sk} = \hat{\mathbf{s}} \in \mathcal{R}_q^{\hat{n}}$.
2. There exists a deterministic Preproc algorithm, which inputs a ciphertext ctxt and outputs $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \in \mathcal{R}_q^{\hat{n}} \times \mathcal{R}_q \leftarrow \text{Preproc}(\text{ctxt})$.
3. For any PPT $\mathcal{A}$, it holds that

$$\Pr \left[\begin{array}{l} \hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 \bmod q \\ = \xi^{-2} \cdot \lfloor q/p \rfloor \cdot \mu + \hat{e} \\ \wedge \|\hat{e}\| \leq B \end{array} \middle| \begin{array}{l} (\text{sk}, \text{ctxt}, \mu) \leftarrow \text{Correct}_{\text{FHE}, \mathcal{A}}(1^\lambda) \\ \text{parse } \text{sk} = \hat{\mathbf{s}} \\ (\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}) \end{array} \right]$$

$\geq 1 - \varepsilon(\lambda)$, where $\text{Correct}_{\text{FHE}, \mathcal{A}}$ is defined in Fig. 1.

If $\varepsilon(\lambda)$ is negligible, we simply say that FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption.

We will also require the FHE scheme to satisfy a security notion that we call “IND-CPA-with-leakage”. This notion says that, for an FHE scheme with linear decryption, the scheme is IND-CPA secure even when additionally giving out a masked leakage on its error term arising from the linear part of the decryption algorithm. Below, we recall standard IND-CPA and define the above variant.

Definition 5 (FHE IND-CPA(-with-Leakage) Security). Let FHE be an FHE scheme. We define the IND-CPA, and IND-CPA-with- (D, L, χ) -Leakage experiments $\text{Exp}_{\text{FHE}, \mathcal{A}}^b$ in Fig. 2. The IND-CPA and IND-CPA-with- (D, L, χ) -Leakage advantage functions of a PPT $\mathcal{A}$ against FHE are defined as

$$\text{Adv}_{\text{FHE}, \mathcal{A}}^{\text{IND-CPA-wL}}(D, L, \chi)(\lambda) := |\Pr[\text{Exp}_{\text{FHE}, \mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp}_{\text{FHE}, \mathcal{A}}^1(1^\lambda) = 1]|.$$

FHE is IND-CPA secure, respectively IND-CPA-with- (D, L, χ) -Leakage secure, if for any PPT $\mathcal{A}$ it holds that $\text{Adv}_{\text{FHE}, \mathcal{A}}^{\text{IND-CPA-wL}}(D, L, \chi)(\lambda) \leq \text{negl}(\lambda)$.

It is easy to see that the IND-CPA-with-Leakage security is implied by standard IND-CPA security if the masking noise for the leakage is large enough, specifically when it is super-polynomially larger than the error term being masked. This follows from a standard noise-flooding lemma, formalised below. Its proof is given in Appendix D.

Theorem 1 (Leakage Security). Let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field of degree φ and expansion factor $\mathbf{c}_\infty$ w.r.t. the coefficient ℓ_∞ -norm. Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be its ring of integers. Let $L, D \in \mathbb{N}$ and FHE be an FHE scheme with $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} . Then, for any PPT adversary $\mathcal{A}$ against the IND-CPA-with- (D, L, χ) -Leakage security of FHE that makes at most $Q \in \text{poly}(\lambda)$ queries to the oracle LeakO, there exists a PPT adversary $\mathcal{B}$ against the IND-CPA security of FHE such that

$$\text{Adv}_{\text{FHE}, \mathcal{A}}^{\text{IND-CPA-wL}}(D, L, \chi)(\lambda) \leq 2 \cdot \left(Q \cdot \varepsilon_{\text{FHE}} + \frac{C \cdot \sqrt{Q} \cdot \varphi \cdot \bar{L} \cdot \mathbf{c}_\infty \cdot D \cdot B}{\chi} \right) + \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda)$$

where $C = \sqrt{\pi/2}$ is a constant. In particular, if $\chi > \lambda^{\omega(1)} \cdot \sqrt{Q} \cdot \varphi \cdot \bar{L} \cdot \mathbf{c}_\infty \cdot D \cdot B$, FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption and is IND-CPA secure, then it is also IND-CPA-with- (D, L, χ) -Leakage secure.

$\text{Exp}_{\text{FHE},\mathcal{A}}^b(1^\lambda)$ $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ $(\text{pk}, \text{sk}) \leftarrow \text{KGen}(\text{pp})$ $L_{\text{Query}} := \emptyset; L_{\text{Leak}} := \emptyset$ $b' \leftarrow \mathcal{A}^{\text{ChalO}, \text{EncO}, \text{EvalO}, \text{LeakO}}(\text{pp}, \text{pk})$ return b' $\text{LeakO}(\text{id}, \mathbf{u} \in \mathcal{R}^L)$ if $L_{\text{Leak}}[\text{id}] = \emptyset$ then assert $\ \mathbf{u}\ \leq D$ // norm check assert $L_{\text{Query}}[\text{id}] \neq \emptyset$ // query id honestly generated parse $(\text{ctxt}, \mu_0, \mu_1) \leftarrow L_{\text{Query}}[\text{id}]$ assert $\mu_0 = \mu_1$ // same plaintext in games 0 and 1 $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt})$ $\mu := \mu_0 = \mu_1$ $\hat{\mathbf{e}} := \hat{\mathbf{c}}_1 - \text{sk}^T \cdot \hat{\mathbf{c}}_0 - \xi^{-2} \left\lfloor \frac{q}{p} \right\rfloor \mu \bmod q$ $\mathbf{f} \leftarrow \mathcal{D}_{\mathcal{R},\chi}^L$ $L_{\text{Leak}}[\text{id}] \leftarrow \mathbf{u} \cdot \hat{\mathbf{e}} + \mathbf{f}$ // error $\mathbf{u} \cdot \hat{\mathbf{e}}$ masked by fresh $\mathbf{f}$ return $L_{\text{Leak}}[\text{id}]$	$\text{Exp}_{\text{thFHE},\mathcal{A}}^b(1^\lambda)$ $\text{pp} \leftarrow \text{Setup}(1^\lambda)$ $\mathcal{C} \leftarrow \mathcal{A}(\text{pp})$ // corrupt set $\mathcal{C}$ assert $\mathcal{C} \subset_{<t} [K]$ $(\text{pk}, (\text{sk}_k)_{k \in [K]}) \leftarrow \text{KGen}(\text{pp})$ $L_{\text{Query}} := \emptyset; L_{\text{Share}} := \emptyset; L_{\text{Party}} := \emptyset$ $b' \leftarrow \mathcal{A}^{\text{EncO}, \text{ChalO}, \text{EvalO}, \text{ParDecO}}(\text{pk}, (\text{sk}_k)_{k \in \mathcal{C}})$ assert $ L_{\text{Share}} \leq Q$ // partial decryptions for $\leq Q$ ciphertexts return b' $\text{ParDecO}(\text{id}, k \in [K])$ assert $L_{\text{Query}}[\text{id}] \neq \emptyset$ // queries id honestly generated $L_{\text{Party}}[\text{id}] := L_{\text{Party}}[\text{id}] \cup \{k\}$ parse $(\text{ctxt}, \mu_0, \mu_1) \leftarrow L_{\text{Query}}[\text{id}]$ if $\mu_0 \neq \mu_1$ then assert $ L_{\text{Party}}[\text{id}] \cup \mathcal{C} < t$ // if different plaintext in games 0 and 1, // unauthorised to decrypt if $L_{\text{Share}}[\text{id}] = \emptyset$ then $\text{pd}_k \leftarrow \text{ParDec}(\text{pk}, \text{sk}_k, \text{ctxt}) \quad \forall k \in [K]$ $L_{\text{Share}}[\text{id}] := (\text{pd}_k)_{k \in [K]}$ parse $(\text{pd}_k)_{k \in [K]} \leftarrow L_{\text{Share}}[\text{id}]$ return pd_k	$\text{ChalO}(\text{id}, \mu_0, \mu_1)$ if $L_{\text{Query}}[\text{id}] = \emptyset$ then $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu_b)$ $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu_0, \mu_1)$ return $L_{\text{Query}}[\text{id}]$ $\text{EncO}(\text{id}, \mu)$ if $L_{\text{Query}}[\text{id}] = \emptyset$ then $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu)$ $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu, \mu)$ return $L_{\text{Query}}[\text{id}]$ $\text{EvalO}(\text{id}, f, (\text{id}_i)_{i \in I})$ if $L_{\text{Query}}[\text{id}] = \emptyset$ then for $i \in I$: assert $L_{\text{Query}}[\text{id}_i] \neq \emptyset$ // queries id_i honestly generated parse $(\text{ctxt}_i, \mu_{0,i}, \mu_{1,i}) \leftarrow L_{\text{Query}}[\text{id}_i]$ $\text{ctxt}' \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_i)_{i \in I})$ $\mu'_0 := f((\mu_{0,i})_{i \in I})$ $\mu'_1 := f((\mu_{1,i})_{i \in I})$ $L_{\text{Query}}[\text{id}] := (\text{ctxt}', \mu'_0, \mu'_1)$ return $L_{\text{Query}}[\text{id}]$
---	---	--

Fig. 2: IND-CPA(-with- (D, L, χ) -Leakage) experiments $\text{Exp}_{\text{FHE},\mathcal{A}}^b(1^\lambda)$ for FHE (Definition 5), and IND-CPA experiment $\text{Exp}_{\text{thFHE},\mathcal{A}}^b(1^\lambda)$ for threshold FHE (Definition 7). Code in [dashed box], i.e. the oracles EncO , EvalO and LeakO , applies only to IND-CPA-with- (D, L, χ) -Leakage. Code in gray for the Q -bounded version of the experiment.

4.2 Threshold FHE Definitions

A t -out-of- K threshold fully-homomorphic encryption, or (t, K) -threshold FHE, for a set of K parties, threshold $t \leq K$, and a message space $\mathcal{M}$, consists of PPT algorithms (Setup , KGen , Enc , Eval , ParDec , Combine) with the syntax:

- $\text{pp} \leftarrow \text{Setup}(1^\lambda)$: The setup algorithm generates the public parameters pp .
- $(\text{pk}, (\text{sk}_k)_{k \in [K]}) \leftarrow \text{KGen}(\text{pp})$: The key generation algorithm, on input the public parameters pp , generates the public key pk and a tuple of K secret keys $(\text{sk}_k)_{k \in [K]}$ for each user $k \in [K]$.
- $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu)$: The encryption algorithm encrypts a message $\mu \in \mathcal{M}$ w.r.t. the public key pk .
- $\text{ctxt}' \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_i)_{i \in I})$: The evaluation algorithm takes the public key pk , a function f , and a set of ciphertexts $(\text{ctxt}_i)_{i \in I}$ indexed by I , and returns an evaluated ciphertext ctxt' .
- $\text{pd}_k \leftarrow \text{ParDec}(\text{pk}, \text{sk}_k, \text{ctxt})$: The partial decryption algorithm takes the secret key sk_k of a user k and a ciphertext ctxt , and outputs a partial decryption pd_k .
- $\mu' \leftarrow \text{Combine}(\text{pk}, T, (\text{pd}_k)_{k \in T}, \text{ctxt})$: The combine algorithm, on input a tuple of partial decryptions $(\text{pd}_k)_{k \in T}$ from a set T of t users and a ciphertext ctxt , outputs some μ' .

Analogous to a (non-threshold) FHE, we define correctness of a threshold FHE by an experiment, in order to capture the adaptive nature of an adversary in choosing ciphertext evaluations and decryption.

Definition 6 (Threshold FHE Correctness). A (t, K) -threshold FHE thFHE is said to have correctness error $\varepsilon(\lambda)$, if for any PPT $\mathcal{A}$,

$$\Pr[\text{Correct}_{\text{thFHE},\mathcal{A}}(1^\lambda) = 1] \leq \varepsilon(\lambda),$$

where $\text{Correct}_{\text{thFHE},\mathcal{A}}$ is defined in Fig. 1. Furthermore, we say that thFHE is correct if ε is negligible in λ .

Setup(1^λ)	Enc(pk, μ)	Eval(pk, $f, (\text{ctxt}_i)_{i \in I}$)
$\text{pp}_{\text{FHE}} \leftarrow \text{FHE.Setup}(1^\lambda)$ return $\text{pp} := \text{pp}_{\text{FHE}}$	parse $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ $\text{ctxt}_{\text{FHE}} \leftarrow \text{FHE.Enc}(\text{pk}_{\text{FHE}}, \mu)$ $\mathbf{t} \leftarrow \mathcal{R}_q^h$ return $\text{ctxt} := (\text{ctxt}_{\text{FHE}}, \mathbf{t})$	parse $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ parse $\text{ctxt}_i = (\text{ctxt}_{\text{FHE}, i}, \mathbf{t}_i) \quad \forall i \in I$ $\text{ctxt}'_{\text{FHE}} \leftarrow \text{FHE.Eval}(\text{pk}_{\text{FHE}}, f, (\text{ctxt}_{\text{FHE}, i})_{i \in I})$ $\mathbf{t} \leftarrow \mathcal{R}_q^h$ return $\text{ctxt}' := (\text{ctxt}'_{\text{FHE}}, \mathbf{t})$
KGen(pp)	ParDec(pk, s_k , ctxt)	Combine(pk, $T, (\text{pd}_k)_{k \in T}, \text{ctxt}$)
parse $\text{pp} = \text{pp}_{\text{FHE}}$ $(\text{pk}_{\text{FHE}}, \text{sk}_{\text{FHE}}) \leftarrow \text{FHE.KGen}(\text{pp}_{\text{FHE}})$ $\text{sk}_{\text{FHE}} = \hat{\mathbf{s}} \in \mathcal{R}_q^{\hat{n}}$ $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ $\mathbf{R} \leftarrow \mathcal{R}_q^{t \times n}$ $\mathbf{r}^\top := \mathbf{v}_0^\top \cdot \mathbf{R} \bmod q$ $\mathbf{e} \leftarrow D_{\mathcal{R}, \bar{\chi}}^m$ $\mathbf{z}^\top := (\hat{\mathbf{s}}^\top, \mathbf{0}_h^\top) \cdot \mathbf{G}_{\beta, q} \bmod q \in \mathcal{R}_q^m$ $\mathbf{b}^\top := \mathbf{r}^\top \mathbf{A} + \mathbf{e}^\top + \mathbf{z}^\top \bmod q$ $\text{pk} := (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ $(\mathbf{s}_k)_{k \in [K]} := \xi \cdot \mathbf{V} \cdot \mathbf{R} \bmod q$ $(\text{sk}_k)_{k \in [K]} := (\mathbf{s}_k)_{k \in [K]}$ return $(\text{pk}, (\text{sk}_k)_{k \in [K]})$	parse $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ parse $\text{sk}_k = \mathbf{s}_k$ parse $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$ $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{FHE.Preproc}(\text{ctxt}_{\text{FHE}})$ $\mathbf{x} := \mathbf{G}_{\beta, q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$ $\mathbf{c}_0 := \mathbf{A} \cdot \mathbf{x} \bmod q$ $e_k \leftarrow D_{\mathcal{R}, \chi}$ $\text{pd}_k := \mathbf{s}_k^\top \cdot \mathbf{c}_0 + e_k \bmod q$ $\text{pd}_k \leftarrow \lfloor \text{pd}_k \rfloor_{2^z}$ return pd_k	parse $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ parse $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$ $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{FHE.Preproc}(\text{ctxt}_{\text{FHE}})$ $\mathbf{x} := \mathbf{G}_{\beta, q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$ $\mathbf{c}_1 := \xi^2 \cdot \mathbf{b}^\top \cdot \mathbf{x} \bmod q$ $\text{pd} \leftarrow \xi \cdot \mathbf{v}_0^\top \cdot \mathbf{V}_T^{-1} \cdot (\text{pd}_k)_{k \in T} \bmod q$ $\hat{\mu} := \xi^2 \cdot \hat{\mathbf{c}}_1 - (\mathbf{c}_1 - \text{pd}) \bmod q \in \mathcal{R}_q$ $\mu' \leftarrow \lfloor p/q \cdot \hat{\mu} \rfloor \bmod p \in \mathcal{R}_p$ return μ'

Fig. 3: thFHE from FHE. $m = (\hat{n} + h) \cdot \lceil \log_{\beta} q \rceil$.

Definition 7 (Threshold FHE IND-CPA Security). A (t, K) -threshold FHE thFHE is (resp. Q -bounded) IND-CPA secure, if for any PPT adversary $\mathcal{A}$ it holds that

$$\text{Adv}_{\text{thFHE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) := \left| \Pr[\text{Exp}_{\text{thFHE}, \mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp}_{\text{thFHE}, \mathcal{A}}^1(1^\lambda) = 1] \right| \leq \text{negl}(\lambda),$$

where $\text{Exp}_{\text{thFHE}, \mathcal{A}}^b$ (resp. augmented with *code in gray*) is defined in Fig. 2.

Remark 3 (Alternative definition for threshold FHE security.). In Definition 7, more specifically in the security experiment defined in Fig. 2, the oracles keep track of ciphertexts ctxt by their identifiers id . An alternative would be to track ciphertexts ctxt by (the representation of) the ciphertexts ctxt themselves. This alternative definition is polynomially equivalent to that stated in Fig. 2 if ciphertexts are collision resistant, i.e. for honestly generated pk , it is infeasible to produce the same ciphertext ctxt by two distinct sequences of $\text{Enc}(\text{pk}, \cdot)$ and $\text{Eval}(\text{pk}, \cdot, \cdot)$ operations with honestly sampled randomness. This alternative definition has some practical benefits, to be elaborated in Remark 6.

4.3 Threshold FHE from FHE

Construction. Let $1 < p \leq q$ be integers. We require two ingredients:

- FHE is an FHE scheme which has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} (Definition 4).
- Ξ is a ξ -subtractive set of cardinality K , with quality measurements $(\rho, \gamma, \theta) = (\rho(t), \gamma(t), \theta(t))$ (Definition 1).

Given the above, in Fig. 3 we construct a (t, K) -threshold FHE scheme $\text{thFHE}_{\text{par}}$ with parameters $\text{par} = (\text{FHE}, \mathcal{R}, n, m, h, q, \beta, \bar{\chi}, \chi, z, \Xi)$.

Theorem 2 (Correctness). If $\xi \in \mathbb{Z}$, FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} and

$$\frac{q}{2p} \geq \frac{|q \bmod p|}{2} + \xi^2 \cdot B + \epsilon_\infty \cdot \left(\sqrt{\lambda} \cdot m \cdot \xi^2 \cdot \bar{\chi} \cdot \frac{\beta}{2} + t \cdot \rho \cdot (\sqrt{\lambda} \cdot \chi + 2^{z-1}) \right). \quad (5)$$

Then, thFHE in Fig. 3 has correctness error $\varepsilon \leq \varepsilon_{\text{FHE}} + \varphi \cdot (m+t) \cdot 2^{-\lambda}$. In particular, if $\min\{\log(1/\varepsilon_{\text{FHE}}), \lambda - \log \varphi - \log(m+t)\} \geq \omega(\log \lambda)$, then thFHE is correct.

The proof of Theorem 2 is given in Appendix E. Here we provide a brief sketch of the proof.

Proof (Proof Sketch). The proof compares the value reconstructed by threshold decryption with the linear-decryption equation of the underlying FHE scheme. Both ParDec and Combine compute the same gadget preimage

$$\mathbf{x} = \mathbf{G}_{\beta,q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}.$$

For a recovery set T , linearly combining the partial decryptions recovers the shared secret $\mathbf{r}$ and gives

$$\mathbf{pd} = \xi^2 \mathbf{r}^T \mathbf{A} \mathbf{x} + \xi \mathbf{v}_0^T \mathbf{V}_T^{-1} \mathbf{e}_T \bmod q.$$

On the other hand, the corresponding ciphertext component satisfies

$$c_1 = \xi^2 \mathbf{r}^T \mathbf{A} \mathbf{x} + \xi^2 \mathbf{e}^T \mathbf{x} + \xi^2 \hat{\mathbf{s}}^T \hat{\mathbf{c}}_0 \bmod q.$$

Hence $c_1 - \mathbf{pd}$ cancels the common term $\xi^2 \mathbf{r}^T \mathbf{A} \mathbf{x}$ and leaves the FHE linear-decryption term, $\xi^2 \hat{\mathbf{s}}^T \hat{\mathbf{c}}_0$, plus a residual error

$$f := \xi^2 \cdot \mathbf{e}^T \cdot \mathbf{x} - \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T.$$

In the final rounding in Combine one has to additionally account for the error $\xi^2 \hat{e}$ arising from the $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption property of FHE. The correctness condition in Eq. (5) is set so that the norm of the whole error term remains below $q/(2p)$, ensuring the scaling by p/q recovers μ except with the stated failure probability. $\square$

Remark 4 (“Average-case” correctness analysis.). In the proof of correctness (Theorem 2) we (probabilistically) upper bound $\|\mathbf{e}^T \cdot \mathbf{x}\|$ by $\mathbf{c}_\infty \cdot m \cdot \bar{\chi} \cdot \sqrt{\lambda} \cdot \beta/2$. This is known as the so called “worst-case” correctness analysis. A standard heuristic is the so-called “average-case” correctness analysis where one would upper bound the norm of $\mathbf{e}^T \cdot \mathbf{x}$ directly by treating it as a discrete Gaussian rather than bounding $\mathbf{e}$ and $\mathbf{x}$ separately. Over $\mathbb{Z}$, this would shave off a square-root-dimension factor, i.e. $\sqrt{\varphi} \cdot \bar{m}$ from the bound. Taking the potential skewness of $\mathcal{R}$ into account, however, we conservatively shave a factor of $\frac{\mathbf{c}_\infty}{\mathbf{c}_2} \cdot \sqrt{m} \leq \sqrt{\varphi} \cdot \bar{m}$ instead, i.e. we take $\|\mathbf{e}^T \cdot \mathbf{x}\| \leq \mathbf{c}_2 \cdot \sqrt{m} \cdot \bar{\chi} \cdot \sqrt{\lambda} \cdot \beta/2$. We can apply the same treatment to the term $\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T$ and upper bound its norm by $\mathbf{c}_2 \cdot \sqrt{t} \cdot \rho \cdot (\chi + 2^{z-1}) \cdot \sqrt{\lambda}$ instead of by $\mathbf{c}_\infty \cdot t \cdot \rho \cdot (\chi \cdot \sqrt{\lambda} + 2^{z-1})$. Here $\mathbf{e}_T$ contains the rounding errors d_k of ParDec, and the heuristic treats each $e_k + d_k$ as noise of width at most $\chi + 2^{z-1}$. To summarise, with the “average-case” heuristic, we can replace the constraint Eq. (5) with

$$\frac{q}{2p} \geq \frac{|q \bmod p|}{2} + \xi^2 \cdot B + \mathbf{c}_2 \cdot \sqrt{\lambda} \cdot \left(\sqrt{m} \cdot \xi^2 \cdot \bar{\chi} \cdot \frac{\beta}{2} + \sqrt{t} \cdot \rho \cdot (\chi + 2^{z-1}) \right). \quad (6)$$

Theorem 3 (Security). *Let FHE be an FHE scheme with $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption and Ξ be a ξ -subtractive set of cardinality K with quality measurements (ρ, γ, θ) (Definition 1). Let $\epsilon \in (0, 1/2]$, $Q \in \text{poly}(\lambda)$, q, ξ be coprime, $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$ be such that there exists $\hat{\chi} > 0$ with $\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2)^{-1} - \hat{\chi}^{-2}}$, and $\chi_2^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$ be such that $\gamma \cdot \sqrt{\varphi m} \cdot \beta/2 \geq 1$ and $\sqrt{2QK} \cdot \varphi m \cdot \mathbf{c}_2 \cdot \mathbf{c}_t^\otimes \cdot \gamma \cdot \beta/2 \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi_2^*)^2)^{-1} - \bar{\chi}^{-2}}$, where $\mathbf{c}_t^\otimes := 1$ if t is a power of two and $\mathbf{c}_t^\otimes := \sqrt{t}$ otherwise. Let $\mathcal{D}_t := \left\{ \tilde{\mathbf{t}} : \mathbf{t} \leftarrow \mathcal{R}_q^h, \tilde{\mathbf{t}} := \mathbf{G}_{\beta,q}^{-1}(\mathbf{t}) \right\}$. Define the parameter sets:*

- thFHE: $\text{par} = (\text{FHE}, \mathcal{R}, n, m, h, q, \beta, \bar{\chi}, \chi, z, \Xi)$,
- LWE: $\text{par}_1 := (\mathcal{R}, h \lceil \log_\beta q \rceil, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_t)$,
- LWE: $\text{par}_2 := (\mathcal{R}, n, m, q, \chi_2^*)$,
- FHE: $\text{par}_3 := (\theta, K, \chi)$.

Consider the (t, K) -threshold FHE scheme $\text{thFHE}_{\text{par}}$ in Fig. 3. For all PPT $\mathcal{A}$ that requests partial decryptions on at most Q ciphertexts (i.e. in the Q -bounded IND-CPA experiment of Definition 7), there exist PPT $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that $\text{Adv}_{\text{thFHE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) \leq \epsilon$ where

$$\begin{aligned} \epsilon := & 2 \cdot t \cdot \left(\text{Adv}_{\text{par}_1, \mathcal{B}_1}^{\text{LWE}}(\lambda) + \text{Adv}_{\text{par}_2, \mathcal{B}_2}^{\text{LWE}}(\lambda) \right) + \frac{12 \cdot Q \cdot \epsilon}{1 - \epsilon} + \text{Adv}_{\text{FHE}, \text{par}_3, \mathcal{B}_3}^{\text{IND-CPA-wL}}(\lambda) \\ & + 4 \cdot t \cdot \left(2QK + \frac{3}{1 - \epsilon} \right) \cdot \epsilon. \end{aligned}$$

The proof of Theorem 3 is given in Appendix F. Here we provide a brief sketch of the proof.

Proof (Proof Sketch). Since the rounding $\lfloor \cdot \rfloor_{2z}$ in ParDec is a deterministic public post-processing of each partial decryption, it suffices to prove security for the scheme without it. The proof proceeds through hybrids, $\text{Hyb}_{b,i}$ for $i = 0, 1, 2, 3$, that first remove the threshold layer and then reduce the remaining indistinguishability to the security of the underlying FHE. Starting from the real experiment $\text{Hyb}_{b,0}$, the first hop to $\text{Hyb}_{b,1}$ rewrites $\mathbf{b}$ and the partial decryptions pd_k so that the threshold-layer correlations are simulated explicitly. This step is justified by standard LWE: the term $\text{Adv}_{\text{par}_1, \mathcal{B}_1}^{\text{LWE}}(\lambda)$ comes from the tag-dependent rerandomisation via $\mathbf{G}^{-1}(\mathbf{t})$, while $\text{Adv}_{\text{par}_2, \mathcal{B}_2}^{\text{LWE}}(\lambda)$ comes from replacing the key-switching-key component $\mathbf{b}$. The two statistical terms in ε are losses involved in the reductions from LWE.

The next hop, $\text{Hyb}_{b,1} \equiv \text{Hyb}_{b,2}$, replaces the partial decryptions of challenge ciphertexts by uniform values. This uses the Vandermonde structure induced by the ξ -subtractive set Ξ and the fact that $\mathcal{A}$ receives fewer than $t - |\mathcal{C}|$ honest partial decryptions for every ciphertext whose plaintext differs between games 0 and 1.

Finally, $\text{Hyb}_{b,2} \equiv \text{Hyb}_{b,3}$ handles ciphertexts whose plaintext is the same in both games. Using the linear-decryption equation from Definition 4, the corresponding partial decryptions are rewritten so that the only remaining dependence on $\text{sk}_{\text{FHE}} = \hat{\mathbf{s}}$ is through the FHE decryption noise $\hat{e}$. These values are simulated by the leakage oracle in the IND-CPA-with-Leakage experiment, giving the term $\text{Adv}_{\text{FHE}, \text{par}_3, \mathcal{B}_3}^{\text{IND-CPA-wL}}(\lambda)$. At this point, challenge shares are uniform and non-challenge shares are simulated from the leakage oracle, so summing the hybrid losses gives the stated bound ε . $\square$

Remark 5 (Relation to IND-CPA^D security of FHE). By setting $K = t = 1$, so that thFHE collapses to a plain FHE scheme, we recover from Definition 7 the IND-CPA^D security notion for FHE defined in [LM21]. This means that our construction thFHE in Fig. 3 is also a compiler which turns any FHE satisfying the requirements in Section 4.1 into an FHE that is IND-CPA^D secure [LM21].

Remark 6 (Pseudorandom tag and context-dependent decryption). In our construction in Fig. 3, the tag $\mathbf{t}$ in a ciphertext $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$ is randomly sampled during encryption and refreshed during homomorphic evaluation, independent of the main part ctxt_{FHE} of the ciphertext. If the underlying FHE has collision-resistant ciphertexts in the sense described in Remark 3, then our construction would be secure wrt. the alternative security definition described in Remark 3 where the oracles keep track of ciphertexts via their own representations rather than identifiers. This means that the security proof would also go through, in the random oracle model, if we replace the truly random tag $\mathbf{t} \leftarrow \mathcal{R}_q^h$ by a pseudorandom $\mathbf{t} := \mathcal{H}(\text{ctxt}_{\text{FHE}})$ where $\mathcal{H}$ is a hash function modelled as a random oracle. Deriving tags in a pseudorandom manner has the obvious advantages of reducing ciphertext size and more importantly reducing use of randomness, which could be scarce in constrained devices. Furthermore, we can easily upgrade our construction to support context-dependent decryption [BBN⁺25] by hashing ctxt_{FHE} together with the context string.

Remark 7 (Optimised Lagrange interpolation in Combine). One optimisation that may not arise naturally from the protocol description concerns Combine, where the message is reconstructed from the t partial decryptions using the (scaled) first row $\xi \cdot \mathbf{v}_0^T \mathbf{V}_T^{-1}$ of the inverse Vandermonde matrix $\mathbf{V}_T$ of the active nodes $\{\zeta_k\}_{k \in T}$. Instead of inverting the $t \times t$ matrix $\mathbf{V}_T$ over $\mathcal{R}_q$, we use that this row is the vector of Lagrange basis polynomials evaluated at 0, namely $(L_k(0))_{k \in T}$ with $L_k(0) = \prod_{j \in T \setminus \{k\}} \zeta_j / (\zeta_j - \zeta_k)$, which costs $O(t^2)$ to form directly. Moreover, since the nodes are exactly the K -th roots of unity, the identity $\prod_{m \in T} (x - \zeta_m) \prod_{m \in T^c} (x - \zeta_m) = x^K - 1$ lets us rewrite each weight as a product over the complement, $L_k(0) = -\frac{1}{K} \prod_{m \in T} (-\zeta_m) \prod_{m \in T^c} (\zeta_k - \zeta_m)$. Evaluating the row over whichever of T or T^c is smaller thus costs $O(t \cdot \min\{t, K - t\})$, keeping Combine cheap for large committees whenever t is close to 1 or to K .¹¹

5 Practical Instantiation

Tulva is a generic construction to build threshold FHE from an FHE scheme. Concretely, its implementation consists of two main components: Preproc and the threshold decryption layer. The first establishes the

¹¹ Asymptotically, the row can be built in quasi-linear time $O(t \log^2 t)$ via fast multipoint evaluation [vzGG13]; this appears less efficient at the committee sizes we consider, but could be relevant for larger ones.

bridge between the FHE scheme and our threshold layer: given a generic FHE ciphertext, it produces a ciphertext satisfying the requirements described in Section 4. This design makes our threshold decryption layer generic wrt. the choice of FHE scheme, at the cost of making **Preproc** scheme dependent. Let us start by discussing potential instantiations of **Preproc** for some of the most popular modern FHE schemes:

BFV/BGV/CKKS [Bra12, FV12, BGV12, CKKS17]: These schemes usually operate over rings that can be directly used by our threshold decryption layer, admitting **Preproc** to be just the identity in some scenarios. Definition 4 expects a BFV-style ciphertext, which can be achieved for BGV through simple transforms (see Appendix A of [AP13]) and for (discrete [BCKS24]) CKKS through bootstrapping (e.g., [Kim25]). For the original (approximate) CKKS [CKKS17], on the other hand, an additional challenge is in separating message from noise, a non-trivial security problem [LM21] that afflicts approximate schemes whenever the existence of decryption oracles is considered. This problem is mostly orthogonal to the threshold setting, and considering potential solutions (e.g., [BCD⁺25a]) is left as future work.

FHEW/TFHE [DM15, CGGI16]: These schemes are typically built on top of ring variants [DM15] of AP-style [AP14] bootstrapping for LWE ciphertexts in the form $\mathbb{Z}_q^{\hat{n}} \times \mathbb{Z}_q$, which cannot be directly used by our threshold layer. For these cases, we employ Packing Key Switching [CGGI16, MS18, CDKS21] techniques, which, given a vector of k LWE ciphertexts encrypting messages $\{m_i\}_{i \in [0, k)} \in \mathbb{Z}_q^k$, produces a single ciphertext of the form $\mathcal{R}_q^{\hat{n}} \times \mathcal{R}_q$ (for some ring of degree at least k) encrypting $\sum_{i=0}^{k-1} m_i X^i \in \mathcal{R}_q$. This ciphertext can either be defined over the ring required by the threshold layer, or ring switched [GHPS12] to it. Alternatively, these schemes could also be instantiated with AP-style bootstrapping for RLWE ciphertexts [MS18, GP25], which already perform similar packing by construction.

Remark 8 (Bootstrapping). In addition to structural requirements, our construction also requires the FHE scheme to provide a minimum ratio between ciphertext modulus and noise magnitude at the end of the homomorphic evaluation. For leveled schemes, one could consider increasing parameters to account for it, but this would introduce performance overheads for the entire computation. A more generic and modular solution is to perform a bootstrapping over the results of the evaluation procedure. We do not specify this bootstrapping as part of **Preproc**, as schemes may have different ways of performing it, but, in practice, including a bootstrapping at the beginning of **Preproc** is a natural solution for most scenarios, and we adopt it in our implementation.

5.1 Implementation Details

To demonstrate compatibility with existing FHE schemes, our implementation¹² natively consumes TFHE ciphertexts produced by the **tfhe-rs** library. A thin compatibility layer (written in Rust, with Python bindings) performs the necessary repacking and key-switching within **tfhe-rs**: since **tfhe-rs** operates on LWE ciphertexts over $\mathbb{Z}_q$, up to φ of them are gathered into a single ciphertext over $\mathcal{R}_q$, as required by our threshold layer.

As a first step of **Preproc**, the ciphertext is bootstrapped to reduce the noise magnitude, include the ξ^{-2} slack, and switch to the modulus q required by our construction, which is significantly larger than the 2^{64} modulus **tfhe-rs** typically operates on. We provide our own implementation of the [CGGI16] and [GP25] bootstrappings: the former is generally faster for small plaintext spaces ($p = 2^4$, as used in our examples), at the cost of higher memory usage and a larger public key, while the latter is more efficient for larger plaintext spaces ($p = 2^6$ or $p = 2^8$). We report runtimes only for [CGGI16], since our example parameters use the small plaintext space, but also include [GP25] (with the [CDKS21] repacking technique) for use in other parameter regimes. Before this bootstrapping step, all homomorphic operations are performed in the smaller modulus and require no parameter adaptation compared to a non-threshold setting; we use the parameters suggested by **tfhe-rs** for 2^{-128} security and correctness errors.

The threshold layer is orchestrated in Python, with the most computationally intensive operations implemented in C and exposed via bindings; this keeps the critical code easy to optimise and the overall implementation simple to modify. The C layer is mainly an adaptation of the code given in [CCC⁺25], and Gaussian sampling, used for both key generation and noise flooding, is a generalised and optimised implementation of the one in [LSS24].

We operate on power-of-two cyclotomic rings, the most common in practical FHE implementations, and employ standard RNS and NTT polynomial arithmetic (also known as double-CRT [GHS12]). We

¹² Available at <https://github.com/osdnk/tulva>.

applied standard AVX-512 optimisations, especially in the NTT transforms, while also optimising for cache locality and parallelism; the code remains portable to any platform, with a performance hit where AVX-512 is unavailable.

5.2 Parameter Selection

Summary of constraints. For the convenience of referencing, we summarise Theorems 1 to 3 and Remark 4 in the following Corollary 1. We note that Item 4 is associated to the “average-case” correctness heuristic.

Corollary 1. *Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be of ring degree φ and with field expansion factor $\mathfrak{c}_p$ w.r.t. the coefficient ℓ_p -norm. Let FHE be an FHE scheme. Let Ξ be a ξ -subtractive set of cardinality K with quality measurements (ρ, γ, θ) (Definition 1). Let $\delta = \delta(\lambda)$ be an independent parameter such that $2^\delta \in \lambda^{\omega(1)}$. Suppose $\text{par} = (\text{FHE}, \mathcal{R}, n, m, h, q, \beta, \bar{\chi}, \chi, z, \Xi)$, $\chi_1^*, \chi_2^* > 0$, $\epsilon \in (0, 1/2] \cap \text{negl}(\lambda)$ and $Q \in \text{poly}(\lambda)$ satisfy:*

1. FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption (Definition 4).
2. FHE is IND-CPA secure.
3. q, ξ are coprime.
4. $\frac{|q \bmod p|}{2} + \xi^2 \cdot B + \mathfrak{c}_2 \cdot \sqrt{\lambda} \cdot \left(\sqrt{m} \cdot \xi^2 \cdot \bar{\chi} \cdot \frac{\beta}{2} + \sqrt{t} \cdot \rho \cdot (\chi + 2^{z-1}) \right) + \frac{1}{2} \leq \frac{q}{2p}$ (Eq. (6)).
5. $\chi \geq 2^\delta \cdot \sqrt{Q \cdot \varphi \cdot K} \cdot \mathfrak{c}_\infty \cdot B \cdot \theta$.¹³
6. $\text{LWE}_{\mathcal{R}, h, \text{poly}(\lambda), q, \chi_1^*}$ ¹⁴, $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$, $\hat{\chi} > 0$ and

$$\frac{\gamma^2}{\chi^2} + \frac{1}{\hat{\chi}^2} \leq \frac{1}{2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2}.$$

7. $\text{LWE}_{\mathcal{R}, n, m, q, \chi_2^*}$, $\chi_2^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$, $\gamma \cdot \sqrt{\varphi m} \cdot \beta / 2 \geq 1$, and, with $\mathfrak{c}_t^\otimes := 1$ if t is a power of two and $\mathfrak{c}_t^\otimes := \sqrt{t}$ otherwise,

$$\frac{Q \cdot K \cdot \varphi \cdot m \cdot \mathfrak{c}_2^2 \cdot (\mathfrak{c}_t^\otimes)^2 \cdot \gamma^2 \cdot \beta^2}{2 \cdot \chi^2} + \frac{1}{\bar{\chi}^2} \leq \frac{1}{2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi_2^*)^2}.$$

8. $8 \cdot (3Q + t \cdot QK + 3t) \cdot \epsilon \leq 2^{-\delta}$.¹⁵

Then, the (t, K) -threshold FHE scheme $\text{thFHE}_{\text{par}}$ in Fig. 3 is correct (with “average-case” correctness) and Q -bounded IND-CPA secure. In particular, if par , χ_1^* and χ_2^* are such that conditions 6, 7 and 8 hold for all $Q \in \text{poly}(\lambda)$, then $\text{thFHE}_{\text{par}}$ in Fig. 3 is also (unbounded) IND-CPA secure.

Proof. Follows from putting Theorems 1 to 3 and Remark 4 together: Items 3–4 imply the correctness constraints in Theorem 2 and Remark 4; Items 1, 2, 5 imply that FHE is IND-CPA-with- (D, K, χ) -Leakage secure with $D = \theta$ by Theorem 1; security follows from Items 3, 6, 7, 8 by invoking Theorem 3. $\square$

We search for parameter sets that minimise the public-key $|\text{pk}|$ and partial-decryption $|\text{pd}|$ sizes, while keeping (partial) decryption reasonably efficient, subject to the correctness and security constraints of Theorems 1 to 3 and Remark 4. We set the plaintext modulus $p = 2^4 = 16$, a common choice for FHEW/TFHE-based applications [BBB⁺23], and target $\lambda = 128$ bits of computational security, $\delta = 64$ bits of statistical security, and $Q = 2^{64}$ partial decryption queries. Recall that the underlying FHE must have $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption; we insist on a noise bound $B \leq 2^{70}$ to leave enough room for FHE.Preproc. When searching for parameters using the Lattice Estimator [APS15], we keep the degree ℓ of gadget decomposition low, for faster partial decryption and decryption, and require $\beta \in [2^{35}, 2^{50}]$ so that it corresponds to one “limb” in the RNS decomposition, allowing a more efficient gadget decomposition.

Following this methodology, we found parameter sets for conductors $\mathfrak{f} \in \{512, 4096\}$ and various threshold profiles. The small ring is selected for low latency (a single decryption is fast to compute, with reasonable per-sample efficiency), the large ring for high throughput (when many messages are to be

¹³ We let 2^δ absorb the $\sqrt{\pi/2}$ factor.

¹⁴ This implies $\text{LWE}_{\mathcal{R}, h \lceil \log_\beta q \rceil, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_t}$ where $\mathcal{D}_t = \mathcal{U}([-\beta/2, \beta/2) \cap \mathbb{Z})^h \lceil \log_\beta q \rceil$ [BLMR13, Theorem 4.3].

¹⁵ This makes the statistical terms of Theorem 3 total at most $2^{-\delta}$, using $1/(1 - \epsilon) \leq 2$.

Table 1: **tulva-tfhe** parameters across conductors $f \in \{512, 4096\}$ (ring degree $\varphi = f/2$), $Q = 2^{64}$, $p = 2^4$. A set of width w supports every threshold $t \in [2, w] \cup [K - w, K]$. Here the gadget base β is constrained to a single RNS limb. The column $|\text{ctxt}|$ gives the size of the input ciphertext, i.e. before **Preproc**.

	K	w	λ	q	ℓ	$\hat{n}$	n	h	m	χ	$\bar{\chi}$	$ \text{ctxt} $ [kB]	$ \text{pk} $ [kB]	$ \text{pd} $ [kB]	$ \text{pd} /\text{bit}$ [b]
S.1	8	4	130	2^{210}	5	32	10	1	165	2^{193}	2^{97}	18	1083	0.7	5.5
S.2	32	16	131	2^{235}	5	36	16	3	195	2^{205}	2^{81}	18	1432	1.0	8.3
S.3	128	16	129	2^{300}	6	45	27	9	324	2^{240}	2^{72}	18	3038	2.0	16.0
S.4	128	64	128	2^{329}	7	49	38	17	462	2^{253}	2^{45}	18	4750	2.5	19.8
S.5	512	16	128	2^{368}	8	55	37	17	576	2^{276}	2^{71}	18	6624	3.0	23.8
L.1	8	4	162	2^{215}	5	5	2	1	30	2^{196}	2^{98}	32	1613	5.8	5.8
L.2	32	16	144	2^{240}	5	5	3	1	30	2^{208}	2^{82}	32	1800	8.5	8.5
L.3	128	16	134	2^{308}	7	6	4	2	56	2^{247}	2^{82}	32	4312	16.8	16.8
L.4	128	64	137	2^{336}	7	7	5	3	70	2^{259}	2^{48}	32	5880	20.5	20.5
L.5	512	16	128	2^{376}	8	7	5	3	80	2^{283}	2^{75}	32	7520	24.8	24.8

decrypted). Instead of specifying the threshold t directly, we specify the number of parties K and a window size w , covering all profiles with $t \in \{2, \dots, w\} \cup \{K - w, \dots, K\}$. Since the partial decryption size and time scale with $\min\{t, K - t\}$, a window of size w collects exactly the profiles whose cost is no worse than that at $\min\{t, K - t\} = w$. The resulting parameters are listed in Table 1.

When calculating the size of a public key, we assume that $\mathbf{A}$ in the public key is sampled using a 256-bit seed, and that the public key only stores this seed and the vector $\mathbf{b}$.

5.3 Experiments

All experiments were conducted on a single AWS **c7a.32xlarge** instance, equipped with 128 AMD EPYC 9R14 (Genoa) vCPUs and 256 GB of RAM. For S. [1–5], we considered two types of experiments: decryption of a single message and decryption of φ messages (the full ring), the latter having a substantial impact on the **Preproc** phase. For L. [1–5], we report only the full-ring decryption results, as employing such a large ring for single-message decryption is unreasonable. The reported times are available in Table 2. We do not report the **KGen** time as it was not optimised (it takes at most 36 seconds) and is a one-time cost that can be amortised over many decryptions. We report **Preproc** as a subroutine separate from **ParDec** and **Combine** to highlight the cost of the pre-decryption phase, which is the most expensive part (yet, secret independent).

Table 2 confirms the expected scaling of **tulva-tfhe**. For the small ring, a single decryption is fast across all profiles: the pre-decryption stays below one second and the online cost is at most a few milliseconds of partial decryption followed by a combination step of tens to hundreds of microseconds (e.g. 0.13s, 0.64ms and 39 μ s for S.1). **Preproc** is by far the most expensive step and, as expected, scales with the number of decrypted messages, growing from sub-second for a single message to between 6 and 31s for the full $\varphi = 256$ ring. At the same time, the per-message partial-decryption and combination times stay within a factor of three. Decrypting the full ring therefore amortises this fixed cost over many messages, and the large ring (L. [1–5]) pushes this further by packing $\varphi = 2048$ messages per batch. Finally, the online cost scales with $\min\{t, K - t\}$ rather than by K : for instance L.5 ($t/K = 16/512$) combines in 0.88ms, faster than L.4 ($t/K = 64/128$) at 1.50ms, which stays consistent with Remark 7.

Comparison with [ZZLP25]. The concurrent work of [ZZLP25] also targets LWE threshold decryption, removing the decryption noise through an interactive *MPC rounding* rather than masking it. We compare their secret-dependent online latency for a single ciphertext [ZZLP25, Table 2] with our **ParDec** + (network delay) + **Combine**, matching their party count to our active threshold t (Table 3). For Tulva, the reported latency covers partial decryption, communication, and share combination after public ciphertext preprocessing has completed. Preprocessing is excluded from this comparison and reported separately in Table 2. The results show lower latency for Tulva and a smaller increase as network delay grows, consistent with its single communication round. Their “preparation” phase is interactive and generates secret correlated randomness, whereas our **Preproc** is public, non-interactive, and ciphertext-dependent, and can be performed by an untrusted server. Tulva’s shorter secret-dependent phase comes

Table 2: Runtimes of the threshold layer on the parameter sets of Table 1, using the [CGGI16] bootstrapping.

Set	t/K	msgs	Preproc [s]	ParDec [ms]	Combine [μ s]
S.1	4/8	1	0.13	0.64	39
		φ	5.83	0.51	39
S.2	16/32	1	0.16	0.60	91
		φ	7.70	0.43	77
S.3	16/128	1	0.24	1.11	94
		φ	12.68	0.77	82
S.4	64/128	1	0.36	2.80	223
		φ	22.58	2.98	610
S.5	16/512	1	0.47	3.68	534
		φ	31.03	4.26	562
L.1	4/8	φ	74.17	0.63	70
L.2	16/32	φ	80.08	0.34	403
L.3	16/128	φ	144.95	0.62	967
L.4	64/128	φ	258.87	1.25	1495
L.5	16/512	φ	303.27	1.10	875

Table 3: Secret-dependent online-phase latency for a single ciphertext.

Set	t	ping	[ZZLP25] [ms]	ours [ms]
S.1	4	1 ms	8.48	1.68
S.2	16	1 ms	24.23	1.69
S.1	4	10 ms	55.62	10.68
S.2	16	10 ms	56.98	10.69

at the cost of ciphertext-dependent preprocessing, which takes 0.13s for S.1 and 0.16s for S.2 (Table 2). The preparation phase of [ZZLP25] can instead be completed before the ciphertext becomes available.

6 Examples of FHEVM Programs

Threshold FHE can be used to implement confidential smart contracts: many users encrypt private inputs under one FHE public key, a neutral evaluator (the validator set) applies a public function f to the ciphertexts while learning nothing about the inputs, and a t -of- K committee runs the **ParDec** and **Combine** procedures of the threshold FHE scheme to reveal only a chosen part of the output. The parties that hold the data are thus separated from the party that computes and from the parties that decrypt. Furthermore, the committee composition can change dynamically over time.

We describe four concrete FHEVM programs and report the runtimes of their implementations using **tulva-tfhe**. Detailed descriptions of the functions f are deferred to Appendix A. Since every value occupies a single $\mathbb{Z}_p$ element, each program is correct only under explicit domain constraints that bound its inputs, e.g. the value ranges such as $[0, B)$ below. The exact constraints are given in Appendix A. For each program we report runtimes over the five small-ring parameter sets S.1–S.5 of Table 1: homomorphic evaluation of f (**Eval**), the functional bootstrap that carries each revealed value to the committee (**Preproc**), and the committee’s partial and final decryption (**ParDec**, **Combine**). The three profile-dependent phases (**Preproc**, **ParDec**, **Combine**) are listed per profile and **Eval** is listed once. As in Section 5.3, we omit key generation, a one-time cost that is amortised over many reveals.

1. Sealed-Bid Auction (Table 4). A pool of n users encrypt their bids over $[0, B)$ and submit them to the contract, which homomorphically computes the winning bid and bidder index. The contract reveals the winner and the price at which the auction clears but not the losing bids.

2. Confidential ERC-20 Transfer (Table 5). ERC-20 is the standard smart-contract interface for fungible Ethereum tokens. In our specific example, account balances $[0, B)$ are encrypted, and a transfer

Table 4: Sealed-bid auction (Algorithm 1).

Set	t/K	Eval [ms]	Preproc [ms]	ParDec [ms]	Combine [μ s]
S.1	4/8		275	0.6	42
S.2	16/32		310	0.6	90
S.3	16/128	513	492	1.2	98
S.4	64/128		724	2.7	220
S.5	16/512		1120	4.0	206

Parameters: $n = 4$ bidders, max. bid $B = 4$.

Table 5: Confidential ERC-20 (Algorithm 2).

Set	t/K	Eval [s]	Preproc [ms]	ParDec [ms]	Combine [μ s]
S.1	4/8		310	0.5	38
S.2	16/32		375	0.4	74
S.3	16/128	98	556	1.3	83
S.4	64/128		902	3.0	574
S.5	16/512		1156	4.0	176

Parameters: $n = 256$ accounts, $R = 10$ rounds, max. balance $B = 4$, max. amount $A = 4$.

Table 6: Dark-pool DEX (Algorithm 3).

Set	t/K	Eval [s]	Preproc [ms]	ParDec [ms]	Combine [μ s]
S.1	4/8		236	0.6	39
S.2	16/32		264	0.4	73
S.3	16/128	9.3	348	1.1	84
S.4	64/128		507	3.0	203
S.5	16/512		653	4.1	178

Parameters: $S = 4$ resting orders, $R = 10$ rounds, max. price $P = 4$, max. volume $V = 4$.

Table 7: Median price oracle (Algorithm 4).

Set	t/K	Eval [s]	Preproc [ms]	ParDec [ms]	Combine [μ s]
S.1	4/8		237	0.5	38
S.2	16/32		262	0.5	73
S.3	16/128	28	348	1.0	80
S.4	64/128		499	3.1	210
S.5	16/512		680	4.3	178

Parameters: $n = 128$ submissions, max. value $H = 8$.

moves an encrypted amount over $[0, A)$ between accounts. Throughout the protocol, multiple transfers can be processed sequentially (each with its own round). The contract only reveals the final balances of selected accounts. The rest of the state remains encrypted.

3. Dark-Pool DEX (Table 6). The contract holds an encrypted order book. It matches incoming requests against it, publishing only the executed fills while prices over $[0, P)$, volumes over $[0, V)$, and unmatched orders remain encrypted. After removing the matched orders, the contract updates the order book with new encrypted submissions. This process repeats for R rounds, and the unmatched orders are never revealed.

4. Median Price Oracle (Table 7). The contract aggregates n encrypted submissions over $[0, H)$ from many nodes into their median. Computing the median homomorphically requires obviously sorting the inputs, for which we employ Batcher’s odd–even mergesort.

Across all four programs, the threshold reveal (Preproc + ParDec + Combine) costs at most a few hundred milliseconds for the small committees and stays below 1.2 seconds even for $K = 512$. For the computa-

tionally heavy programs this overhead is negligible relative to the evaluation. Thus, for any non-trivial computation, `tulva-tfhe` adds little overhead over a non-threshold deployment.

AI Disclosure

The construction, all proof ideas and the parameter analysis are due to the authors. The first version of this work was written entirely by hand and was then proofread by Claude Opus 5 and 5.5 as well as by GPT-5.6 Sol. The models pointed out small gaps in some proofs and proposed corrections, which the authors verified before adopting, and the authors revised every passage edited by the models in subsequent editing rounds. The scripts that select the parameters of Table 1 were first written by hand, and Claude Opus 5 and 5.5 were then used to optimise their search logic and to refactor them. The authors checked that the resulting scripts are consistent with the analysis in the paper. The algorithms in the implementation were first written by hand, and Claude Opus 5 and 5.5 as well as GPT-6 Astra were used extensively to rewrite and optimise existing parts of the implementation.

Acknowledgements

Russell W. F. Lai and Michał Osadnik are supported by the Research Council of Finland grant 358951 and the Future Makers Program project SPICED of the Technology Industries of Finland Centennial Foundation. Ivy K. Y. Woo is supported by the Research Council of Finland grant 358950. Antonio Guimarães is partially supported by grants JDC2024-055789-I and CEX2024-001471-M funded by MICIU/AEI/10.13039/501100011033 and ESF+, and by the ERC Proof of Concept grant VERIFHE (GA: 101287502). Valerio Cini is supported by the European Union’s Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101202597 (ACryL). Ivy K. Y. Woo would like to thank Yongsoo Song for sharing the reduction in [HKM⁺26] during the Novel Lattice Assumptions Workshop in July 2026, which significantly improved the results of this work.

References

- ABMP26. Andreea Alexandru, Ahmad Al Badawi, Daniele Micciancio, and Yuriy Polyakov. Application-aware approximate homomorphic encryption: Configuring FHE for practical use. *CiC*, 2(4):3, 2026. [doi:10.62056/ay183z10k](#). 11
- AJL⁺12. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501. Springer, Berlin, Heidelberg, April 2012. [doi:10.1007/978-3-642-29011-4_29](#). 4
- AL21. Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 519–548, Virtual Event, August 2021. Springer, Cham. [doi:10.1007/978-3-030-84245-1_18](#). 9, 10, 28, 30
- AP13. Jacob Alperin-Sheriff and Chris Peikert. Practical bootstrapping in quasilinear time. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, August 2013. [doi:10.1007/978-3-642-40041-4_1](#). 17
- AP14. Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 297–314. Springer, Berlin, Heidelberg, August 2014. [doi:10.1007/978-3-662-44371-2_17](#). 17
- APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015. [doi:10.1515/jmc-2015-0016](#). 18
- Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993. [doi:10.1007/BF01445125](#). 30
- BBB⁺23. Loris Bergerat, Anas Boudi, Quentin Bourgerie, Ilaria Chillotti, Damien Ligier, Jean-Baptiste Orfila, and Samuel Tap. Parameter optimization and larger precision for (T)FHE. *Journal of Cryptology*, 36(3):28, July 2023. [doi:10.1007/s00145-023-09463-5](#). 18
- BBN⁺25. Dan Boneh, Benedikt Bünz, Kartik Nayak, Lior Rotem, and Victor Shoup. Context-dependent threshold decryption and its applications. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 506–538. Springer, Singapore, December 2025. [doi:10.1007/978-981-95-5119-4_16](#). 16

- BCD⁺25a. Flávio Bergamaschi, Anamaria Costache, Dana Dachman-Soled, Hunter Kippen, Lucas LaBuff, and Rui Tang. Revisiting the security of approximate FHE with noise-flooding countermeasures. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part V*, volume 15678 of *LNCS*, pages 102–134. Springer, Cham, May 2025. doi:10.1007/978-3-031-91832-2_4. 17
- BCD⁺25b. Carl Bootland, Kelong Cong, Daniel Demmler, Tore Kasper Frederiksen, Benoit Libert, Jean-Baptiste Orfila, Dragos Rotaru, Nigel P. Smart, Titouan Tanguy, Samuel Tap, and Michael Walter. Threshold (fully) homomorphic encryption. Cryptology ePrint Archive, Report 2025/699, 2025. URL: <https://eprint.iacr.org/2025/699>. 3
- BCF⁺25. Chris Brzuska, Sébastien Canard, Caroline Fontaine, Duong Hieu Phan, David Pointcheval, Marc Renard, and Renaud Sirdey. Relations among new CCA security notions for approximate FHE. *CiC*, 2(1):20, 2025. doi:10.62056/ae0iv7sf. 11
- BCK⁺22. Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 517–550. Springer, Cham, August 2022. doi:10.1007/978-3-031-15985-5_18. 4
- BCKS24. Youngjin Bae, Jung Hee Cheon, Jaehyung Kim, and Damien Stehlé. Bootstrapping bits with CKKS. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 94–123. Springer, Cham, May 2024. doi:10.1007/978-3-031-58723-8_4. 17
- BFM⁺25. Zvika Brakerski, Offir Friedman, Avichai Marmor, Dolev Mutzari, Yuval Spiizer, and Ni Trieu. Threshold FHE with efficient asynchronous decryption. Cryptology ePrint Archive, Report 2025/712, 2025. URL: <https://eprint.iacr.org/2025/712>. 3, 4, 5, 8, 9
- 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. doi:10.1007/978-3-319-96884-1_19. 3, 4, 8
- 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. doi:10.1145/2090236.2090262. 17
- BJSW25. Olivier Bernard, Marc Joye, Nigel P. Smart, and Michael Walter. Drifting towards better error probabilities in fully homomorphic encryption schemes. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part VIII*, volume 15608 of *LNCS*, pages 181–211. Springer, Cham, May 2025. doi:10.1007/978-3-031-91101-9_7. 11
- BKW26. Chris Brzuska, Michael Kloof, and Ivy K. Y. Woo. Threshold public-key encryption: Definitions, relations, and CPA-to-CCA transforms. In Shi Bai and Edoardo Persichetti, editors, *PKC 2026, Part IV*, volume 16554 of *LNCS*, pages 162–197. Springer, Cham, May 2026. doi:10.1007/978-3-032-26740-5_6. 3, 5, 8
- BLMR13. Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40041-4_23. 11, 18
- BP26. Luís T. A. N. Brandão and René Peralta. NIST first call for multi-party threshold schemes. NIST IR 8214C, 2026. URL: <https://csrc.nist.gov/pubs/ir/8214/c/final>. 3
- Bra12. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, Berlin, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_50. 6, 17
- BS23. Katharina Boudgoust and Peter Scholl. Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part I*, volume 14438 of *LNCS*, pages 371–404. Springer, Singapore, December 2023. doi:10.1007/978-981-99-8721-4_12. 3, 4, 8
- CCC⁺25. Ignacio Cascudo, Anamaria Costache, Daniele Cozzo, Dario Fiore, Antonio Guimarães, and Eduardo Soria-Vazquez. Verifiable computation for approximate homomorphic encryption schemes. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VII*, volume 16006 of *LNCS*, pages 643–677. Springer, Cham, August 2025. doi:10.1007/978-3-032-01907-3_21. 17
- CDKS21. Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. Efficient homomorphic conversion between (ring) LWE ciphertexts. In Kazue Sako and Nils Ole Tippenhauer, editors, *ACNS 2021, Part I*, volume 12726 of *LNCS*, pages 460–479. Springer, Cham, June 2021. doi:10.1007/978-3-030-78372-3_18. 6, 17
- CGGI16. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 3–33. Springer, Berlin, Heidelberg, December 2016. doi:10.1007/978-3-662-53887-6_1. 6, 17, 20

- CKKS17. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 409–437. Springer, Cham, December 2017. doi:10.1007/978-3-319-70694-8_15. 6, 17
- CLW25. Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. Pilvi: Lattice threshold PKE with small decryption shares and improved security. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 539–569. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5119-4_17. 4, 5, 6, 8, 9, 10, 34, 39, 43, 51, 59
- DDEK⁺23. Morten Dahl, Daniel Demmler, Sarah El Kazdadi, Arthur Meyre, Jean-Baptiste Orfila, Dragos Rotaru, Nigel P. Smart, Samuel Tap, and Michael Walter. Noah’s Ark: Efficient threshold-FHE using noise flooding. In *Proceedings of the 11th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pages 35–46, 2023. doi:10.1145/3605759.3625259. 3, 4, 5
- DLN⁺21. Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 659–690. Springer, Cham, May 2021. doi:10.1007/978-3-030-75245-3_24. 8
- DM15. Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 617–640. Springer, Berlin, Heidelberg, April 2015. doi:10.1007/978-3-662-46800-5_24. 17
- Fhe24. Fhenix. Fhenix: Private DeFi on Ethereum with fully homomorphic encryption. <https://www.fhenix.io/>, 2024. Accessed: 2026-06-09. 3
- FV12. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. URL: <https://eprint.iacr.org/2012/144>. 6, 17
- GHPS12. Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Ring switching in BGV-style homomorphic encryption. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12*, volume 7485 of *LNCS*, pages 19–37. Springer, Berlin, Heidelberg, September 2012. doi:10.1007/978-3-642-32928-9_2. 6, 17
- GHS12. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 850–867. Springer, Berlin, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_49. 17
- 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. doi:10.1007/978-3-030-45374-9_21. 47, 48
- GP25. Antonio Guimarães and Hilder V. L. Pereira. Fast amortized bootstrapping with small keys and polynomial noise overhead. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025*, pages 2967–2981. ACM Press, October 2025. doi:10.1145/3719027.3765181. 17
- HKM⁺26. Yisol Hwang, Shuichi Katsumata, Seonhong Min, Guilhem Niot, and Yongsoo Song. Proving threshold Regev PKE from adaptive hint-MLWE: Efficient, non-interactive, and CCA secure. Cryptology ePrint Archive, Paper 2026/1585, 2026. URL: <https://eprint.iacr.org/2026/1585>. 9, 22, 46, 48, 50
- Kim25. Jaehyung Kim. Bootstrapping GBFV with CKKS. Cryptology ePrint Archive, Report 2025/888, 2025. URL: <https://eprint.iacr.org/2025/888>. 17
- KLL⁺26. Won Kim, Changmin Lee, JeongHwan Lee, Alain Passelègue, and Damien Stehlé. Asynchronous Lagrange-based threshold FHE with smaller modulus overhead. In *CRYPTO 2026*, 2026. doi:10.1007/978-3-032-35374-0_1. 4, 5, 6, 8, 9, 30, 31, 32, 34
- KLNO24a. Michael Klooß, Russell W. F. Lai, Ngoc Khanh Nguyen, and Michal Osadnik. RoK, paper, SISsors – toolkit for lattice-based succinct arguments. Cryptology ePrint Archive, Report 2024/1972, 2024. URL: <https://eprint.iacr.org/2024/1972>. 28
- KLNO24b. Michael Klooß, Russell W. F. Lai, Ngoc Khanh Nguyen, and Michal Osadnik. RoK, paper, SISsors toolkit for lattice-based succinct arguments - (extended abstract). In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part V*, volume 15488 of *LNCS*, pages 203–235. Springer, Singapore, December 2024. doi:10.1007/978-981-96-0935-2_7. 9, 10
- 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. doi:10.1007/978-3-031-38554-4_18. 46
- LM21. Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 648–677. Springer, Cham, October 2021. doi:10.1007/978-3-030-77870-5_23. 16, 17

- 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. doi:[10.1145/3658644.3690330](https://doi.org/10.1145/3658644.3690330). 17
- LSW25. Russell W. F. Lai, Monisha Swarnakar, and Ivy K. Y. Woo. Leaky LWE: Learning with errors with semi-adaptive secret- and error-leakage. *CiC*, 2(3):21, 2025. doi:[10.62056/ah89ksuc2](https://doi.org/10.62056/ah89ksuc2). 9, 10, 29, 46, 47, 51, 55
- MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Berlin, Heidelberg, April 2012. doi:[10.1007/978-3-642-29011-4_41](https://doi.org/10.1007/978-3-642-29011-4_41). 6, 7, 11
- 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. doi:[10.1109/FOCS.2004.72](https://doi.org/10.1109/FOCS.2004.72). 10
- MS18. Daniele Micciancio and Jessica Sorrell. Ring packing and amortized FHEW bootstrapping. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPICs*, pages 100:1–100:14. Schloss Dagstuhl, July 2018. doi:[10.4230/LIPICs.ICALP.2018.100](https://doi.org/10.4230/LIPICs.ICALP.2018.100). 6, 17
- OT25. Hiroki Okada and Tsuyoshi Takagi. Low communication threshold FHE from standard (module-)LWE. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VII*, volume 16251 of *LNCS*, pages 165–198. Springer, Singapore, December 2025. doi:[10.1007/978-981-95-5122-4_6](https://doi.org/10.1007/978-981-95-5122-4_6). 3, 4, 8
- PS24. Alain Passelègue and Damien Stehlé. Low communication threshold fully homomorphic encryption. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 297–329. Springer, Singapore, December 2024. doi:[10.1007/978-981-96-0875-1_10](https://doi.org/10.1007/978-981-96-0875-1_10). 4
- vzGG13. Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013. doi:[10.1017/CB09781139856065](https://doi.org/10.1017/CB09781139856065). 16
- Zam23a. Zama. fhEVM: Confidential EVM smart contracts using fully homomorphic encryption. Whitepaper, <https://github.com/zama-ai/fhevm/blob/main/fhevm-whitepaper.pdf>, 2023. Accessed: 2026-06-09. 3
- Zam23b. Zama. TFHE-rs: A pure Rust implementation of the TFHE scheme for Boolean and integer arithmetics over encrypted data, version 1.4.1. <https://github.com/zama-ai/tfhe-rs>, 2023. Accessed: 2026-02-01. 4, 6
- ZZLP25. Guy Zyskind, Doron Zarchy, Max Leibovich, and Chris Peikert. High-throughput universally composable threshold FHE decryption. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025*, pages 2339–2353. ACM Press, October 2025. doi:[10.1145/3719027.3744884](https://doi.org/10.1145/3719027.3744884). 3, 4, 5, 19, 20

Table 8: Functional-bootstrap tables instantiated by the FHEVM programs.

Table	Definition	Description	Program(s)
<i>One-input table</i>			
$T_{=c}$	$\mathbf{1}\{x = c\}$	equality test of x against a public constant c	Auction, Dark-pool
$T_{\leq a,b}$	$(x \leq a) ? x : b$	keep x when $x \leq a$, otherwise lift it to b	Dark-pool
T_{refresh}	x	identity bootstrap (a “refresh”): clears accumulated noise	Dark-pool
<i>Two-input table</i>			
$T_{>}$	$\mathbf{1}\{x > y\}$	greater-than test of two bids	Auction
T_{sel}	$x ? y : 0$	multiplexer: pass value y when bit x is set, else 0	Auction, Dark-pool
T_{debit}	$(x \geq y) ? x - y : x$	new sender balance: x less amount y when affordable	ERC-20
T_{credit}	$(x \geq y) ? y : 0$	amount credited to the receiver, or 0 if balance $x < y$	ERC-20
$T_{=}$	$\mathbf{1}\{x = y\}$	equality test of two encrypted values	Dark-pool
T_{scan}	$\mathbf{1}\{x = 1 \wedge y = 0\}$	prefix-scan step: fires when equality bit $x = 1$ and carry $y = 0$	Dark-pool
$T_{\wedge}$	$x \wedge y$	logical AND (one-hot x , empty-slot bit y): the replace flag	Dark-pool
<i>Two-input sort</i>			
T_{sort}	$(\min(x, y), \max(x, y))$	sort the pair (x, y) into $(\min, \max)$	Dark-pool, Oracle

Algorithm 1: Sealed-bid first-price auction.

Input: distinct bids $b_0, \dots, b_{n-1} \in \{0, \dots, B-1\}$
Output: winner index w and clearing price b_w

```

1 for  $i \leftarrow 0$  to  $n-1$  do
2    $s_i \leftarrow 0$  // win count of  $i$ .
3   for  $j \leftarrow 0$  to  $n-1$  with  $j \neq i$  do
4     if  $b_i > b_j$  then  $s_i \leftarrow s_i + 1$ 
      //  $s_i \leftarrow s_i + T_{>}(b_i, b_j) : n(n-1)$  bts
5  $o_i \leftarrow \mathbf{1}\{s_i = n-1\}$  for all  $i$ , then  $w \leftarrow \sum_i i \cdot o_i$  //  $w \leftarrow \sum_i i \cdot T_{=c}(s_i) : n$  bts
6  $b_w \leftarrow \sum_i (o_i ? b_i : 0)$  //  $b_w \leftarrow \sum_i T_{\text{sel}}(o_i, b_i) : n$  bts
7 return  $(w, b_w)$ 
```

A Algorithms for FHEVM Programs

For each of the four programs in Section 6, we state the plaintext algorithm and its homomorphic evaluation at the level of ciphertexts. Throughout we write $\text{Enc}(x)$ for a ciphertext encrypting x , i.e. shorthand for $\text{thFHE}.\text{Enc}(\text{pk}, x)$. We use $\{0, 1, \dots, p-1\}$ to represent $\mathbb{Z}_p$. We write $\mathbf{1}\{c\}$ for the bit 1 when a condition c holds and 0 otherwise, and $(c ? a : b)$ for the value a if c holds and b otherwise.

Homomorphic addition and multiplication by a public scalar need no bootstrapping (bts) and are evaluated directly. Everything else is built from the tables in Table 8, the gadgets shared across the four programs. Each table T takes one or two encrypted inputs, returns one encrypted output (or two for T_{sort}), and is evaluated by a single functional bootstrap. Using these gadgets, we describe the homomorphic realisations of the four programs below, with pseudocode listed in Algorithms 1 to 4. In each program, the **highlighted comments** indicate the number of bootstraps required for each step.

1. Sealed-Bid Auction (Algorithm 1). A first-price auction reveals only the winner and the price they pay. Assume the bids are distinct; then one bidder i outbids all $n-1$ others. To find that bidder, form the win count $s_i = \sum_{j \neq i} T_{>}(b_i, b_j)$ for bids in $[0, B)$. Computing all n of them costs $n(n-1)$ bootstraps. Bidder i wins exactly when $s_i = n-1$, so $o_i = T_{=c}(s_i)$ at $c = n-1$ is a one-hot indicator of the winner. The two outputs are linear in o_i . The winner index is $w = \sum_i i o_i$; the clearing price is $b_w = \sum_i T_{\text{sel}}(o_i, b_i)$, where the multiplexer T_{sel} keeps b_i on the winning slot and zeroes the rest. Evaluation is $n^2 + n$ bootstraps (Table 4), and each of w, b_w is read by one **Preproc**. Correctness needs $B^2 \leq p$ for the two-input comparison $T_{>}$ on bids in $[0, B)$, and $n \leq p$ to keep the win counts s_i and the winner index w below p .

Algorithm 2: Confidential ERC-20 transfers.

Input: sender balances $s_0, \dots, s_{n-1}$, receiver balances $r_0, \dots, r_{n-1}$, per-round transfer amounts $a_i^{(\rho)}$
Output: updated balances $(s_i, r_i)_i$

```
1 for  $\rho \leftarrow 1$  to  $R$  do
2   for  $i \leftarrow 0$  to  $n-1$  do
3     if  $s_i \geq a_i^{(\rho)}$  then
4        $r_i \leftarrow r_i + a_i^{(\rho)}$  //  $r_i \leftarrow T_{\text{credit}}(s_i, a_i^{(\rho)}): Rn \text{ bts}$ 
5        $s_i \leftarrow s_i - a_i^{(\rho)}$  //  $s_i \leftarrow T_{\text{debit}}(s_i, a_i^{(\rho)}): Rn \text{ bts}$ 
6 return  $(s_i, r_i)_i$  // queried accounts only. 1 Preproc each
```

Algorithm 3: Dark-pool order matching (stateful, R rounds).

Input: book of S resting asks $(\pi_s, \nu_s)_s$ (price, volume); requests $(\ell^{(\rho)}, q^{(\rho)})$ and pending orders $(\hat{\pi}^{(\rho)}, \hat{\nu}^{(\rho)})$,
for $\rho = 1, \dots, R$
Output: per-round fills $f^{(1)}, \dots, f^{(R)}$

```
1 for  $\rho \leftarrow 1$  to  $R$  do
2   for  $s \leftarrow 0$  to  $S-1$  do  $\kappa_s \leftarrow (\pi_s \leq \ell^{(\rho)}) ? \pi_s : P$ 
3     //  $\kappa_s \leftarrow T_{\leq \ell^{(\rho)}, P}(\pi_s): RS \text{ bts}$ 
4    $\beta \leftarrow \min_s \kappa_s$ 
5   //  $S-1$  comparisons  $(\beta, \cdot) \leftarrow T_{\text{sort}}(\beta, \kappa_s)$ , then  $\beta \leftarrow T_{\text{refresh}}(\beta): RS \text{ bts}$ 
6   if  $\beta = P$  then
7      $f^{(\rho)} \leftarrow 0$  // nothing crosses (branchless)
8   else
9      $s^* \leftarrow \text{first } s \text{ with } \pi_s = \beta$ 
10    //  $\forall s \text{ (S slots): } e_s \leftarrow T_{=}(\pi_s, \beta), m_s \leftarrow T_{\text{scan}}(e_s, g): 2RS \text{ bts}$ 
11     $f^{(\rho)} \leftarrow \min(q^{(\rho)}, \nu_{s^*})$ 
12    //  $v \leftarrow T_{\text{refresh}}(\sum_s T_{\text{sel}}(m_s, \nu_s)): S+1 \text{ bts},$ 
13    //  $(f^{(\rho)}, \cdot) \leftarrow T_{\text{sort}}(q^{(\rho)}, v): +1 \text{ bt, total } R(S+2)$ 
14     $\nu_{s^*} \leftarrow \nu_{s^*} - f^{(\rho)}$  //  $\forall s: \nu_s \leftarrow T_{\text{sel}}(m_s, f^{(\rho)})$  (only  $s^*$  changes):  $RS \text{ bts}$ 
15    if  $\nu_{s^*} = 0$  then  $(\pi_{s^*}, \nu_{s^*}) \leftarrow (\hat{\pi}^{(\rho)}, \hat{\nu}^{(\rho)})$ 
16    //  $d \leftarrow T_{=0}(\nu_{s^*}), \forall s: h_s \leftarrow T_{\wedge}(m_s, d),$ 
17    //  $\pi_s \leftarrow T_{\text{sel}}(h_s, \hat{\pi}^{(\rho)}) + T_{\text{sel}}(1-h_s, \pi_s),$ 
18    //  $\nu_s \leftarrow T_{\text{sel}}(h_s, \hat{\nu}^{(\rho)}) + T_{\text{sel}}(1-h_s, \nu_s): (5S+1)R \text{ bts}$ 
19 return  $(f^{(\rho)})_\rho$  // book stays hidden. 1 Preproc per round
```

2. Confidential ERC-20 Transfer (Algorithm 2). Each account on a confidential ERC-20 ledger carries an encrypted balance, and a transfer moves tokens without exposing any balance or the amount sent. Account i holds a sender balance s_i and a receiver balance r_i as persistent state. A transfer debits the sender only when its funds suffice. In round ρ , two two-input bootstraps on $(s_i, a_i^{(\rho)})$ decide account i 's transfer obliviously: $T_{\text{debit}}(s_i, a_i^{(\rho)})$ returns the new sender balance $(s_i - a_i^{(\rho)})$ if $s_i \geq a_i^{(\rho)}$, else s_i and $T_{\text{credit}}(s_i, a_i^{(\rho)})$ the moved amount $(a_i^{(\rho)})$ if affordable, else 0. The receiver is credited by $r_i \leftarrow r_i + T_{\text{credit}}(s_i, a_i^{(\rho)})$, and $T_{\text{debit}}(s_i, a_i^{(\rho)})$ replaces s_i , for $2Rn$ bootstraps in total. A balance is read with one Preproc. The encoding is correct while $AB \leq p$ for the two-input tables $T_{\text{debit}}, T_{\text{credit}}$ on balances in $[0, B)$ and amounts in $[0, A)$, and every reachable balance stays below p as a receiver balance accumulates credits across rounds.

3. Dark-Pool DEX (Algorithm 3). The resting book of a dark pool consists of S orders (π_s, ν_s) , prices and volumes held encrypted state and never revealed; only the trades executed against it become public. Each round matches one buy request $(\ell^{(\rho)}, q^{(\rho)})$, a limit price and a quantity, against the cheapest ask it crosses, reports the fill $f^{(\rho)}$, and updates the book, refilling an emptied slot from a pending order $(\hat{\pi}^{(\rho)}, \hat{\nu}^{(\rho)})$. To keep the matched slot secret, every step runs over all S slots and carries the match in a

Algorithm 4: Median price oracle.

Input: price submissions $x_0, \dots, x_{n-1} \in \{0, \dots, H-1\}$
Output: robust aggregate price

```

1  $(y_0, \dots, y_{n-1}) \leftarrow \text{sort}(x_0, \dots, x_{n-1})$  // nondecreasing.
// Batcher network of  $T_{\text{sort}}$ :  $((\log_2 n)^2 - \log_2 n + 4) n/4 - 1$  bts
2 return  $y_{\lfloor n/2 \rfloor}$  // median only. 1 Preproc

```

one-hot indicator m_s . The round has three stages: *cross* prices the eligible asks and takes the lowest, β ; *locate* marks the first slot at β as m_s ; and *fill and update* computes the fill and writes the book back.

Cross. An ask crosses when $\pi_s \leq \ell^{(\rho)}$. The crossing key $\kappa_s = T_{\leq \ell^{(\rho)}, P}(\pi_s)$ keeps a crossing ask at its own price and lifts every other to a reserved value P above all prices (prices in $[0, P)$), so the lowest crossing price is $\beta = \min_s \kappa_s$. This minimum is a tournament of $S-1$ pairwise sorts T_{sort} , then one T_{refresh} .

Locate. The match is the first slot at price β , a one-hot m_s . A per-slot equality $e_s = T_{=}(\pi_s, \beta)$ flags every slot at β , and a left-to-right prefix scan keeps the first. The scan carries a flag g , initially 0, that flips once a match has been passed: T_{scan} sets $m_s = 1$ exactly when $e_s = 1$ and $g = 0$, then $g += m_s$. The scan is sequential, slot s reading slot $s-1$'s flag.

Fill and update. The matched volume $v = \sum_s T_{\text{sel}}(m_s, \nu_s)$ is cleaned by T_{refresh} , and the fill $f^{(\rho)} = \min(q^{(\rho)}, v)$ is one further T_{sort} . The matched ask is drawn down by $\nu_s -= T_{\text{sel}}(m_s, f^{(\rho)})$, and an emptied match is refilled: $d = T_{=0}(\nu_{s^*})$ tests emptiness, $h_s = T_{\wedge}(m_s, d)$ flags the slot, and every slot is rewritten as $\pi_s \leftarrow T_{\text{sel}}(h_s, \hat{\pi}^{(\rho)}) + T_{\text{sel}}(1 - h_s, \pi_s)$ (ν_s analogously). All S slots are rewritten, so only the matched one changes.

For volumes in $[0, V)$, the round is correct while $P \leq V$, $2V \leq p$ (the fill sorts the volumes $q^{(\rho)}$ and the matched volume, both $< p/2$), and $P < p/2$.

4. Median Price Oracle (Algorithm 4). A median price oracle aggregates n encrypted submissions into their median. The circuit sorts them obviously and returns the middle element. The sorting algorithm is Batcher's odd-even merge-sort. Each comparator is one two-input sort T_{sort} , hence one bootstrap. For n a power of two the network has $((\log_2 n)^2 - \log_2 n + 4) n/4 - 1$ comparators (1471 at $n = 128$). Reading the median $y_{\lfloor n/2 \rfloor}$ is one **Preproc**. Correctness needs every submission in $[0, H)$ with $H = p/2$, the domain of the two-input sort.

B Additional Preliminaries

We give some additional preliminaries beyond those in Section 3 that are used in the proofs of our main theorems.

We write $\otimes$ for the tensor (i.e. Kronecker) product of matrices. The matrix tensor product satisfies the mixed product property: For all matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ of suitable dimensions, we have $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$.

Supplementary lemmas. Below are lemmas on the ring expansion factors, subtractive sets, Gaussian norm bounds, and noise flooding.

Lemma 1. *If $\mathfrak{f}$ is a power of two, then $\mathfrak{c}_2 \leq \sqrt{\varphi}$ and $\mathfrak{c}_\infty \leq \varphi$. If $\mathfrak{f}$ is an odd prime, then $\mathfrak{c}_2 \leq \mathfrak{f}$ and $\mathfrak{c}_\infty \leq 2\varphi$.*

Proof. The claims about $\mathfrak{c}_\infty$ follow by realising that the proof of [AL21, Proposition 2] not only applies to the ring of integers $\mathcal{R}$ of $\mathcal{K}$ but also to $\mathcal{K}$ in general. For the claims about $\mathfrak{c}_2$, we use Items (i) and (ii) of [KLNO24a, Lemma 1], which state the following: For any $x \in \mathcal{K}$, (i) $\|\psi(x)\|_2 \leq \sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \|\sigma(x)\|_2$ and (ii) $\|\sigma(x)\|_2 \leq \sqrt{\hat{\mathfrak{f}}} \cdot \|\psi(x)\|_2$ where $\text{rad}(\mathfrak{f})$ is the radical (i.e. product of all distinct prime factors) of $\mathfrak{f}$ and $\hat{\mathfrak{f}} = \mathfrak{f}$ if $\mathfrak{f}$ is odd and $\hat{\mathfrak{f}} = \mathfrak{f}/2$ if $\mathfrak{f}$ is even. Consider $a, b \in \mathcal{K}$. Substituting $x = a \cdot b$, we have $\|\psi(a \cdot b)\|_2 \leq \sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \|\sigma(a \cdot b)\|_2 \leq \sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \|\sigma(a)\|_2 \cdot \|\sigma(b)\|_2 \leq \sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \hat{\mathfrak{f}} \cdot \|\psi(a)\|_2 \cdot \|\psi(b)\|_2$. Observe that $\sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \hat{\mathfrak{f}} = \mathfrak{f}$ if $\mathfrak{f}$ is an odd prime and $\sqrt{\frac{\text{rad}(\mathfrak{f})}{\mathfrak{f}}} \cdot \hat{\mathfrak{f}} = \sqrt{\hat{\mathfrak{f}}} = \sqrt{\varphi}$ if $\mathfrak{f}$ is a power of 2. $\square$

Lemma 2. Let f be a power of two. Apply each component σ_i , $i \in \mathbb{Z}_f^\times$, of the canonical embedding σ of $\mathcal{K}$ entrywise to vectors and matrices over $\mathcal{K}$. Then for any matrix $\mathbf{M} \in \mathcal{K}^{n \times d}$,

$$\|\mathbf{M}\|_{2,2} = \max_{i \in \mathbb{Z}_f^\times} \|\sigma_i(\mathbf{M})\|_{2,2}.$$

Proof. Throughout, let i and j range over $\mathbb{Z}_f^\times$, let σ_i be the embedding with $\sigma_i(\zeta) = \zeta^i$, and $\mathbf{M} \in \mathcal{K}^{n \times d}$. We assume $f \geq 2$, since $f = 1$ and $f = 2$ give the same field $\mathcal{K} = \mathbb{Q}$. Then $\mathbb{Z}_f^\times$ is the set of odd residues modulo f , and $\varphi = f/2$. We will show that (1) $\Psi(\mathbf{M})$ is unitarily equivalent to the block-diagonal matrix

$$\mathbf{D} := \begin{bmatrix} \sigma_1(\mathbf{M}) & & \\ & \ddots & \\ & & \sigma_{f-1}(\mathbf{M}) \end{bmatrix} \in \mathbb{C}^{\varphi n \times \varphi d},$$

(2) the former implies $\|\mathbf{M}\|_{2,2} = \|\mathbf{D}\|_{2,2}$, and (3) $\|\mathbf{D}\|_{2,2} = \max_i \|\sigma_i(\mathbf{M})\|_{2,2}$.

(1): First, let $\mathbf{U} \in \mathbb{C}^{\varphi \times \varphi}$ be the matrix such that its (i, k) -th entry is $\zeta^{ik}/\sqrt{\varphi}$, for all rows i and columns $k \in \{0, \dots, \varphi - 1\}$, so that

$$\sigma(x) = \sqrt{\varphi} \cdot \mathbf{U} \cdot \psi(x) \quad \forall x \in \mathcal{K}.$$

That is, $\mathbf{U}$ is the conversion matrix between the coefficient and canonical embeddings of $\mathbf{x} \in \mathcal{K}$ up to scaling. The inner product of rows i and j of $\mathbf{U}$ is $\frac{1}{\varphi} \sum_{k=0}^{\varphi-1} \omega^k$ where $\omega := \zeta^{i-j}$. If $i = j$ this equals 1. If $i \neq j$, then $i - j$ is even and nonzero modulo f , so $\omega \neq 1$ and $\omega^\varphi = \zeta^{f(i-j)/2} = 1$, hence the sum is $(\omega^\varphi - 1)/(\omega - 1) = 0$. Therefore $\mathbf{U}$ is unitary.

Next, for a vector $\mathbf{x} \in \mathcal{K}^d$, let

$$\hat{\sigma}(\mathbf{x}) := (\sigma_i(\mathbf{x}))_i \in \mathbb{C}^{\varphi d}$$

list the embeddings of $\mathbf{x}$ grouped by embedding rather than by entry. Also, let $\mathbf{P}_d$ be the permutation matrix reordering coordinates from grouping by entry to grouping by embedding, so that we have

$$\hat{\sigma}(\mathbf{x}) = \sqrt{\varphi} \cdot \mathbf{W}_d \cdot \psi(\mathbf{x}) \quad \text{where} \quad \mathbf{W}_d := \mathbf{P}_d \cdot (\mathbf{I}_d \otimes \mathbf{U}).$$

Observe that the matrix $\mathbf{W}_d$ is unitary, as it is a product of unitary matrices. Since each σ_i is a ring homomorphism, $\sigma_i(\mathbf{M}\mathbf{x}) = \sigma_i(\mathbf{M}) \cdot \sigma_i(\mathbf{x})$ for every i , implying $\hat{\sigma}(\mathbf{M}\mathbf{x}) = \mathbf{D} \cdot \hat{\sigma}(\mathbf{x})$. Together with $\psi(\mathbf{M}\mathbf{x}) = \Psi(\mathbf{M}) \cdot \psi(\mathbf{x})$ (Section 3), this gives

$$\mathbf{W}_n \cdot \Psi(\mathbf{M}) \cdot \psi(\mathbf{x}) = \mathbf{D} \cdot \mathbf{W}_d \cdot \psi(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{K}^d.$$

As $\psi(\mathcal{K}^d) = \mathbb{Q}^{\varphi d}$ contains a basis of $\mathbb{C}^{\varphi d}$, it follows that the block diagonalisation of $\Psi(\mathbf{M})$ is

$$\Psi(\mathbf{M}) = \mathbf{W}_n^* \cdot \mathbf{D} \cdot \mathbf{W}_d,$$

that is, $\Psi(\mathbf{M})$ is unitarily equivalent to $\mathbf{D}$.

(2): Since $\Psi(\mathbf{M})$ is real, its conjugate transpose is $\Psi(\mathbf{M})^T$, and since $\mathbf{W}_n$ is unitary, (1) gives

$$\Psi(\mathbf{M})^T \cdot \Psi(\mathbf{M}) = \mathbf{W}_d^* \cdot \mathbf{D}^* \cdot \mathbf{W}_n \cdot \mathbf{W}_n^* \cdot \mathbf{D} \cdot \mathbf{W}_d = \mathbf{W}_d^* \cdot (\mathbf{D}^* \mathbf{D}) \cdot \mathbf{W}_d.$$

Hence $\Psi(\mathbf{M})^T \Psi(\mathbf{M})$ and $\mathbf{D}^* \mathbf{D}$ are unitarily similar and have the same eigenvalues, so $s_{\max}(\Psi(\mathbf{M})) = s_{\max}(\mathbf{D})$. Since $\|\mathbf{M}\|_{2,2} = s_{\max}(\Psi(\mathbf{M}))$ by [LSW25, Lemma 2] (Section 3), we get $\|\mathbf{M}\|_{2,2} = \|\mathbf{D}\|_{2,2}$.

(3): The matrix $\mathbf{D}^* \mathbf{D}$ is block diagonal with diagonal blocks $\sigma_i(\mathbf{M})^* \sigma_i(\mathbf{M}) \in \mathbb{C}^{d \times d}$. The eigenvalues of a block-diagonal matrix are those of its diagonal blocks, so the largest eigenvalue of $\mathbf{D}^* \mathbf{D}$ is the maximum over i of the largest eigenvalue of $\sigma_i(\mathbf{M})^* \sigma_i(\mathbf{M})$. Taking square roots gives

$$\|\mathbf{D}\|_{2,2} = s_{\max}(\mathbf{D}) = \max_i s_{\max}(\sigma_i(\mathbf{M})) = \max_i \|\sigma_i(\mathbf{M})\|_{2,2}.$$

Finally, combining (2) and (3), we get $\|\mathbf{M}\|_{2,2} = \|\mathbf{D}\|_{2,2} = \max_i \|\sigma_i(\mathbf{M})\|_{2,2}$. $\square$

Lemma 3 (Adapted from [AL21, Theorems 1 and 2]). If $\mathfrak{f}$ is a power of two, then $\{1, \zeta, \dots, \zeta^{\mathfrak{f}-1}\}$ is ξ -subtractive for $\xi(t) = 2^{\lceil \log t \rceil}$. If $\mathfrak{f} = p$ is a prime, then the set $\left\{ \sum_{j=0}^{i-1} \zeta^j : i \in [p-1] \right\}$ is ξ -subtractive for $\xi(t) = 1$.

The following lemma, adapted from [KLL⁺26], provides bounds on the quality of the subtractive set we adopt in our implementation. Its proof is given in Appendix C.

Lemma 4 (Quality of Subtractive Sets. Adapted from [KLL⁺26, Corollary 5.1]). Let $\mathfrak{f}$ and $K \leq \mathfrak{f}$ be powers of two, $\eta := \zeta^{\mathfrak{f}/K}$ a primitive K -th root of unity, and let $\Xi' := \{\eta^i : 0 \leq i < K\}$. For any $2 \leq t \leq K$ and $\xi = \xi(t) = 2^{\lceil \log t \rceil}$, the set Ξ' and any subset thereof have the quality

$$\begin{aligned} \rho(t) &\leq \xi \cdot 2^{-(t-1)} \prod_{k=1}^{\lfloor t/2 \rfloor} \frac{1}{\sin^2(\pi k/K)}, \\ \theta(t) &\leq \xi \cdot A, \\ \gamma(t) &\leq \xi \cdot \sqrt{(K-t+1)(A^2 + (t-1)B^2) + 1}, \end{aligned}$$

where

$$A := 2^{t-1} \prod_{k=1}^{\lfloor (t-2)/2 \rfloor} \cos^2\left(\frac{\pi k}{K}\right), \quad B := \prod_{k=1}^{\lfloor (t-3)/2 \rfloor} \cos^2\left(\frac{\pi k}{K}\right) \prod_{k=1}^{\lfloor (t-1)/2 \rfloor} \frac{1}{\sin^2(\pi k/K)}.$$

Lemma 5 (Adapted from [Ban93, Lemma 1.5]). Let $\mathcal{R} = \mathbb{Z}[\zeta]$ with ring degree φ . For any $m \in \mathbb{N}$, $\chi > 0$, $\delta > 0$,

$$\Pr[\|\mathbf{e}\|_2 > \chi\sqrt{\varphi m} \mid \mathbf{e} \leftarrow \$ D_{\mathcal{R},\chi}^m] \leq 2^{-\varphi m}, \quad (7)$$

$$\Pr[\|\mathbf{e}\|_\infty > \chi\sqrt{\delta} \mid \mathbf{e} \leftarrow \$ D_{\mathcal{R},\chi}^m] \leq \varphi m \cdot 2^{-\delta}. \quad (8)$$

The following lemma is standard and can be shown easily by bounding the KL divergence and then via Pinsker's inequality.

Lemma 6 (Noise Flooding). For any $n \in \mathbb{N}$, $\chi > 0$, $\mathbf{c} \in \mathcal{R}^n$, it holds that $\text{KL}(D_{\mathcal{R},\chi}^n, D_{\mathcal{R},\chi}^n + \mathbf{c}) \leq \pi \cdot \frac{\|\mathbf{c}\|_2^2}{\chi^2}$ and $\text{SD}(D_{\mathcal{R},\chi}^n, D_{\mathcal{R},\chi}^n + \mathbf{c}) \leq \sqrt{\frac{\pi}{2}} \cdot \frac{\|\mathbf{c}\|_2}{\chi}$, where KL and SD denote the KL divergence and the statistical distance, respectively. In particular, if $\chi \geq \lambda^{\omega(1)} \cdot \|\mathbf{c}\|_2$, one has $D_{\mathcal{R},\chi}^n \approx_s D_{\mathcal{R},\chi}^n + \mathbf{c}$.

For our security proofs, we will need to apply noise flooding over Q many adaptive oracle queries. Naively making Q hybrid hops will lead to a factor Q blow-up in the advantage bound. The following lemma provides a tighter bound with only $\sqrt{Q}$ blow-up, it follows from bounding the KL divergence over all Q queries carefully and then applying Pinsker's inequality.

Lemma 7. Let $n \in \mathbb{N}$ and $\chi, B > 0$. For $b \in \{0, 1\}$ and any (computationally unbounded) adversary $\mathcal{A}$, let $\text{Flood}_{\mathcal{A}}^b$ be the game defined below:

$\text{Flood}_{\mathcal{A}}^b$	$\mathcal{O}_b(\mathbf{c} \in \mathcal{R}^n)$
$\text{return } \mathcal{A}^{\mathcal{O}_b}()$	$\text{assert } \ \mathbf{c}\ _2 \leq B$ $\mathbf{e} \leftarrow \$ D_{\mathcal{R},\chi}^n$ $\text{return } \mathbf{a} := b \cdot \mathbf{c} + \mathbf{e}$

For any (computationally unbounded) $\mathcal{A}$ that makes at most $Q \in \mathbb{N}$ queries to $\mathcal{O}_b$, it holds that

$$|\Pr[\text{Flood}_{\mathcal{A}}^0 = 1] - \Pr[\text{Flood}_{\mathcal{A}}^1 = 1]| \leq \sqrt{\frac{\pi}{2}} \cdot \frac{\sqrt{Q} \cdot B}{\chi}.$$

Proof. Fix any $\mathcal{A}$. As $\mathcal{A}$ can be unbounded, we may assume it is deterministic – fixing its coins to the ones such that the LHS of the concerned inequality is maximised. The view of $\mathcal{A}$ is the tuple of all oracle answers $(\mathbf{a}_1, \dots, \mathbf{a}_Q)$. Below, denote by P and $\hat{P}$ the joint distributions of these answers in $\text{Flood}_{\mathcal{A}}^1$ and $\text{Flood}_{\mathcal{A}}^0$, respectively. The LHS quantity is thus at most $\text{SD}(P, \hat{P})$, which we bound in the rest. We will use the chain-rule for KL divergence, for which we provide a proof below for completeness.

Proposition 1. For any two distributions $P, \hat{P}$ over tuples $a = (a_1, \dots, a_Q)$, the KL divergence between P and $\hat{P}$ satisfies

$$\text{KL}(P \parallel \hat{P}) = \sum_{i \in [Q]} \sum_{a_{<i}} P(a_{<i}) \cdot \text{KL}(P(a_i \mid a_{<i}) \parallel \hat{P}(a_i \mid a_{<i})). \quad (9)$$

Proof. The probability chain rule gives $P(a) = \prod_{i \in [Q]} P(a_i \mid a_{<i})$ and $\hat{P}(a) = \prod_{i \in [Q]} \hat{P}(a_i \mid a_{<i})$, where $a_{<i} = (a_1, \dots, a_{i-1})$. Hence

$$\begin{aligned} \text{KL}(P \parallel \hat{P}) &= \sum_a P(a) \log \frac{P(a)}{\hat{P}(a)} \\ &= \sum_a P(a) \sum_{i \in [Q]} \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})} \\ &= \sum_{i \in [Q]} \sum_a P(a) \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})} \\ &= \sum_{i \in [Q]} \sum_{a_{\leq i}} \left(\sum_{a_{>i}} P(a_{\leq i}, a_{>i}) \right) \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})} \\ &= \sum_{i \in [Q]} \sum_{a_{\leq i}} P(a_{\leq i}) \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})} \\ &= \sum_{i \in [Q]} \sum_{a_{<i}} P(a_{<i}) \sum_{a_i} P(a_i \mid a_{<i}) \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})} \\ &= \sum_{i \in [Q]} \sum_{a_{<i}} P(a_{<i}) \cdot \text{KL}(P(a_i \mid a_{<i}) \parallel \hat{P}(a_i \mid a_{<i})). \end{aligned}$$

In the above, the fourth line uses $a = (a_{\leq i}, a_{>i})$ where we denote $a_{\leq i} = (a_1, \dots, a_i)$ and $a_{>i} = (a_{i+1}, \dots, a_Q)$. The fifth line holds by definition of marginal distribution, i.e. $\sum_{a_{>i}} P(a_{\leq i}, a_{>i}) = P(a_{\leq i})$. The sixth line uses $P(a_{\leq i}) = P(a_{<i}) P(a_i \mid a_{<i})$, and the last line is by realising that the inner sum over a_i is the conditional KL divergence $\text{KL}(P(a_i \mid a_{<i}) \parallel \hat{P}(a_i \mid a_{<i})) = \sum_{a_i} P(a_i \mid a_{<i}) \log \frac{P(a_i \mid a_{<i})}{\hat{P}(a_i \mid a_{<i})}$. $\square$

With $\mathcal{A}$ deterministic, once the prefix $\mathbf{a}_{<i}$ is fixed, the query $\mathbf{c}_i$ is a fixed vector and the only fresh randomness in $\mathbf{a}_i$ is $\mathbf{e}_i \leftarrow D_{\mathcal{R}, \chi}^n$, drawn independently of the prefix. The conditional distributions are therefore $P(\mathbf{a}_i \mid \mathbf{a}_{<i}) = \mathbf{c}_i + D_{\mathcal{R}, \chi}^n$ and $\hat{P}(\mathbf{a}_i \mid \mathbf{a}_{<i}) = D_{\mathcal{R}, \chi}^n$.

By the KL divergence bound in Lemma 6 and the norm-check in the oracle $\mathbf{O}_b$, for every prefix $\mathbf{a}_{<i}$,

$$\text{KL}(P(\mathbf{a}_i \mid \mathbf{a}_{<i}) \parallel \hat{P}(\mathbf{a}_i \mid \mathbf{a}_{<i})) \leq \frac{\pi \|\mathbf{c}_i\|_2^2}{\chi^2} \leq \frac{\pi B^2}{\chi^2}.$$

Note that the above upper bound is a fixed value independent of the distribution of $\|\mathbf{c}_i\|$ (the latter may adaptively depend on the prefix $\mathbf{a}_{<i}$). Plugging into Eq. (9), we get

$$\text{KL}(P \parallel \hat{P}) \leq \sum_{i \in [Q]} \sum_{\mathbf{a}_{<i}} P(\mathbf{a}_{<i}) \cdot \frac{\pi B^2}{\chi^2} = \frac{\pi Q B^2}{\chi^2},$$

where the last equality uses $\sum_{\mathbf{a}_{<i}} P(\mathbf{a}_{<i}) = 1$. Finally, using Pinsker's inequality gives

$$\text{SD}(P, \hat{P}) \leq \sqrt{\frac{1}{2} \text{KL}(P \parallel \hat{P})} \leq \sqrt{\frac{\pi}{2}} \cdot \frac{\sqrt{Q} \cdot B}{\chi}. \quad \square$$

C Proof of Lemma 4 (Quality of Subtractive Sets)

To prove Lemma 4, we will use the following Lemma 8 which provides more general bounds for a subtractive set that consists of roots of unity. It is adapted from [KLL⁺26] to suit the norm convention in this work.

Lemma 8 (Adapted from [KLL⁺26, Theorem 4.3]). Let $\mathfrak{f}$ be a power of two, so that $\varphi = \mathfrak{f}/2$, let $\Xi = \{1, \zeta, \dots, \zeta^{\mathfrak{f}-1}\}$, and let $\xi = \xi(t) = 2^{\lceil \log t \rceil}$. For any $2 \leq t \leq \mathfrak{f}$, the set Ξ and any subset thereof have the quality

$$\begin{aligned}\rho(t) &\leq \xi \cdot 2^{-(t-1)} \prod_{k=1}^{\lfloor t/2 \rfloor} \frac{1}{\sin^2(\pi k/\mathfrak{f})}, \\ \theta(t) &\leq \xi \cdot A, \\ \gamma(t) &\leq \xi \cdot \sqrt{(\mathfrak{f}-t+1)(A^2 + (t-1)B^2) + 1},\end{aligned}$$

where

$$A := 2^{t-1} \prod_{k=1}^{\lfloor (t-2)/2 \rfloor} \cos^2\left(\frac{\pi k}{\mathfrak{f}}\right), \quad B := \prod_{k=1}^{\lfloor (t-3)/2 \rfloor} \cos^2\left(\frac{\pi k}{\mathfrak{f}}\right) \prod_{k=1}^{\lfloor (t-1)/2 \rfloor} \frac{1}{\sin^2(\pi k/\mathfrak{f})}.$$

Proof. Throughout this proof, for a set $S \subseteq \{0\} \cup \Xi$ and an element $\beta \notin S$, we write $\lambda_\mu^{S \rightarrow \beta} := \prod_{\nu \in S \setminus \{\mu\}} \frac{\beta - \nu}{\mu - \nu}$ for the Lagrange coefficient of the element $\mu \in S$.¹⁶ With this notation, the row vector $\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1}$ has its μ -th entry being $\xi \cdot \lambda_\mu^{T \rightarrow 0}$, and the matrix $\xi \cdot \mathbf{V}_{\Xi \setminus T^*} \cdot \mathbf{V}_{\{0\} \cup T^*}^{-1}$ has its (α, μ) -th entry being $\xi \cdot \lambda_\mu^{\{0\} \cup T^* \rightarrow \alpha}$, where the row index α ranges over $\Xi \setminus T^*$ and the column index μ over $\{0\} \cup T^*$. By Lemma 3, all these elements lie in $\mathcal{R}$.

Since $\mathfrak{f}$ is a power of two,

$$\|x\| \leq \|\sigma(x)\|_\infty = \max_\ell |\sigma_\ell(x)| \quad (10)$$

for any $x \in \mathcal{K}$, so it suffices to bound the canonical embedding. Fix one embedding σ_ℓ , sending ζ to a primitive $\mathfrak{f}$ -th root of unity $\omega = \exp(\pi i r/\varphi)$ with r coprime to $\mathfrak{f}$. For exponents a, b we have $|\sigma_\ell(\zeta^a)| = 1$, $|\sigma_\ell(\zeta^a - 0)| = 1$, and

$$|\sigma_\ell(\zeta^a - \zeta^b)| = |\omega^{a-b} - 1| = |\exp(2\pi i r(a-b)/\mathfrak{f}) - 1| = 2 |\sin(\pi r(a-b)/\mathfrak{f})|.$$

As r is coprime to $\mathfrak{f}$, multiplication by r permutes the integers modulo $\mathfrak{f}$, so we may assume $r = 1$ without loss of generality.

Proposition 2 (Sine-product). For any m distinct non-zero integers $d_1, \dots, d_m$ modulo $\mathfrak{f}$,

$$\prod_{j=1}^m \frac{1}{|\sin(\pi d_j/\mathfrak{f})|} \leq \prod_{k=1}^{\lceil m/2 \rceil} \frac{1}{\sin^2(\pi k/\mathfrak{f})}. \quad (11)$$

Proof. Indeed, $|\sin(\pi d/\mathfrak{f})|$ depends only on $\bar{d} := \min(d, \mathfrak{f}-d) \in \{1, \dots, \varphi\}$, since $\sin(\pi(\mathfrak{f}-d)/\mathfrak{f}) = \sin(\pi d/\mathfrak{f})$, and is strictly increasing in $\bar{d}$. Each value of $\bar{d}$ is attained by at most two integers modulo $\mathfrak{f}$, namely d and $\mathfrak{f}-d$. Hence the LHS in the proposition is the largest when the multiset $\{\bar{d}_j\}$ takes the smallest possible values each repeated at most twice, that is, $\{1, 1, 2, 2, \dots\}$ of size m , whose largest element is $\lceil m/2 \rceil$. When m is odd we enlarge the product by one factor $\sin^{-2} \geq 1$, which gives Eq. (11). $\square$

Proposition 3 (Cosine-product). For any m distinct non-zero integers $d_1, \dots, d_m$ modulo $\mathfrak{f}$,

$$\prod_{j=1}^m |\sin(\pi d_j/\mathfrak{f})| \leq \prod_{k=1}^{\lfloor (m-1)/2 \rfloor} \cos^2(\pi k/\mathfrak{f}). \quad (12)$$

Proof. Writing $e := \varphi - \bar{d} \in \{0, \dots, \varphi-1\}$, we have $|\sin(\pi d/\mathfrak{f})| = \sin(\pi(\varphi - e)/\mathfrak{f}) = \cos(\pi e/\mathfrak{f})$, which is the largest when e is the smallest. Each value $e = k \geq 1$ is attained by at most two integers modulo $\mathfrak{f}$ (those with $\bar{d} = \varphi - k$), and $e = 0$ by one (the integer $d = \varphi$), so the LHS in the proposition is the largest for the multiset $\{0, 1, 1, 2, 2, \dots\}$ of size m . For even m the largest element $m/2$ is unpaired, and bounding its single factor by $\cos(\pi(m/2)/\mathfrak{f}) \leq 1$ gives Eq. (12). $\square$

¹⁶ This is consistent with the notations in [KLL⁺26].

Bound on ρ . By definition $\rho(t) = \max_{T \subseteq t, \Xi} \|\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1}\| = \max_{T \subseteq t, \Xi} \|(\xi \cdot \lambda_\mu^{T \rightarrow 0})_{\mu \in T}\|$, that is, the maximum of $\|\xi \cdot \lambda_\mu^{T \rightarrow 0}\|$ over all $\mu \in T$ and all sets $T \subseteq t, \Xi$. By Eq. (10), to bound $\rho(t)$ it suffices to bound $|\sigma_\ell(\xi \cdot \lambda_\mu^{T \rightarrow 0})|$ for any μ, T and ℓ .

Fix a size- t set $T \subseteq t, \Xi$ and $\mu \in T$, and write each $\nu \in T$ as ζ^{c_ν} . Then $|\sigma_\ell(\nu)| = 1$ and $|\sigma_\ell(\mu - \nu)| = 2|\sin(\pi(c_\mu - c_\nu)/f)|$, so

$$|\sigma_\ell(\xi \cdot \lambda_\mu^{T \rightarrow 0})| = \xi \prod_{\nu \in T \setminus \{\mu\}} \frac{|\sigma_\ell(\nu)|}{|\sigma_\ell(\mu - \nu)|} = \xi \cdot 2^{-(t-1)} \prod_{\nu \in T \setminus \{\mu\}} \frac{1}{|\sin(\pi(c_\mu - c_\nu)/f)|}.$$

The $t-1$ integers $c_\mu - c_\nu$ are distinct and non-zero modulo f , so Eq. (11) with $m = t-1$ and $\lceil (t-1)/2 \rceil = \lfloor t/2 \rfloor$ upper-bounds this by $\xi \cdot 2^{-(t-1)} \prod_{k=1}^{\lfloor t/2 \rfloor} \sin^{-2}(\pi k/f)$, which is the stated bound.

Bound on γ . By definition $\gamma(t) = \max_{T^* \subseteq t-1, \Xi} \left\| \xi \cdot \mathbf{V}_\Xi \cdot \mathbf{V}_{\{0\} \cup T^*}^{-1} \right\|_{2,2}$. Fix any $T^* \subseteq t-1, \Xi$, define the shorthand $S := \{0\} \cup T^*$, and denote $\mathbf{M} := \xi \cdot \mathbf{V}_{\Xi \setminus T^*} \cdot \mathbf{V}_S^{-1}$. We bound separately the rows of $\xi \cdot \mathbf{V}_\Xi \cdot \mathbf{V}_S^{-1}$ indexed by $\Xi \setminus T^*$, which form $\mathbf{M}$, and those indexed by T^* , and then combine the two bounds.

For any row $\alpha \in \Xi \setminus T^*$ and column $\mu \in S$, the (α, μ) -th entry of $\mathbf{M}$ is

$$\mathbf{M}_{\alpha, \mu} = \xi \cdot \lambda_\mu^{S \rightarrow \alpha} = \xi \prod_{\nu \in S \setminus \{\mu\}} \frac{\alpha - \nu}{\mu - \nu}. \quad (13)$$

Write the evaluation point as $\alpha = \zeta^a$ and each node $\nu \in T^*$ as $\nu = \zeta^{c_\nu}$. Relative to the bound on $\rho(t)$, the main difference here is that the numerator is $\alpha - \nu$ (instead of $-\nu$) which is no longer a unit of $\mathcal{R}$, and will yield a sine factor $|\sigma_\ell(\alpha - \nu)| = 2|\sin(\pi(a - c_\nu)/f)|$ as shown below.

For a canonical embedding σ_ℓ , extend σ_ℓ entrywise to matrices and vectors of ring elements, so that $\sigma_\ell(\mathbf{M}) = (\sigma_\ell(\mathbf{M}_{\alpha, \mu}))_{\alpha, \mu} \in \mathbb{C}^{(f-t+1) \times t}$. Since f is a power of two, Lemma 2 applies to $\mathbf{M}$, and

$$\|\mathbf{M}\|_{2,2} = \max_\ell \|\sigma_\ell(\mathbf{M})\|_{2,2} \leq \max_\ell \|\sigma_\ell(\mathbf{M})\|_F = \max_\ell \sqrt{\sum_{\alpha, \mu} |\sigma_\ell(\mathbf{M}_{\alpha, \mu})|^2}, \quad (14)$$

where the inequality upper-bounds $\|\cdot\|_{2,2}$ by the Frobenius norm $\|\cdot\|_F$. In the following we bound $|\sigma_\ell(\mathbf{M}_{\alpha, \mu})|$ for each index α, μ . We recall that $\mathbf{M}_{\alpha, \mu}$ can be expressed as in Eq. (13).

Fix any row $\alpha \in \Xi \setminus T^*$. We consider two cases: the column index of $\mathbf{M}_{\alpha, \mu}$ is $\mu = 0$ or $\mu \in T^*$.

For $\mu = 0$, the denominator $\prod_{\nu \in T^*} (0 - \nu)$ in Eq. (13) is a product of units, so

$$|\sigma_\ell(\xi \cdot \lambda_0^{S \rightarrow \alpha})| = \xi \prod_{\nu \in T^*} |\sigma_\ell(\alpha - \nu)| = \xi \cdot 2^{t-1} \prod_{\nu \in T^*} |\sin(\pi(a - c_\nu)/f)|.$$

The $t-1$ integers $a - c_\nu$ are distinct and non-zero modulo f , so Eq. (12) with $m = t-1$ upper-bounds this by

$$|\sigma_\ell(\xi \cdot \lambda_0^{S \rightarrow \alpha})| \leq \xi \cdot 2^{t-1} \prod_{k=1}^{\lfloor (t-2)/2 \rfloor} \cos^2(\pi k/f) = \xi \cdot A.$$

For $\mu \in T^*$, writing $\mu = \zeta^c$ for some c and factoring out the term with $\nu = 0$,

$$|\sigma_\ell(\xi \cdot \lambda_\mu^{S \rightarrow \alpha})| = \xi \cdot \frac{|\sigma_\ell(\alpha - 0)|}{|\sigma_\ell(\mu - 0)|} \prod_{\nu \in T^* \setminus \{\mu\}} \frac{|\sigma_\ell(\alpha - \nu)|}{|\sigma_\ell(\mu - \nu)|} = \xi \prod_{\nu \in T^* \setminus \{\mu\}} \frac{|\sin(\pi(a - c_\nu)/f)|}{|\sin(\pi(c - c_\nu)/f)|},$$

where the last line is because both α and μ are units so $|\sigma_\ell(\alpha)| = |\sigma_\ell(\mu)| = 1$. Within each of the numerator and the denominator in Eq. (13), the $t-2$ integers $a - c_\nu$ (respectively $c - c_\nu$) are distinct and non-zero modulo f . Hence, we can bound the numerator by Eq. (12) with $m = t-2$, and the inverse of the denominator by Eq. (11) with $m = t-2$ (using $\lceil (t-2)/2 \rceil = \lfloor (t-1)/2 \rfloor$), obtaining

$$|\sigma_\ell(\xi \cdot \lambda_\mu^{S \rightarrow \alpha})| \leq \xi \prod_{k=1}^{\lfloor (t-3)/2 \rfloor} \cos^2(\pi k/f) \prod_{k=1}^{\lfloor (t-1)/2 \rfloor} \frac{1}{\sin^2(\pi k/f)} = \xi \cdot B.$$

For any $\alpha \in \Xi \setminus T^*$, each row of $\sigma_\ell(\mathbf{M})$ indexed by α has 1 entry with $\mu = 0$ and $t - 1$ entries with $\mu \in T^*$. Using the bounds from the above two cases,

$$\sum_{\mu} |\sigma_\ell(\mathbf{M}_{\alpha, \mu})|^2 \leq (\xi \cdot A)^2 + (t - 1)(\xi \cdot B)^2 = \xi^2 \cdot (A^2 + (t - 1)B^2).$$

Summing over the $\mathfrak{f} - t + 1$ rows, we get that

$$\sum_{\alpha, \mu} |\sigma_\ell(\mathbf{M}_{\alpha, \mu})|^2 \leq \xi^2 \cdot (\mathfrak{f} - t + 1)(A^2 + (t - 1)B^2).$$

This bound is independent of ℓ , so taking square roots yields an upper-bound for $\|\mathbf{M}\|_{2,2}$ by Eq. (14).

It remains to account for the rows of $\xi \cdot \mathbf{V}_\Xi \cdot \mathbf{V}_S^{-1}$ indexed by T^* . Each $\mu \in T^*$ indexes a row of $\mathbf{V}_S$, so row μ of $\mathbf{V}_{T^*} \cdot \mathbf{V}_S^{-1}$ is the unit row vector selecting the column μ . Hence, up to a permutation of rows, $\xi \cdot \mathbf{V}_\Xi \cdot \mathbf{V}_S^{-1}$ is $\mathbf{M}$ stacked on $\xi \cdot \mathbf{P}$, where $\mathbf{P} \in \{0, 1\}^{(t-1) \times t}$ selects the entries of a vector indexed by T^* . For any $\mathbf{x} \in \mathcal{K}^t$, since ξ is an integer and $\|\mathbf{P}\mathbf{x}\|_2 \leq \|\mathbf{x}\|_2$,

$$\begin{aligned} \|\xi \cdot \mathbf{V}_\Xi \cdot \mathbf{V}_S^{-1} \cdot \mathbf{x}\|_2^2 &= \|\mathbf{M}\mathbf{x}\|_2^2 + \xi^2 \cdot \|\mathbf{P}\mathbf{x}\|_2^2 \\ &\leq (\|\mathbf{M}\|_{2,2}^2 + \xi^2) \cdot \|\mathbf{x}\|_2^2 \\ &\leq \xi^2 \cdot ((\mathfrak{f} - t + 1)(A^2 + (t - 1)B^2) + 1) \cdot \|\mathbf{x}\|_2^2. \end{aligned}$$

Maximising over T^* yields the stated bound on $\gamma(t)$.

Bound on θ . This quantity concerns the same matrix $\mathbf{M} := \xi \cdot \mathbf{V}_{\Xi \setminus T^*} \cdot \mathbf{V}_{\{0\} \cup T^*}^{-1}$ as above. In particular, $\theta(t) = \max_{T^*} \|\mathbf{M} \cdot \mathbf{v}_0\|$ is the ℓ_∞ -norm of the column with $\mu = 0$. We have

$$\theta(t) \leq \max_{\alpha} \|\mathbf{M}_{\alpha, 0}\| \leq \max_{\alpha} \|\sigma(\mathbf{M}_{\alpha, 0})\|_\infty \leq \xi \cdot A,$$

where the second inequality is by Eq. (10). Maximising over T^* gives the bound on $\theta(t)$. $\square$

Selecting subtractive set elements that are evenly spread among the $\mathfrak{f}$ monomials reduces the effective ring degree and can lead to better upper bounds. This is observed in [CLW25] and formalised in [KLL⁺26, Section 5.1], and is also the choice in our implementation, captured by Lemma 4.

Proof (of Lemma 4). Since $\mathfrak{f}$ is a power of two and $K \mid \mathfrak{f}$, the element $\eta = \zeta^{\mathfrak{f}/K}$ is a primitive K -th root of unity with $\eta^{K/2} = \zeta^{\mathfrak{f}/2} = -1$, so the subring $\mathcal{R}' := \mathbb{Z}[\eta] \subseteq \mathcal{R}$ is isomorphic to the K -th cyclotomic ring $\mathbb{Z}[Y]/(Y^{K/2} + 1)$, of degree $K/2$. The set $\Xi' = \{\eta^i : 0 \leq i < K\}$ is precisely the set of powers of η , which is a ξ -subtractive set over $\mathcal{R}'$ for the same slack $\xi = 2^{\lceil \log t \rceil}$ by Lemma 3 (as K is also a power of two). Applying Lemma 8 over $\mathcal{R}'$ gives the four bounds with conductor K and degree $K/2$.

It remains to note that all the norms involved are unchanged when an element of $\mathcal{R}'$ is viewed in $\mathcal{R}$. Any $a = \sum_{j=0}^{K/2-1} c_j \cdot \eta^j \in \mathcal{R}'$ equals $\sum_j c_j \cdot \zeta^{(\mathfrak{f}/K)j}$, whose exponents $(\mathfrak{f}/K)j < \varphi$ are distinct, so the coefficient vector of a in the power basis of $\mathcal{R}$ is that of a in $\mathcal{R}'$ padded with zeros. Hence $\|a\|_p$ agrees in $\mathcal{R}$ and $\mathcal{R}'$ for every p , and the canonical-embedding magnitudes $|\sigma_\ell(a)|$ used in the proof of Lemma 8 are those of $\mathcal{R}'$, which give the trigonometric factors with argument $\pi k/K$.

For γ , fix $T^* \subseteq_{t-1} \Xi'$, set $S := \{0\} \cup T^*$, and let $\mathbf{M} := \xi \cdot \mathbf{V}_{\Xi' \setminus T^*} \cdot \mathbf{V}_S^{-1}$ be the matrix with entries in $\mathcal{R}'$, having $K - t + 1$ rows indexed by $\alpha \in \Xi' \setminus T^*$ and t columns indexed by $\mu \in S$. Viewing $\mathbf{M}$ as a matrix over $\mathcal{R}$, Eq. (14) bounds its ℓ_2 -induced operator norm by $\|\mathbf{M}\|_{2,2} \leq \max_{\ell} \|\sigma_\ell(\mathbf{M})\|_F$, where ℓ ranges over the φ embeddings of $\mathcal{R}$. By the previous paragraph, the magnitude $|\sigma_\ell(\mathbf{M}_{\alpha, \mu})|$ viewed in $\mathcal{R}$ equals the magnitude in $\mathcal{R}'$, whose trigonometric factors carry the argument $\pi k/K$. Hence the bounds $|\sigma_\ell(\mathbf{M}_{\alpha, 0})| \leq \xi A$ and $|\sigma_\ell(\mathbf{M}_{\alpha, \mu})| \leq \xi B$ for $\mu \in T^*$, established in the proof of Lemma 8, hold with A and B evaluated at conductor K . As each of the $K - t + 1$ rows carries one entry of the first kind and $t - 1$ of the second,

$$\|\sigma_\ell(\mathbf{M})\|_F^2 = \sum_{\alpha, \mu} |\sigma_\ell(\mathbf{M}_{\alpha, \mu})|^2 \leq \xi^2 \cdot (K - t + 1)(A^2 + (t - 1)B^2)$$

for every ℓ . Accounting for the $t - 1$ rows of $\xi \cdot \mathbf{V}_{\Xi'} \cdot \mathbf{V}_S^{-1}$ indexed by T^* exactly as in the proof of Lemma 8 gives

$$\|\xi \cdot \mathbf{V}_{\Xi'} \cdot \mathbf{V}_S^{-1}\|_{2,2}^2 \leq \|\mathbf{M}\|_{2,2}^2 + \xi^2 \leq \xi^2 \cdot ((K - t + 1)(A^2 + (t - 1)B^2) + 1).$$

Taking square roots and maximising over T^* yields the stated bound for γ . $\square$

D Proof of Theorem 1 (FHE Leakage Security)

Theorem 1 (Leakage Security). Let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field of degree φ and expansion factor $\mathbf{c}_\infty$ w.r.t. the coefficient ℓ_∞ -norm. Let $\mathcal{R} = \mathbb{Z}[\zeta]$ be its ring of integers. Let $L, D \in \mathbb{N}$ and FHE be an FHE scheme with $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} . Then, for any PPT adversary $\mathcal{A}$ against the IND-CPA-with- (D, L, χ) -Leakage security of FHE that makes at most $Q \in \text{poly}(\lambda)$ queries to the oracle LeakO, there exists a PPT adversary $\mathcal{B}$ against the IND-CPA security of FHE such that

$$\text{Adv}_{\text{FHE}, \mathcal{A}, (D, L, \chi)}^{\text{IND-CPA-wL}}(\lambda) \leq 2 \cdot \left(Q \cdot \varepsilon_{\text{FHE}} + \frac{C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B}{\chi} \right) + \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda)$$

where $C = \sqrt{\pi/2}$ is a constant. In particular, if $\chi > \lambda^{\omega(1)} \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B$, FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption and is IND-CPA secure, then it is also IND-CPA-with- (D, L, χ) -Leakage secure.

Proof. We consider the following sequence of hybrids.

Hyb_{b,0}: This is the IND-CPA-with- (D, L, χ) -Leakage security experiment for challenge bit b . To recall, the view of the adversary consists of

public parameters pp , public key pk ,
ciphertexts ctxt from EncO , ChalO or EvalO , and their underlying plaintexts μ_b ,
leakages $\mathbf{u} \cdot \hat{e} + \mathbf{f}$ from LeakO on ciphertexts ctxt .

Hyb_{b,1}: We change the experiment as follows. For each query $(\text{id}, \mathbf{u})$ to the oracle LeakO , compute $\hat{e} := \hat{c}_1 - \text{sk}^T \cdot \hat{c}_0 - \xi^{-2} \left\lfloor \frac{q}{p} \right\rfloor \mu \bmod q$. If $\|\hat{e}\| > B$, then set $L_{\text{Leak}}[\text{id}] := \mathbf{f} \leftarrow \$ D_{\mathcal{R}, \chi}^L$ and use this as the oracle answer. Otherwise, there is no change, i.e. set $L_{\text{Leak}}[\text{id}] := \mathbf{u} \cdot \hat{e} + \mathbf{f}$.

Hyb_{b,2}: We change the experiment as follows. For each $(\text{id}, \mathbf{u})$ to the oracle LeakO , sample $\mathbf{f} \leftarrow \$ D_{\mathcal{R}, \chi}^L$. The leakage is computed as $L_{\text{Leak}}[\text{id}] := \mathbf{f}$.

For $i = 0, 1, 2$, let $\text{Adv}_i(\mathcal{A})$ denote the advantage of $\mathcal{A}$ in experiments $(\text{Hyb}_{0,i}, \text{Hyb}_{1,i})$, that is

$$\text{Adv}_i(\mathcal{A}) := |\Pr[\text{Hyb}_{0,i,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,i,\mathcal{A}}(1^\lambda) = 1]|$$

By definition of $\text{Hyb}_{b,0}$, one has

$$\text{Adv}_{\text{FHE}, \mathcal{A}, (D, L, \chi)}^{\text{IND-CPA-wL}}(\lambda) = \text{Adv}_0(\mathcal{A})$$

– By Lemma 9, if FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} , we have

$$\text{Adv}_0(\mathcal{A}) \leq \text{Adv}_1(\mathcal{A}) + 2 \cdot Q \cdot \varepsilon_{\text{FHE}}.$$

– By Lemma 10, we have

$$\text{Adv}_1(\mathcal{A}) \leq \text{Adv}_2(\mathcal{A}) + 2 \cdot \frac{C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B}{\chi}.$$

– By Lemma 11, for every PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_2(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda).$$

Putting everything together, one obtains

$$\text{Adv}_{\text{FHE}, \mathcal{A}, (D, L, \chi)}^{\text{IND-CPA-wL}}(\lambda) \leq 2 \cdot \left(Q \cdot \varepsilon_{\text{FHE}} + \frac{C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B}{\chi} \right) + \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda),$$

as claimed.

Lemma 9. Adopt the notations in Theorem 1. We have $\text{Adv}_0(\mathcal{A}) \leq \text{Adv}_1(\mathcal{A}) + 2 \cdot Q \cdot \varepsilon_{\text{FHE}}$.

Proof. By the definition of $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} , the norm of each $\hat{e}$ computed by the **LeakO** is bounded by B except with probability ε_{FHE} . By a hybrid argument over the Q queries made by $\mathcal{A}$ to **LeakO**, it follows that, for each $b \in \{0, 1\}$, the statistical distance between $\text{Hyb}_{b,0}$ and $\text{Hyb}_{b,1}$ is at most $Q \cdot \varepsilon_{\text{FHE}}$. Thus, by the triangle inequality,

$$\begin{aligned}
\text{Adv}_0(\mathcal{A}) &= |\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1]| \\
&= \left| \begin{aligned} &\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1] \end{aligned} \right| \\
&\leq |\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1]| \\
&\quad + |\Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1]| \\
&\quad + |\Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1]| \\
&\leq Q \cdot \varepsilon_{\text{FHE}} + \text{Adv}_1(\mathcal{A}) + Q \cdot \varepsilon_{\text{FHE}} \\
&= \text{Adv}_1(\mathcal{A}) + 2 \cdot Q \cdot \varepsilon_{\text{FHE}}. \quad \square
\end{aligned}$$

Lemma 10. *Adopt the notations in Theorem 1. We have $\text{Adv}_1(\mathcal{A}) \leq \text{Adv}_2(\mathcal{A}) + 2 \cdot \frac{C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B}{\chi}$.*

Proof. The only difference between $\text{Hyb}_{b,1}$ and $\text{Hyb}_{b,2}$ is that whenever $\|\hat{e}\| \leq B$, the answer from **LeakO** is changed from $\mathbf{u} \cdot \hat{e} + \mathbf{f}$ to $\mathbf{f}$. For each such query $\mathbf{u}$, it holds that

$$\|\mathbf{u} \cdot \hat{e}\|_2 \leq \sqrt{\varphi \cdot L} \cdot \|\mathbf{u} \cdot \hat{e}\|_\infty \leq \sqrt{\varphi \cdot L} \cdot \mathbf{c}_\infty \cdot \|\mathbf{u}\|_\infty \cdot \|\hat{e}\|_\infty \leq \sqrt{\varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B,$$

where we use that $\|\mathbf{u}\|_\infty \leq D$ by the norm-check in **leakO**.

For $b \in \{0, 1\}$, we invoke Lemma 7 with $n = L$ and norm bound $B' := \sqrt{\varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B$. The (unbounded) reduction is trivial, so we only give a sketch. It simulates $\text{Hyb}_{b,1}$ and $\text{Hyb}_{b,2}$ as follows: on each **leakO** query it computes $\hat{e}$ using **sk**, queries its oracle $\mathcal{O}_b$ on $\mathbf{c} := \mathbf{u} \cdot \hat{e}$ if $\|\hat{e}\| \leq B$ and $\mathbf{c} := \mathbf{0}$ otherwise (so $\|\mathbf{c}\|_2 \leq B'$ in both cases), and answers the query with the oracle output. The remaining parts are identical in both hybrids, which are computed by the reduction itself. It is direct to see that it perfectly simulates either hybrid depending on the challenge bit in the experiment in Lemma 7. In particular, when $\|\hat{e}\| > B$, the reduction answers $\mathbf{f}$ as defined in $\text{Hyb}_{b,1}$. Since there are at most Q queries, Lemma 7 gives

$$|\Pr[\text{Hyb}_{b,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{b,2,\mathcal{A}}(1^\lambda) = 1]| \leq (C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B) / \chi$$

for both $b \in \{0, 1\}$. Then by the triangle inequality,

$$\begin{aligned}
\text{Adv}_1(\mathcal{A}) &= |\Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1]| \\
&= \left| \begin{aligned} &\Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,2,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{0,2,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,2,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{1,2,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] \end{aligned} \right| \\
&\leq |\Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,2,\mathcal{A}}(1^\lambda) = 1]| \\
&\quad + |\Pr[\text{Hyb}_{0,2,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,2,\mathcal{A}}(1^\lambda) = 1]| \\
&\quad + |\Pr[\text{Hyb}_{1,2,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1]| \\
&\leq (C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B) / \chi + \text{Adv}_2(\mathcal{A}) + (C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B) / \chi \\
&= \text{Adv}_2(\mathcal{A}) + 2 \cdot (C \cdot \sqrt{Q \cdot \varphi \cdot L} \cdot \mathbf{c}_\infty \cdot D \cdot B) / \chi. \quad \square
\end{aligned}$$

Lemma 11. *Adopt the notations in Theorem 1. We have $\text{Adv}_2(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda)$.*

Proof. Consider a PPT algorithm $\mathcal{A}$ distinguishing $\text{Hyb}_{0,2}$ and $\text{Hyb}_{1,2}$ with advantage $\text{Adv}_2(\mathcal{A})$. We construct a reduction $\mathcal{B}$ against the IND-CPA security of FHE, which proceeds as follows:

- It receives (pp, pk) from the IND-CPA challenger and sends them to $\mathcal{A}$.

- It internally maintains lists $L_{\text{Query}}, L_{\text{leak}}$, initially empty. It answers $\mathcal{A}$'s queries to $\text{EncO}, \text{EvalO}, \text{LeakO}$ by honestly performing the oracle computations (which are identical in $\text{Hyb}_{0,2}$ and $\text{Hyb}_{1,2}$) itself, and forwards the ChalO queries to its own ChalO oracle available from the IND-CPA challenger. In more detail, it answers the oracle queries as follows:
 - $\text{EncO}(\text{id}, \mu)$: If $L_{\text{Query}}[\text{id}] = \emptyset$, it computes $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu)$ and sets $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu, \mu)$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.
 - $\text{ChalO}(\text{id}, \mu_0, \mu_1)$: If $L_{\text{Query}}[\text{id}] = \emptyset$, it queries its own ChalO on $(\text{id}, \mu_0, \mu_1)$ to receive ctxt , and sets $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu_0, \mu_1)$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.
 - $\text{EvalO}(\text{id}, f, (\text{id}_i)_{i \in I})$: If $L_{\text{Query}}[\text{id}] = \emptyset$, it asserts $L_{\text{Query}}[\text{id}_i] \neq \emptyset$ and parses $(\text{ctxt}_i, \mu_{0,i}, \mu_{1,i}) \leftarrow L_{\text{Query}}[\text{id}_i]$ for each $i \in I$, then computes $\text{ctxt}' \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_i)_{i \in I})$, and sets $L_{\text{Query}}[\text{id}] := (\text{ctxt}', f((\mu_{0,i})_{i \in I}), f((\mu_{1,i})_{i \in I}))$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.
 - $\text{LeakO}(\text{id}, \mathbf{u} \in \mathcal{R}^L)$: If $L_{\text{leak}}[\text{id}] = \emptyset$, it asserts $\|\mathbf{u}\| \leq D$ and $L_{\text{Query}}[\text{id}] \neq \emptyset$, parses $(\text{ctxt}, \mu_0, \mu_1) \leftarrow L_{\text{Query}}[\text{id}]$ and asserts $\mu_0 = \mu_1$, samples $\mathbf{f} \leftarrow \$ D_{\mathcal{R}, \chi}^L$, and sets $L_{\text{leak}}[\text{id}] := \mathbf{f}$. It returns $L_{\text{leak}}[\text{id}]$ to $\mathcal{A}$.
- Eventually, it receives b' from $\mathcal{A}$ and returns b' .

If $\mathcal{B}$ is interacting with the IND-CPA experiment 0, then queries to $\text{ChalO}(\text{id}, \mu_0, \mu_1)$ return $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu_0)$. Therefore, the view of $\mathcal{A}$ is identical to $\text{Hyb}_{0,2}$.

If $\mathcal{B}$ is interacting with the IND-CPA experiment 1, then queries to $\text{ChalO}(\text{id}, \mu_0, \mu_1)$ return $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu_1)$. Therefore, the view of $\mathcal{A}$ is identical to $\text{Hyb}_{1,2}$.

It follows that $\mathcal{B}$ perfectly simulates the view of either $\text{Hyb}_{0,2}$ or $\text{Hyb}_{1,2}$ to $\mathcal{A}$, if it is interacting either with the IND-CPA experiment 0 or 1. Therefore $\text{Adv}_2(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}}^{\text{IND-CPA}}(\lambda)$ as claimed. $\square$

E Proof of Theorem 2 (Tulva Correctness)

Theorem 2 (Correctness). *If $\xi \in \mathbb{Z}$, FHE has $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption with error ε_{FHE} and*

$$\frac{q}{2p} \geq \frac{|q \bmod p|}{2} + \xi^2 \cdot B + \epsilon_\infty \cdot \left(\sqrt{\lambda} \cdot m \cdot \xi^2 \cdot \bar{\chi} \cdot \frac{\beta}{2} + t \cdot \rho \cdot (\sqrt{\lambda} \cdot \chi + 2^{z-1}) \right). \quad (5)$$

Then, thFHE in Fig. 3 has correctness error $\varepsilon \leq \varepsilon_{\text{FHE}} + \varphi \cdot (m+t) \cdot 2^{-\lambda}$. In particular, if $\min\{\log(1/\varepsilon_{\text{FHE}}), \lambda - \log \varphi - \log(m+t)\} \geq \omega(\log \lambda)$, then thFHE is correct.

Proof. First observe that, for any thFHE ciphertext $(\text{ctxt}_{\text{FHE}}, \mathbf{t})$, the third and fourth lines of ParDec and Combine in Fig. 3 are identical, i.e. they compute the same $\mathbf{x} = \mathbf{G}_{\beta, q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$, where $(\hat{\mathbf{c}}_0, \hat{c}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$.

The algorithm ParDec , on input $(\text{pk}, \text{sk}_k, \text{ctxt})$, computes the following quantity

$$\text{pd}_k = \lfloor \mathbf{s}_k^T \cdot \mathbf{c}_0 + e_k \rfloor_{2^z} = \mathbf{s}_k^T \cdot \mathbf{A} \cdot \mathbf{x} + e_k + d_k \bmod q, \quad (15)$$

where $d_k := \lfloor y \rfloor_{2^z} - y \in \mathcal{R}$ is the rounding error, for y the representative of $\mathbf{s}_k^T \cdot \mathbf{c}_0 + e_k \bmod q$ with coefficients in $[-q/2, q/2)$. Writing y_i for the coefficients of y , each coefficient of d_k is $2^z \cdot (\lfloor y_i/2^z \rfloor - y_i/2^z)$, so $\|d_k\| \leq 2^{z-1}$.

Below, for any recovery set T , denote by $\mathbf{e}_T := (e_k + d_k)_{k \in T}$ the vertical concatenation of the errors $e_k + d_k$. The algorithm Combine , on input $(\text{pk}, T, (\text{pd}_k)_{k \in T}, \text{ctxt})$, computes pd and c_1 which are

$$\begin{aligned} \text{pd} &= \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot (\text{pd}_k)_{k \in T} \bmod q \\ &= \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot (\mathbf{s}_k^T \cdot \mathbf{A} \cdot \mathbf{x} + e_k + d_k)_{k \in T} \bmod q \quad // \text{ by Eq. (15)} \\ &= \xi^2 \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{V}_T \cdot \mathbf{R} \cdot \mathbf{A} \cdot \mathbf{x} + \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T \bmod q \quad // (\mathbf{s}_k)_{k \in T} = \xi \cdot \mathbf{V}_T \cdot \mathbf{R} \bmod q, (e_k + d_k)_{k \in T} = \mathbf{e}_T \\ &= \xi^2 \cdot \mathbf{v}_0^T \cdot \mathbf{R} \cdot \mathbf{A} \cdot \mathbf{x} + \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T \bmod q \\ &= \xi^2 \cdot \mathbf{r}^T \cdot \mathbf{A} \cdot \mathbf{x} + \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T \bmod q \quad // \mathbf{r}^T = \mathbf{v}_0^T \cdot \mathbf{R} \bmod q \end{aligned} \quad (16)$$

and

$$c_1 = \xi^2 \cdot \mathbf{b}^T \cdot \mathbf{x} \bmod q$$

$$\begin{aligned}
&= \xi^2 \cdot (\mathbf{r}^\top \cdot \mathbf{A} + \mathbf{e}^\top + \mathbf{z}^\top) \cdot \mathbf{x} \bmod q \quad // \quad \mathbf{b}^\top = \mathbf{r}^\top \mathbf{A} + \mathbf{e}^\top + \mathbf{z}^\top \bmod q \\
&= \xi^2 \cdot (\mathbf{r}^\top \cdot \mathbf{A} + \mathbf{e}^\top + (\hat{\mathbf{s}}^\top, \mathbf{0}_h^\top) \cdot \mathbf{G}_{\beta,q}) \cdot \mathbf{x} \bmod q \quad // \quad \mathbf{z}^\top = ((\hat{\mathbf{s}}^\top, \mathbf{0}_h^\top) \cdot \mathbf{G}_{\beta,q} \bmod q) \\
&= \xi^2 \cdot \mathbf{r}^\top \cdot \mathbf{A} \cdot \mathbf{x} + \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} + \xi^2 \cdot ((\hat{\mathbf{s}}^\top, \mathbf{0}_h^\top) \cdot \mathbf{G}_{\beta,q} \cdot \mathbf{x}) \bmod q \\
&= \xi^2 \cdot \mathbf{r}^\top \cdot \mathbf{A} \cdot \mathbf{x} + \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} + \xi^2 \cdot \left((\hat{\mathbf{s}}^\top, \mathbf{0}_h^\top) \cdot \mathbf{G}_{\beta,q} \cdot \mathbf{G}_{\beta,q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \right) \bmod q \quad // \quad \mathbf{x} = \mathbf{G}_{\beta,q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \\
&= \xi^2 \cdot \mathbf{r}^\top \cdot \mathbf{A} \cdot \mathbf{x} + \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} + \xi^2 \cdot \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0 \bmod q.
\end{aligned} \tag{17}$$

Putting Eqs. (16) and (17) together, we have

$$\begin{aligned}
c_1 - \text{pd} &= (\xi^2 \cdot \mathbf{r}^\top \cdot \mathbf{A} \cdot \mathbf{x} + \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} + \xi^2 \cdot \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0) - (\xi^2 \cdot \mathbf{r}^\top \cdot \mathbf{A} \cdot \mathbf{x} + \xi \cdot \mathbf{v}_0^\top \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T) \bmod q \\
&= \xi^2 \cdot \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0 + \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} - \xi \cdot \mathbf{v}_0^\top \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T \bmod q \\
&= \xi^2 \cdot \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0 + f \bmod q,
\end{aligned} \tag{18}$$

where in the last line we use the shorthand

$$f := \xi^2 \cdot \mathbf{e}^\top \cdot \mathbf{x} - \xi \cdot \mathbf{v}_0^\top \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T.$$

Using Eq. (18), we obtain

$$\begin{aligned}
\hat{\mu} &= \xi^2 \cdot \hat{c}_1 - (c_1 - \text{pd}) \bmod q \\
&= \xi^2 \cdot \hat{c}_1 - \xi^2 \cdot \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0 - f \bmod q \\
&= \xi^2 \cdot \left(\xi^{-2} \cdot \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \hat{e} \right) - f \bmod q \\
&= \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \xi^2 \cdot \hat{e} - f \bmod q \\
&= \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \xi^2 \cdot \hat{e} - f + q \cdot h \quad // \quad \text{for some } h \in \mathcal{R}
\end{aligned} \tag{19}$$

where the third equality follows from the $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption property of FHE (Definition 4), which guarantees that

$$\hat{c}_1 - \hat{\mathbf{s}}^\top \cdot \hat{\mathbf{c}}_0 \bmod q = \xi^{-2} \cdot \left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \hat{e}$$

except with probability at most ε_{FHE} . Finally, the algorithm **Combine** returns

$$\begin{aligned}
\mu' &= \left\lfloor \frac{p}{q} \cdot \hat{\mu} \right\rfloor \bmod p \\
&= \left\lfloor \frac{p}{q} \cdot \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \mu + \xi^2 \cdot \hat{e} - f + q \cdot h \right) \right\rfloor \bmod p \quad // \quad \text{By Eq. (19)} \\
&= \left\lfloor \frac{p}{q} \cdot \left(\frac{q}{p} - \left\{ \frac{q}{p} \right\} \right) \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f + p \cdot h \right\rfloor \bmod p \quad // \quad \frac{q}{p} = \left\lfloor \frac{q}{p} \right\rfloor + \left\{ \frac{q}{p} \right\} \\
&= \left\lfloor \mu - \frac{p}{q} \cdot \left\{ \frac{q}{p} \right\} \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f + p \cdot h \right\rfloor \bmod p \\
&= \mu + \left\lfloor -\frac{p}{q} \cdot \left\{ \frac{q}{p} \right\} \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f \right\rfloor + p \cdot h \bmod p \\
&= \mu + \left\lfloor -\frac{p}{q} \cdot \left\{ \frac{q}{p} \right\} \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f \right\rfloor \bmod p.
\end{aligned}$$

Thus, the plaintext μ is correctly recovered as long as

$$\begin{aligned}
\left\lfloor -\frac{p}{q} \cdot \left\{ \frac{q}{p} \right\} \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f \right\rfloor = 0 &\iff \left\| -\frac{p}{q} \cdot \left\{ \frac{q}{p} \right\} \cdot \mu + \frac{p}{q} \cdot \xi^2 \cdot \hat{e} - \frac{p}{q} \cdot f \right\| \leq \frac{1}{2} \\
&\iff \left\| -\left\{ \frac{q}{p} \right\} \cdot \mu + \xi^2 \cdot \hat{e} - f \right\| \leq \frac{q}{2p},
\end{aligned}$$

which is implied by

$$\left\| \left\{ \frac{q}{p} \right\} \cdot \mu \right\| + \|\xi^2 \cdot \hat{e}\| + \|f\| \leq \frac{q}{2p}. \quad (20)$$

In the rest, we show that Eq. (5) in the theorem statement implies Eq. (20) in the above except with probability at most $\varepsilon_{\text{FHE}} + \varphi \cdot (m+t) \cdot 2^{-\lambda}$, which completes the proof. We have

$$\begin{aligned} \left\| \left\{ \frac{q}{p} \right\} \cdot \mu \right\| &\leq \left\| \left\{ \frac{q}{p} \right\} \cdot \frac{p}{2} \right\| = \frac{|q \bmod p|}{2}, \\ \|\xi^2 \cdot \hat{e}\| &= \|\xi^2\| \cdot \|\hat{e}\| \quad // \xi \in \mathbb{Z} \\ &\leq \xi^2 \cdot B, \quad // \text{By } (\mathcal{R}, \hat{n}, p, q, \xi, B)\text{-linear-decryption with error } \varepsilon_{\text{FHE}} \\ \|f\| &= \|\xi^2 \cdot \mathbf{e}^T \cdot \mathbf{x} - \xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T\| \\ &\leq \|\xi^2 \cdot \mathbf{e}^T \cdot \mathbf{x}\| + \|\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T\| \\ &\leq \left(\mathbf{c}_\infty \cdot m \cdot \|\xi^2\| \cdot \|\mathbf{e}\| \cdot \|\mathbf{x}\| \quad // \xi \in \mathbb{Z}, \|\mathbf{e}^T \cdot \mathbf{x}\| \leq \mathbf{c}_\infty \cdot m \cdot \|\mathbf{e}\| \cdot \|\mathbf{x}\| \right. \\ &\quad \left. + \mathbf{c}_\infty \cdot t \cdot \|\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1}\| \cdot \|\mathbf{e}_T\| \right) \quad // \|\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1} \cdot \mathbf{e}_T\| \leq \mathbf{c}_\infty \cdot t \cdot \|\xi \cdot \mathbf{v}_0^T \cdot \mathbf{V}_T^{-1}\| \cdot \|\mathbf{e}_T\| \\ &\leq \left(\mathbf{c}_\infty \cdot m \cdot \xi^2 \cdot \bar{\chi} \cdot \sqrt{\lambda} \cdot \frac{\beta}{2} \quad // \|\mathbf{e}\| \leq \bar{\chi} \cdot \sqrt{\lambda} \text{ by Eq. (8)}, \|\mathbf{x}\| \leq \frac{\beta}{2}, \xi \in \mathbb{Z} \right. \\ &\quad \left. + \mathbf{c}_\infty \cdot t \cdot \rho \cdot (\chi \cdot \sqrt{\lambda} + 2^{z-1}) \right) \quad // \|\mathbf{e}_T\| \leq \chi \cdot \sqrt{\lambda} + 2^{z-1} \text{ by Eq. (8)}, \|d_k\| \leq 2^{z-1} \\ &= \mathbf{c}_\infty \cdot \left(\sqrt{\lambda} \cdot m \cdot \xi^2 \cdot \bar{\chi} \cdot \frac{\beta}{2} + t \cdot \rho \cdot (\sqrt{\lambda} \cdot \chi + 2^{z-1}) \right), \end{aligned}$$

where we recall that ρ in the above is a quality measurement of Ξ (Definition 1). Putting the above bounds together with Eq. (5) yields Eq. (20) as claimed. Indeed, if $\min\{\log(1/\varepsilon_{\text{FHE}}), \lambda - \log \varphi - \log(m+t)\} \geq \omega(\log \lambda)$, then the above bounds hold except with probability negligible in λ . $\square$

F Proofs for Theorem 3 (Tulva Security)

This section provides the proof of Theorem 3 and is structured as follows. In Appendix F.1, we recall the theorem statement and state its proof which relies on several bridging lemmas. Then, in Appendix F.2, we state and prove these bridging lemmas Lemmas 12 to 15. One of the bridging lemmas (Lemma 12) will further rely on a separate utility lemma (Lemma 21), the latter being a generalisation of a lemma in [CLW25] and is to be provided in Appendix G.

F.1 Theorem Statement and Proof

Theorem 3 (Security). *Let FHE be an FHE scheme with $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption and Ξ be a ξ -subtractive set of cardinality K with quality measurements (ρ, γ, θ) (Definition 1). Let $\epsilon \in (0, 1/2]$, $Q \in \text{poly}(\lambda)$, q, ξ be coprime, $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$ be such that there exists $\hat{\chi} > 0$ with $\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2)^{-1} - \hat{\chi}^{-2}}$, and $\chi_2^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$ be such that $\gamma \cdot \sqrt{\varphi m} \cdot \beta/2 \geq 1$ and $\sqrt{2QK} \cdot \varphi m \cdot \mathbf{c}_2 \cdot \mathbf{c}_t^\otimes \cdot \gamma \cdot \beta/2 \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi_2^*)^2)^{-1} - \bar{\chi}^{-2}}$, where $\mathbf{c}_t^\otimes := 1$ if t is a power of two and $\mathbf{c}_t^\otimes := \sqrt{t}$ otherwise. Let $\mathcal{D}_t := \{\tilde{\mathbf{t}} : \mathbf{t} \leftarrow \mathcal{R}_q^h, \tilde{\mathbf{t}} := \mathbf{G}_{\beta, q}^{-1}(\mathbf{t})\}$. Define the parameter sets:*

- thFHE: $\text{par} = (\text{FHE}, \mathcal{R}, n, m, h, q, \beta, \bar{\chi}, \chi, z, \Xi)$,
- LWE: $\text{par}_1 := (\mathcal{R}, h \lceil \log_\beta q \rceil, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_t)$,
- LWE: $\text{par}_2 := (\mathcal{R}, n, m, q, \chi_2^*)$,
- FHE: $\text{par}_3 := (\theta, K, \chi)$.

Consider the (t, K) -threshold FHE scheme $\text{thFHE}_{\text{par}}$ in Fig. 3. For all PPT $\mathcal{A}$ that requests partial decryptions on at most Q ciphertexts (i.e. in the Q -bounded IND-CPA experiment of Definition 7), there exist PPT $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that $\text{Adv}_{\text{thFHE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) \leq \varepsilon$ where

$$\varepsilon := 2 \cdot t \cdot \left(\text{Adv}_{\text{par}_1, \mathcal{B}_1}^{\text{LWE}}(\lambda) + \text{Adv}_{\text{par}_2, \mathcal{B}_2}^{\text{LWE}}(\lambda) \right) + \frac{12 \cdot Q \cdot \epsilon}{1 - \epsilon} + \text{Adv}_{\text{FHE}, \text{par}_3, \mathcal{B}_3}^{\text{IND-CPA-wL}}(\lambda) \\ + 4 \cdot t \cdot \left(2QK + \frac{3}{1 - \epsilon} \right) \cdot \epsilon.$$

Proof. Let thFHE' denote the scheme in Fig. 3 with the step $\text{pd}_k \leftarrow \lfloor \text{pd}_k \rfloor_{2^z}$ of ParDec removed, i.e. the scheme without bit dropping. Since $\lfloor \cdot \rfloor_{2^z}$ is a deterministic public function applied only to the outputs of ParDec , any adversary $\mathcal{A}$ against thFHE yields an adversary $\mathcal{A}'$ against thFHE' that runs $\mathcal{A}$, forwards all its oracle queries, and applies $\lfloor \cdot \rfloor_{2^z}$ to every partial decryption before passing it to $\mathcal{A}$. The adversary $\mathcal{A}'$ is PPT, requests partial decryptions on the same ciphertexts as $\mathcal{A}$, and perfectly simulates $\text{Exp}_{\text{thFHE}, \mathcal{A}}^b$ for both $b \in \{0, 1\}$, so $\text{Adv}_{\text{thFHE}, \mathcal{A}}^{\text{IND-CPA}}(\lambda) = \text{Adv}_{\text{thFHE}', \mathcal{A}'}^{\text{IND-CPA}}(\lambda)$. It therefore suffices to show $\text{Adv}_{\text{thFHE}', \mathcal{A}'}^{\text{IND-CPA}}(\lambda) \leq \varepsilon$ for every PPT $\mathcal{A}$ against thFHE' that requests partial decryptions on at most Q ciphertexts, with ε as in Theorem 3. In the rest of this proof, $\mathcal{A}$ denotes such an adversary and partial decryptions are those in $\text{Exp}_{\text{thFHE}', \mathcal{A}}^b$.

Throughout this proof (and in Lemmas 12 to 15 and 21 which support this proof), we adopt the following notations:

- For any subset $\mathcal{C} \subseteq [K]$, denote by $\mathbf{V}_{\mathcal{C}} \in \mathcal{R}^{|\mathcal{C}| \times t}$ the submatrix of $\mathbf{V}$ whose rows are indexed by $\mathcal{C}$. Similarly, $\mathbf{S}_{\mathcal{C}} \in \mathcal{R}_q^{|\mathcal{C}| \times n}$ is the submatrix of $\mathbf{S} = \mathbf{V} \cdot \mathbf{R}$ whose rows are indexed by $\mathcal{C}$, i.e., $\mathbf{S}_{\mathcal{C}} = \mathbf{V}_{\mathcal{C}} \cdot \mathbf{R}$.
- For any set $\mathcal{C} \subset [K]$ of corrupt parties with $|\mathcal{C}| < t$, we define:
 - $\mathcal{H} := [K] \setminus \mathcal{C}$ the set of honest (i.e. non-corrupt) parties, and
 - $\mathbf{T}_{\mathcal{C}} \in \mathcal{R}^{t \times t}$ a matrix such that it holds $\xi^{-1} \cdot \mathbf{T}_{\mathcal{C}} \cdot \mathbf{V}_{\mathcal{C}^*} = \mathbf{I}_t$ for some size- t set $\mathcal{C}^*$ satisfying $(\{0\} \cup \mathcal{C}) \subseteq \mathcal{C}^* \subseteq [0, K]$. ($\mathbf{T}_{\mathcal{C}}$ is not unique given $\mathcal{C}$ and we may fix an arbitrary one.)

We consider the following sequence of hybrids.

$\text{Hyb}_{b,0}$: This is the thFHE' security experiment for challenge bit b . To recall, the view of the adversary consists of $\text{pk} = (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ where

$$\mathbf{b}^T := \mathbf{r}^T \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q, \quad \mathbf{z}^T = (\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \mathbf{G}_{\beta, q} \bmod q,$$

corrupt user keys $(\mathbf{s}_k)_{k \in \mathcal{C}}$, ciphertexts $\text{ctxt} = (\text{ctxt}_{\text{FHE}}, \mathbf{t})$ from EncO , ChalO or EvalO , partial decryptions $\text{pd}_k = \mathbf{s}_k^T \mathbf{A} \mathbf{x} + e_k \bmod q$ where $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$ and the vector $\hat{\mathbf{c}}_0$ is obtained from running $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{FHE.Preproc}(\text{ctxt}_{\text{FHE}})$, which is possible because of the $(\mathcal{R}, \hat{n}, p, q, \xi, B)$ -linear-decryption property of FHE. (We recall that Preproc is a public operation, thus the adversary can also compute the above and observe $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1)$.)

$\text{Hyb}_{b,1}$: This experiment is identical to $\text{Hyb}_{b,0}$ except for the following change. For each ctxt , sample a random vector $\mathbf{w}_{\mathbf{x}} \leftarrow \mathcal{R}_q^{t - |\mathcal{C}| - 1}$ and an error $\mathbf{e}_{\mathbf{x}} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$, its partial decryptions are computed as

$$(\text{pd}_k)_{k \in [K]} = \left[\begin{array}{c|c} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_{\mathcal{C}} & \\ \hline & \xi \mathbf{I}_{\mathcal{C}} \end{array} \right] \left[\begin{array}{c} (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{array} \right] + \mathbf{e}_{\mathbf{x}} \bmod q \\ =: \left[\begin{array}{c|c} \mathbf{V}_{\mathcal{H}} \cdot (\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2) & \\ \hline & \xi \mathbf{I}_{\mathcal{C}} \end{array} \right] \left[\begin{array}{c} (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{array} \right] + \mathbf{e}_{\mathbf{x}} \bmod q \quad (21)$$

where we recall $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$ and $\mathbf{z}^T = (\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \mathbf{G}_{\beta, q} \bmod q$.

In the last line, we define the shorthand

$$\mathbf{T}_{\mathcal{C}} = (\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2) \in \mathcal{R}^{t \times 1} \times \mathcal{R}^{t \times t - |\mathcal{C}| - 1} \times \mathcal{R}^{t \times |\mathcal{C}|}, \quad (22)$$

for the block columns of $\mathbf{T}_{\mathcal{C}}$, which will be used in the subsequent hybrids.

Hyb_{b,2}: This experiment is identical to **Hyb_{b,1}** except for the following change. For each **ParDecO** query (id, k) , if $L_{\text{Query}}[\text{id}] = (\text{ctxt}, \mu_0, \mu_1)$ is such that $\mu_0 \neq \mu_1$ and $k \notin \mathcal{C}$, we change the partial decryption answer for k to a uniformly random $\mathcal{R}_q$ element. For corrupt parties $k \in \mathcal{C}$, the answer is kept as in **Hyb_{b,1}**.

Hyb_{b,3}: This experiment is identical to **Hyb_{b,2}** except for the following change. For each **ParDecO** query id , if $L_{\text{Query}}[\text{id}] = (\text{ctxt}, \mu_0, \mu_1)$ is such that $\mu_0 = \mu_1$, we express its partial decryptions as

$$(\text{pd}_k)_{k \in [K]} = \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) \right] \\ + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_{\mathcal{C}}} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_{\mathbf{x}} \bmod q, \quad (23)$$

where $\mu := \mu_0 = \mu_1$, and $\hat{e} \in \mathcal{R}$ is defined as

$$\hat{e} := \hat{c}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 - \xi^{-2} \left\lfloor \frac{q}{p} \right\rfloor \mu \bmod q \quad (24)$$

We recall $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$, and $\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2$ are defined in **Hyb_{b,1}**.

For $i \in [0, 3]$, let $\text{Adv}_i(\mathcal{A})$ denote the advantage of $\mathcal{A}$ in experiments $(\text{Hyb}_{0,i}, \text{Hyb}_{1,i})$, that is $\text{Adv}_i(\mathcal{A}) := |\Pr[\text{Hyb}_{0,i,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,i,\mathcal{A}}(1^\lambda) = 1]|$. By definition of **Hyb_{b,0}**, one has $\text{Adv}_{\text{thFHE}', \mathcal{A}}^{\text{IND-CPA}}(\lambda) = \text{Adv}_0(\mathcal{A})$. By Lemma 12, under the condition stated in Theorem 3, for any PPT $\mathcal{A}$, there exist PPT $\mathcal{B}_1, \mathcal{B}_2$ such that

$$\text{Adv}_0(\mathcal{A}) \leq \text{Adv}_1(\mathcal{A}) + 2 \cdot t \cdot \left(\text{Adv}_{\text{par}_1, \mathcal{B}_1}^{\text{LWE}}(\lambda) + \text{Adv}_{\text{par}_2, \mathcal{B}_2}^{\text{LWE}}(\lambda) + \frac{6 \cdot Q \cdot \epsilon}{t(1 - \epsilon)} + 4 \cdot QK \cdot \epsilon + \frac{6 \cdot \epsilon}{1 - \epsilon} \right).$$

By Lemma 13, $\text{Adv}_1(\mathcal{A}) = \text{Adv}_2(\mathcal{A})$. By Lemma 14, $\text{Adv}_2(\mathcal{A}) = \text{Adv}_3(\mathcal{A})$. By Lemma 15, $\forall$ PPT $\mathcal{A}$, $\exists$ PPT $\mathcal{B}_3$ such that $\text{Adv}_3(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}_3, \text{par}_3}^{\text{IND-CPA-wL}}(\lambda)$. Putting everything together, one obtains the claimed result. Lemmas 12 to 15 and their proofs are provided in Appendix F.2. $\square$

F.2 Proofs of Lemmas 12 to 15

Lemma 12. *Adopt the notations in Theorem 3. Suppose q and ξ are coprime, $0 < \epsilon \leq 1/2$, and the noise parameters χ_1^*, χ_2^* satisfy both of the following:*

- $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$ is such that there exists $\hat{\chi} > 0$ satisfying $\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2)^{-1} - \hat{\chi}^{-2}}$;
- $\chi_2^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$, $\gamma \cdot \sqrt{\varphi m} \cdot \beta/2 \geq 1$, and, with $\mathbf{c}_t^\otimes := 1$ if t is a power of two and $\mathbf{c}_t^\otimes := \sqrt{t}$ otherwise, $\sqrt{2QK} \cdot \varphi m \cdot \mathbf{c}_2 \cdot \mathbf{c}_t^\otimes \cdot \gamma \cdot \beta/2 \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi_2^*)^2)^{-1} - \bar{\chi}^{-2}}$.

Then, for any PPT $\mathcal{A}$, there exist PPT $\mathcal{B}_1, \mathcal{B}_2$ such that

$$\text{Adv}_0(\mathcal{A}) \leq \text{Adv}_1(\mathcal{A}) + 2 \cdot t \cdot \left(\text{Adv}_{(\mathcal{R}, h \lceil \log_\beta q \rceil, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_t), \mathcal{B}_1}^{\text{LWE}}(\lambda) + \text{Adv}_{(\mathcal{R}, n, m, q, \chi_2^*), \mathcal{B}_2}^{\text{LWE}}(\lambda) \right. \\ \left. + \frac{6 \cdot Q \cdot \epsilon}{t(1 - \epsilon)} + 4 \cdot QK \cdot \epsilon + \frac{6 \cdot \epsilon}{1 - \epsilon} \right)$$

Proof. Let $\text{par} = ((\mathcal{R}, n, m, q, Q, \bar{\chi}, \chi, \mathcal{D}_A, \mathcal{D}_t, \beta/2), (t, K, \Xi))$, where

$$\mathcal{D}_A := \mathcal{U}(\mathcal{R}_q^{n \times m}), \quad \text{and} \quad \mathcal{D}_t := \left\{ \tilde{\mathbf{t}} : \mathbf{t} \leftarrow \mathcal{R}_q^h, \tilde{\mathbf{t}} := \mathbf{G}_{\beta, q}^{-1}(\mathbf{t}) \right\}.$$

We will show that for any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_0(\mathcal{A}) \leq \text{Adv}_1(\mathcal{A}) + 2 \cdot \text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda),$$

where $\text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda)$ is defined in Lemma 21. The claimed result will follow by plugging the upper-bound for $\text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda)$ from Lemma 21, and by recalling that the parameter $\mathcal{D}_A$ is omitted from the parameters of the LWE instance when $\mathcal{D}_A$ is the uniform distribution (see Definition 2).

Suppose $\mathcal{A}$ is a PPT distinguisher for $(\text{Hyb}_{b,0}, \text{Hyb}_{b,1})$. We construct a reduction $\mathcal{B}$ distinguishing $\text{Real}_{\text{par}, \mathcal{B}}$ and $\text{Ideal}_{\text{par}, \mathcal{B}}$. The reduction $\mathcal{B}$ works as follows:

- It receives $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ from the challenger. It samples $\text{pp}_{\text{FHE}} \leftarrow \text{FHE.Setup}(1^\lambda)$, sets $\text{pp} := \text{pp}_{\text{FHE}}$, and passes pp to $\mathcal{A}$, receiving from which $\mathcal{C} \subset_{<t} [K]$.
- It samples $(\text{pk}_{\text{FHE}}, \text{sk}_{\text{FHE}}) \leftarrow \text{FHE.KGen}(\text{pp}_{\text{FHE}})$, parses $\text{sk}_{\text{FHE}} = \hat{\mathbf{s}} \in \mathcal{R}_q^{\hat{n}}$, and sets $\mathbf{z}^T := (\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \mathbf{G}_{\beta, q} \bmod q \in \mathcal{R}_q^m$. It forwards $(\mathcal{C}, \mathbf{z})$ to the challenger, receiving from which $(\mathbf{b}, \mathbf{S}_C = (\text{sk}_k)_{k \in \mathcal{C}})$.
- It sets $\text{pk} := (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$ and $(\text{sk}_k)_{k \in \mathcal{C}} := (\text{sk}_k)_{k \in \mathcal{C}}$, and sends $(\text{pk}, (\text{sk}_k)_{k \in \mathcal{C}})$ to $\mathcal{A}$.
- It internally maintains lists $L_{\text{Query}}, L_{\text{Share}}, L_{\text{Party}}$, initially empty. Define two subroutines **Pre** and **Tag** below, which will be used by the reduction to answer oracle queries, and where **Tag** invokes the experiment's oracle **GenO**:

$\text{Pre}(\text{ctxt}_{\text{FHE}})$	$\text{Tag}(\text{id}, \text{ctxt}_{\text{FHE}})$
$(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$	$\tilde{\mathbf{c}} := \text{Pre}(\text{ctxt}_{\text{FHE}})$
return $\mathbf{G}_{\beta, q}^{-1}(\hat{\mathbf{c}}_0)$	$\mathbf{x} \leftarrow \text{GenO}(\text{id}, \tilde{\mathbf{c}})$
	parse $\begin{pmatrix} \tilde{\mathbf{c}} \\ \tilde{\mathbf{t}} \end{pmatrix} \leftarrow \mathbf{x}$
	$\mathbf{t} := \mathbf{G}_{\beta, q} \cdot \tilde{\mathbf{t}} \bmod q$
	return $\text{ctxt} := (\text{ctxt}_{\text{FHE}}, \mathbf{t})$

The reduction answers the oracle queries of $\mathcal{A}$ as follows:

- **EncO**(id, μ): If $L_{\text{Query}}[\text{id}] = \emptyset$, it computes

$$\text{ctxt}_{\text{FHE}} \leftarrow \text{FHE.Enc}(\text{pk}_{\text{FHE}}, \mu) \quad \text{and} \quad \text{ctxt} \leftarrow \text{Tag}(\text{id}, \text{ctxt}_{\text{FHE}}), \quad \text{and sets } L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu, \mu).$$

It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- **ChalO**(id, μ_0, μ_1): If $L_{\text{Query}}[\text{id}] = \emptyset$, it computes

$$\text{ctxt}_{\text{FHE}} \leftarrow \text{FHE.Enc}(\text{pk}_{\text{FHE}}, \mu_b) \quad \text{and} \quad \text{ctxt} \leftarrow \text{Tag}(\text{id}, \text{ctxt}_{\text{FHE}}), \quad \text{and sets } L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu_0, \mu_1).$$

It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- **EvalO**($\text{id}, f, (\text{id}_i)_{i \in I}$): If $L_{\text{Query}}[\text{id}] = \emptyset$, then:
 - * For each $i \in I$, it returns $\perp$ if $L_{\text{Query}}[\text{id}_i] = \emptyset$, and otherwise parses $(\text{ctxt}_i, \mu_{0,i}, \mu_{1,i}) \leftarrow L_{\text{Query}}[\text{id}_i]$ and $(\text{ctxt}_{\text{FHE}, i}, \mathbf{t}_i) \leftarrow \text{ctxt}_i$.
 - * It computes

$$\text{ctxt}'_{\text{FHE}} \leftarrow \text{Eval}(\text{pk}, f, (\text{ctxt}_{\text{FHE}, i})_{i \in I}) \quad \text{and} \quad \text{ctxt} \leftarrow \text{Tag}(\text{id}, \text{ctxt}'_{\text{FHE}}),$$

$$\text{and sets } L_{\text{Query}}[\text{id}] := (\text{ctxt}, f((\mu_{0,i})_{i \in I}), f((\mu_{1,i})_{i \in I})).$$

It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- **ParDecO**(id, k): If $L_{\text{Query}}[\text{id}] = \emptyset$, it returns $\perp$. Otherwise:
 - * It adds $\{k\}$ to $L_{\text{Party}}[\text{id}]$, and parses $(\text{ctxt}, \mu_0, \mu_1) \leftarrow L_{\text{Query}}[\text{id}]$ and $(\text{ctxt}_{\text{FHE}}, \mathbf{t}) \leftarrow \text{ctxt}$.
 - * If $\mu_0 \neq \mu_1$ and $|L_{\text{Party}}[\text{id}] \cup \mathcal{C}| \geq t$, it returns $\perp$.
 - * Otherwise, if $L_{\text{Share}}[\text{id}, k] = \emptyset$, it queries

$$d_k \leftarrow \text{ShareO}(\text{id}, k) \quad \text{and sets } L_{\text{Share}}[\text{id}, k] := d_k.$$

It returns $L_{\text{Share}}[\text{id}, k]$ to $\mathcal{A}$.

- Eventually receive b' from $\mathcal{A}$, return b' .

Below, we show that $\mathcal{B}$ perfectly simulates the view of either $\text{Hyb}_{b,0}$ or $\text{Hyb}_{b,1}$ to $\mathcal{A}$, if it is interacting either with the $\text{Real}_{\text{par}, \mathcal{B}}$ or the $\text{Ideal}_{\text{par}, \mathcal{B}}$ experiment.

If $\mathcal{B}$ is interacting with the $\text{Real}_{\text{par}, \mathcal{B}}$ experiment, then

$$\mathbf{b}^T = \mathbf{r}^T \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q$$

$$\mathbf{x} = \begin{pmatrix} \tilde{\mathbf{c}} \\ \tilde{\mathbf{t}} \end{pmatrix} \quad \text{for } \tilde{\mathbf{c}} = \mathbf{G}_{\beta, q}^{-1}(\hat{\mathbf{c}}_0) \quad \text{and } \tilde{\mathbf{t}} \leftarrow_{\$} \mathcal{D}_{\mathbf{t}}$$

$$\mathbf{d}^T = (\text{pd}_k)_{k \in [K]} = \xi \cdot \mathbf{VRAx} + \mathbf{e}_{\mathbf{x}} \bmod q$$

for $\mathbf{e}_x \leftarrow \$ D_{\mathcal{R}, \chi}^K$. Using that $\mathcal{D}_t = \left\{ \tilde{\mathbf{t}} : \mathbf{t} \leftarrow \$ \mathcal{R}_q^h, \tilde{\mathbf{t}} := \mathbf{G}_{\beta, q}^{-1}(\mathbf{t}) \right\}$ and that $(\mathbf{s}_k)_{k \in [K]} = \xi \cdot \mathbf{V}\mathbf{R} \bmod q$, one obtains that

$$(\mathbf{pd}_k)_{k \in [K]} = \underbrace{\xi \cdot \mathbf{V} \cdot \mathbf{R}}_{(\mathbf{s}_k)_{k \in [K]}} \cdot \underbrace{\mathbf{A} \cdot \mathbf{x}}_{\mathbf{c}_0} + \underbrace{\mathbf{e}_x}_{(e_k)_{k \in [K]}} = (\mathbf{s}_k^T \cdot \mathbf{c}_0 + e_k)_{k \in [K]}$$

Therefore, the view of $\mathcal{A}$ is identical to $\text{Hyb}_{b,0}$.

If $\mathcal{B}$ is interacting with the $\text{Ideal}_{\text{par}, \mathcal{B}}$ experiment, then

$$\begin{aligned} \mathbf{b} &\leftarrow \$ \mathcal{R}_q^m \\ \mathbf{x} &= \begin{pmatrix} \tilde{\mathbf{c}} \\ \tilde{\mathbf{t}} \end{pmatrix} \quad \text{for } \tilde{\mathbf{c}} = \mathbf{G}_{\beta, q}^{-1}(\hat{\mathbf{c}}_0) \text{ and } \tilde{\mathbf{t}} \leftarrow \$ \mathcal{D}_t \\ \mathbf{d}^T &= (\mathbf{pd}_k)_{k \in [K]} = \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_{\mathcal{C}} \\ \mathbf{I} \\ \xi \mathbf{I}_{\mathcal{C}} \end{bmatrix} \begin{bmatrix} (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_{\mathbf{x}} \bmod q \end{aligned}$$

Using as above that $\mathcal{D}_t = \left\{ \tilde{\mathbf{t}} : \mathbf{t} \leftarrow \$ \mathcal{R}_q^h, \tilde{\mathbf{t}} := \mathbf{G}_{\beta, q}^{-1}(\mathbf{t}) \right\}$, we conclude that the view of $\mathcal{A}$ is identical to $\text{Hyb}_{b,1}$. By the triangle inequality, it follows that

$$\begin{aligned} \text{Adv}_0(\mathcal{A}) &= |\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1]| \\ &= \left| \begin{aligned} &\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] \\ &+ \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1] \end{aligned} \right| \\ &\leq |\Pr[\text{Hyb}_{0,0,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1]| \\ &\quad + |\Pr[\text{Hyb}_{0,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1]| \\ &\quad + |\Pr[\text{Hyb}_{1,1,\mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Hyb}_{1,0,\mathcal{A}}(1^\lambda) = 1]| \\ &\leq \text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda) + \text{Adv}_1(\mathcal{A}) + \text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda) \\ &= \text{Adv}_1(\mathcal{A}) + 2 \cdot \text{Adv}_{\text{par}, \mathcal{B}}^{\text{RealIdeal}}(\lambda), \end{aligned}$$

as wanted. As mentioned at the beginning of the proof, plugging the upper-bound for $\text{Adv}_{\text{par}, \mathcal{A}}^{\text{RealIdeal}}(\lambda)$ from Lemma 21 yields the claimed results. $\square$

Lemma 13. *Adopt the notations in Theorem 3. We have $\text{Adv}_1(\mathcal{A}) = \text{Adv}_2(\mathcal{A})$.*

Proof. Fix any id such that $L_{\text{Query}}[\text{id}] = (\text{ctxt}, \mu_0, \mu_1)$ and $\mu_0 \neq \mu_1$. Denote by $\bar{\mathcal{H}} \subseteq \mathcal{H} = [K] \setminus \mathcal{C}$ the set of honest parties for which an adversary has queried its partial decryptions. By design of the security experiment, it must hold that $|\bar{\mathcal{H}}| < t - |\mathcal{C}|$.

Now recall that, in $\text{Hyb}_{b,1}$, the set of partial decryptions for the parties $\bar{\mathcal{H}}$ is defined by Eq. (21), which is

$$\begin{aligned} (\mathbf{pd}_k)_{k \in \bar{\mathcal{H}}} &= [\mathbf{V}_{\bar{\mathcal{H}}} \cdot (\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2)] \begin{bmatrix} (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_{\mathbf{x}} \bmod q \\ &= (\mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{u}_0 \cdot (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} + \mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_1 \cdot \mathbf{w}_{\mathbf{x}} + \mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_2 \cdot \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x}) + \mathbf{e}_{\mathbf{x}} \bmod q, \end{aligned}$$

where we recall $\mathbf{w}_{\mathbf{x}}$ is a random vector sampled by the experiment and $\mathbf{x} = \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \tilde{\mathbf{t}} \end{pmatrix}$. By [CLW25, Lemma 9, Eprint], the columns of $\mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_1$ generate $\mathcal{R}_q^{|\bar{\mathcal{H}}|}$. Therefore, since $\mathbf{w}_{\mathbf{x}}$ is uniformly random, so is $\mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_1 \cdot \mathbf{w}_{\mathbf{x}} \bmod q$. That is,

$$\mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_1 \cdot \mathbf{w}_{\mathbf{x}} \bmod q \sim \mathcal{U}(\mathcal{R}_q^{|\bar{\mathcal{H}}|}),$$

and therefore we get

$$(\mathbf{pd}_k)_{k \in \bar{\mathcal{H}}} \equiv \left(\mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{u}_0 \cdot (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} + \mathcal{U}(\mathcal{R}_q^{|\bar{\mathcal{H}}|}) + \mathbf{V}_{\bar{\mathcal{H}}} \cdot \mathbf{U}_2 \cdot \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \right) + \mathbf{e}_{\mathbf{x}} \bmod q$$

$$\equiv \mathcal{U}(\mathcal{R}_q^{|\bar{\mathcal{H}}|}).$$

In other words, the set of partial decryptions $(\text{pd}_k)_{k \in \bar{\mathcal{H}}}$ for id is uniformly random, identical to that in $\text{Hyb}_{b,2}$. Therefore, $\text{Adv}_1(\mathcal{A}) = \text{Adv}_2(\mathcal{A})$. $\square$

Lemma 14. *Adopt the notations in Theorem 3. We have $\text{Adv}_2(\mathcal{A}) = \text{Adv}_3(\mathcal{A})$.*

Proof. Fix any id such that $L_{\text{Query}}[\text{id}] = (\text{ctxt}, \mu_0, \mu_1)$, $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$, and $\mu_0 = \mu_1$. Denote $\mu := \mu_0 = \mu_1$ and let

$$\hat{e} := \hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 - \xi^{-2} \left\lfloor \frac{q}{p} \right\rfloor \mu \bmod q \quad (25)$$

Notice that

$$\begin{aligned} \mathbf{z}^T \mathbf{x} \bmod q &= ((\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \mathbf{G}_{\beta,q}) \cdot \mathbf{G}_{\beta,q}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \bmod q \\ &= \left((\hat{\mathbf{s}}^T, \mathbf{0}_h^T) \cdot \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix} \right) \bmod q \\ &= (\hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0) \bmod q \\ &= \left(\hat{c}_1 - \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) + q \cdot h'' \right) \bmod q \quad // \text{ By Eq. (25)} \\ &= \left(\hat{c}_1 - \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) \right) \bmod q. \end{aligned}$$

Hence,

$$\begin{aligned} &\left[\begin{array}{c|c|c} \mathbf{V}_{\mathcal{H}} \cdot (\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2) & & \xi \mathbf{I}_C \end{array} \right] \left[\begin{array}{c} (\mathbf{b} - \mathbf{e})^T \mathbf{x} - \mathbf{z}^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{array} \right] + \mathbf{e}_{\mathbf{x}} \bmod q \\ &= \left[\begin{array}{c|c|c} \mathbf{V}_{\mathcal{H}} \cdot (\mathbf{u}_0, \mathbf{U}_1, \mathbf{U}_2) & & \xi \mathbf{I}_C \end{array} \right] \left[\begin{array}{c} (\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) \right) \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{array} \right] + \mathbf{e}_{\mathbf{x}} \bmod q \\ &= \left[\begin{array}{c} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{array} \right] \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) \right) \right] + \left[\begin{array}{c|c} \mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2) & \xi \mathbf{I}_C \end{array} \right] \left[\begin{array}{c} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{array} \right] + \mathbf{e}_{\mathbf{x}} \bmod q, \end{aligned}$$

that is, the partial decryptions for id in $\text{Hyb}_{b,2}$ and $\text{Hyb}_{b,3}$ are identical. Therefore, $\text{Adv}_2(\mathcal{A}) = \text{Adv}_3(\mathcal{A})$. $\square$

Lemma 15. *Adopt the notations in Theorem 3. Then, for any PPT $\mathcal{A}$ there exists PPT $\mathcal{B}_3$ such that $\text{Adv}_3(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}_3, (\theta, K, \chi)}^{\text{IND-CPA-wL}}(\lambda)$.*

Proof. Suppose $\mathcal{A}$ is a PPT distinguisher for $(\text{Hyb}_{0,3}, \text{Hyb}_{1,3})$. We construct a reduction $\mathcal{B}_3$ against the IND-CPA under leakage security of FHE, which proceeds as follows:

- It receives $(\text{pp}_{\text{FHE}}, \text{pk}_{\text{FHE}})$ from the IND-CPA-under-leakage challenger, parses $\text{pp} = \text{pp}_{\text{FHE}}$, and passes pp to $\mathcal{A}$, receiving from which $\mathcal{C} \subset_{<t} [K]$.
- It samples $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, $\mathbf{R} \leftarrow \mathcal{R}_q^{t \times n}$, $\mathbf{b} \leftarrow \mathcal{R}_q^m$, and $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^m$, sets $(\text{sk}_k)_{k \in [K]} := \xi \cdot \mathbf{V} \cdot \mathbf{R} \bmod q$ and $\text{pk} := (\text{pk}_{\text{FHE}}, \mathbf{A}, \mathbf{b})$, and sends $(\text{pk}, (\text{sk}_k)_{k \in \mathcal{C}})$ to $\mathcal{A}$.
- It internally maintains lists $L_{\text{Query}}, L_{\text{Share}}, L_{\text{Party}}$, initially empty. It answers $\mathcal{A}$'s queries to $\text{EncO}, \text{ChalO}, \text{EvalO}$ by forwarding to its own $\text{EncO}, \text{ChalO}, \text{EvalO}$ from the IND-CPA-under-leakage challenger, and appending a freshly sampled $\mathbf{t}$ itself. In more detail, it answers the oracle queries as follows:

- $\text{EncO}(\text{id}, \mu)$: If $L_{\text{Query}}[\text{id}] = \emptyset$, then

$$\text{query}(\text{ctxt}_{\text{FHE}}, \mu, \mu) \leftarrow \text{EncO}(\text{id}, \mu), \quad \text{sample } \mathbf{t} \leftarrow \mathcal{R}_q^h, \quad \text{set } \text{ctxt} := (\text{ctxt}_{\text{FHE}}, \mathbf{t})$$

and $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu, \mu)$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- $\text{ChalO}(\text{id}, \mu_0, \mu_1)$: If $L_{\text{Query}}[\text{id}] = \emptyset$, then

$$\text{query}(\text{ctxt}_{\text{FHE}}, \mu_0, \mu_1) \leftarrow \text{ChalO}(\text{id}, \mu_0, \mu_1), \quad \text{sample } \mathbf{t} \leftarrow \mathcal{R}_q^h, \quad \text{set } \text{ctxt} := (\text{ctxt}_{\text{FHE}}, \mathbf{t})$$

and $L_{\text{Query}}[\text{id}] := (\text{ctxt}, \mu_0, \mu_1)$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- $\text{EvalO}(\text{id}, f, (\text{id}_i)_{i \in I})$: If $L_{\text{Query}}[\text{id}] = \emptyset$, it parses $(\text{ctxt}_i, \mu_{0,i}, \mu_{1,i}) \leftarrow L_{\text{Query}}[\text{id}_i]$ for each $i \in I$, then

$$\text{query}(\text{ctxt}'_{\text{FHE}}, \mu'_0, \mu'_1) \leftarrow \text{EvalO}(\text{id}, f, (\text{id}_i)_{i \in I}), \quad \text{sample } \mathbf{t} \leftarrow \mathcal{R}_q^h, \quad \text{set } \text{ctxt}' := (\text{ctxt}'_{\text{FHE}}, \mathbf{t})$$

and $L_{\text{Query}}[\text{id}] := (\text{ctxt}', \mu'_0, \mu'_1)$. It returns $L_{\text{Query}}[\text{id}]$ to $\mathcal{A}$.

- To answer $\mathcal{A}$'s oracle queries to $\text{ParDecO}(\text{id}, k)$: If $L_{\text{Query}}[\text{id}] = \emptyset$, the reduction returns $\perp$. Otherwise it adds $\{k\}$ to $L_{\text{Party}}[\text{id}]$ and parses $(\text{ctxt}, \mu_0, \mu_1) \leftarrow L_{\text{Query}}[\text{id}]$. If $\mu_0 \neq \mu_1$ and $|L_{\text{Party}}[\text{id}] \cup \mathcal{C}| \geq t$, it returns $\perp$. Otherwise, if $L_{\text{Share}}[\text{id}] = \emptyset$, it computes $(\text{pd}_k)_{k \in [K]}$ by distinguishing two cases:

- If $\mu_0 = \mu_1$: It simulates $(\text{pd}_k)_{k \in [K]}$ using the leakO answer from the challenger. Specifically,

$$\text{set } \mathbf{u} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix}, \quad \text{query } \text{leak} \leftarrow \text{LeakO}(\text{id}, \mathbf{u}),$$

where $\mathbf{u}_0$ was defined in Eq. (22). Then, it parses $(\text{ctxt}_{\text{FHE}}, \mathbf{t}) \leftarrow \text{ctxt}$, computes $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$, samples $\mathbf{w}_{\mathbf{x}} \leftarrow \mathcal{R}_q^{t-|\mathcal{C}|-1}$, sets $\mu := \mu_0 = \mu_1$ and $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$, and it sets

$$(\text{pd}_k)_{k \in [K]} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{\mathbf{c}}_1 - \left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu \right) \right] + \text{leak} + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_{\mathcal{C}}} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{bmatrix} \bmod q.$$

- If $\mu_0 \neq \mu_1$: It parses $(\text{ctxt}_{\text{FHE}}, \mathbf{t}) \leftarrow \text{ctxt}$, computes $(\hat{\mathbf{c}}_0, \hat{\mathbf{c}}_1) \leftarrow \text{Preproc}(\text{ctxt}_{\text{FHE}})$ and $\mathbf{x} := \mathbf{G}^{-1} \begin{pmatrix} \hat{\mathbf{c}}_0 \\ \mathbf{t} \end{pmatrix}$,

and samples $\mathbf{e}_{\mathbf{x}} \leftarrow D_{\mathcal{R}, \chi}^{|\mathcal{C}|}$. It then samples the honest coordinates uniformly at random and computes the corrupt coordinates as real partial decryptions, that is

$$(\text{pd}_k)_{k \in \mathcal{H}} \leftarrow \mathcal{R}_q^{|\mathcal{H}|} \quad \text{and} \quad (\text{pd}_k)_{k \in \mathcal{C}} := \xi \cdot \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} + \mathbf{e}_{\mathbf{x}} \bmod q.$$

In either case, it sets $L_{\text{Share}}[\text{id}] := (\text{pd}_k)_{k \in [K]}$. It returns $\text{pd}_k \leftarrow L_{\text{Share}}[\text{id}]$ to $\mathcal{A}$.

- Eventually, it receives b' from $\mathcal{A}$ and returns b' .

Below we analyse the reduction's simulation. Recall that $\mathcal{C}^*$ satisfies $(\{0\} \cup \mathcal{C}) \subseteq \mathcal{C}^* \subseteq [0, K]$. Notice that

$$\|\mathbf{u}\| = \left\| \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \right\| = \|\mathbf{V}_{\mathcal{H}} \cdot \xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \cdot \mathbf{v}_0\| \leq \theta(t),$$

where the second equality is by definition of $\mathbf{u}_0 = \mathbf{T}_{\mathcal{C}} \cdot \mathbf{v}_0 = \xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \cdot \mathbf{v}_0$, $\theta(t)$ is one of the quality measurements of the subtractive set Ξ (Definition 1, Eq. (4)), and the inequality holds by taking $T^* = \mathcal{C}^* \setminus \{0\}$ in Eq. (4). Here $\mathbf{V}_{\mathcal{H}}$ carries more rows than $\mathbf{V}_{\Xi \setminus T^*}$ when $|\mathcal{C}| < t - 1$, but the additional rows, indexed by the filler honest parties $(\mathcal{C}^* \setminus \{0\}) \setminus \mathcal{C}$, vanish on $\mathbf{u}_0$ since $\mathbf{V}_{\mathcal{C}^*} \cdot \mathbf{u}_0 = \xi \cdot \mathbf{v}_0$, so the two give the same norm. Therefore, the queries made by $\mathcal{B}_3$ to the oracle LeakO are valid.

Below, we show that $\mathcal{B}_3$ perfectly simulates the view of either $\text{Hyb}_{0,3}$ or $\text{Hyb}_{1,3}$ to $\mathcal{A}$, if it is interacting either with the IND-CPA-with- $(\theta(t), K, \chi)$ -leakage experiment 0 or 1.

Consider the $\mu_0 = \mu_1$ case. By definition of the IND-CPA-with- $(\theta(t), K, \chi)$ -leakage security experiment (Fig. 2), one has

$$\text{leak} = \mathbf{u} \cdot \hat{\mathbf{e}} + \mathbf{f} = \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \cdot \hat{\mathbf{e}} + \mathbf{f}.$$

where

$$\hat{\mathbf{e}} := \hat{\mathbf{c}}_1 - \hat{\mathbf{s}}^T \cdot \hat{\mathbf{c}}_0 - \xi^{-2} \left\lfloor \frac{q}{p} \right\rfloor \mu \bmod q$$

as also defined in Eq. (24). It follows that

$$(\text{pd}_k)_{k \in [K]} = \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{\mathbf{c}}_1 - \left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu \right) \right] + \text{leak} + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_{\mathcal{C}}} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{A} \mathbf{x} \end{bmatrix} \bmod q$$

$$\begin{aligned}
&= \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu \right) \right] \\
&\quad + \left(\begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \cdot \hat{e} + \mathbf{f} \right) + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_C} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} \bmod q \\
&= \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu - \hat{e} \right) \right] + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_C} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{f} \bmod q \\
&= \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \mathbf{u}_0 \\ \mathbf{0} \end{bmatrix} \left[(\mathbf{b} - \mathbf{e})^T \mathbf{x} - \left(\hat{c}_1 - \left(\left\lfloor \frac{q}{p} \right\rfloor \cdot \xi^{-2} \cdot \mu + \hat{e} \right) \right) \right] + \left[\frac{\mathbf{V}_{\mathcal{H}}(\mathbf{U}_1, \mathbf{U}_2)}{\xi \mathbf{I}_C} \right] \begin{bmatrix} \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_{\mathbf{x}} \bmod q
\end{aligned}$$

where in the last line we have used that $\mathbf{f}$ and $\mathbf{e}_{\mathbf{x}}$ are identically distributed. This matches the distribution of $(\mathbf{pd}_k)_{k \in [K]}$ defined in Eq. (23).

For the $\mu_0 \neq \mu_1$ case, recall that the distribution of $(\mathbf{pd}_k)_{k \in [K]}$ is identical in $\text{Hyb}_{b,3}$ and $\text{Hyb}_{b,2}$, where in the latter $(\mathbf{pd}_k)_{k \in \mathcal{H}}$ are uniformly random and $(\mathbf{pd}_k)_{k \in \mathcal{C}}$ are real partial decryptions. This is exactly the distribution of $(\mathbf{pd}_k)_{k \in [K]}$ sampled by the reduction.

It follows that the ParDecO is correctly simulated in both cases.

If $\mathcal{B}_3$ is interacting with the IND-CPA-with- $(\theta(t), K, \chi)$ -leakage experiment 0, then queries to $\text{ChalO}(\text{id}, \mu_0, \mu_1)$ return $\text{ctxt}_{\text{FHE}} \leftarrow \text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, \mu_0)$ and therefore $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu_0)$. Therefore, the view of $\mathcal{A}$ is identical to $\text{Hyb}_{0,3}$.

If $\mathcal{B}_3$ is interacting with the IND-CPA-with- $(\theta(t), K, \chi)$ -leakage experiment 1, then queries to $\text{ChalO}(\text{id}, \mu_0, \mu_1)$ return $\text{ctxt}_{\text{FHE}} \leftarrow \text{FHE}.\text{Enc}(\text{pk}_{\text{FHE}}, \mu_1)$ and therefore $\text{ctxt} \leftarrow \text{Enc}(\text{pk}, \mu_1)$. Therefore, the view of $\mathcal{A}$ is identical to $\text{Hyb}_{1,3}$.

Therefore, $\text{Adv}_3(\mathcal{A}) \leq \text{Adv}_{\text{FHE}, \mathcal{B}_3, (\theta, K, \chi)}^{\text{IND-CPA-wL}}(\lambda)$, as claimed. $\square$

G A Utility Lemma and Intermediate Assumption

In Appendix G.1, we state the Leaky LWE assumption and provide lemmas on reducing it from standard LWE. The Leaky LWE assumption serves as an intermediate interface which simplifies our proof steps. Using this, in Appendix G.2, we state and prove a utility lemma (Lemma 21), which concerns the computational indistinguishability of a pair of real and ideal experiments. This utility lemma is invoked in the proof of Theorem 3, specifically in Lemma 12, in Appendix F. Additional results assisting this utility lemma are in Appendices G.3 and G.4.

G.1 Intermediate lattice assumption

The Leaky LWE assumption [LSW25] supports leakages on the LWE errors, i.e. the adversary declares the set of leakage functions and sees the noisy evaluated LWE errors, conditioned on that the leakage functions are norm-bounded. We state both the semi- and fully-adaptive versions below.

Definition 8 (LLWE [LSW25], AdpLLWE). *Let $\text{par} = (\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}, \mathcal{D}_{\mathbf{A}})$ be parametrised by λ , where $\mathcal{D}_{\mathbf{A}} \sim \mathcal{R}_q^{n \times m}$ is a distribution. The semi-adaptive Leaky LWE assumption LLWE_{par} (resp. adaptive Leaky LWE assumption $\text{AdpLLWE}_{\text{par}}$) states that for any PPT adversary $\mathcal{A}$, for the experiments Exp-LLWE^b given in Fig. 4, it holds that*

$$\text{Adv}_{\text{par}, \mathcal{A}}^{\left[\begin{smallmatrix} \text{LLWE} \\ \text{AdpLLWE} \end{smallmatrix} \right]} := \left| \Pr[\text{Exp-LLWE}_{\text{par}, \mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp-LLWE}_{\text{par}, \mathcal{A}}^1(1^\lambda) = 1] \right| \leq \text{negl}(\lambda).$$

The parameter $\mathcal{D}_{\mathbf{A}}$ is suppressed if it is the uniform distribution over $\mathcal{R}_q^{n \times m}$.

[LSW25], resp. [HKM⁺26]¹⁷, showed that the semi-adaptive, resp. full-adaptive, Leaky LWE is implied by standard LWE under a polynomial-size modulus q . We recall both results below, with that of [HKM⁺26] adapted to the formulation and norm convention in this work. For completeness, we also provide the adapted proof.

¹⁷ More specifically, [HKM⁺26] showed that the Adaptive Hint-MLWE assumption is implied by Hint-MLWE [KLSS23], the latter further implied by LWE. Leaky LWE is a variant of Hint-MLWE, and the reduction of [HKM⁺26] equally applies to Leaky LWE.

Exp-LLWE $_{\text{par}, \mathcal{A}}^b(1^\lambda)$	LeakO($\mathbf{z}$)
$\mathbf{A} \leftarrow \mathcal{D}_A$	assert $i < Q$
$\mathbf{r} \leftarrow \mathcal{R}_q^n; \bar{\mathbf{e}} \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^m$	$i := i + 1$
if $b = 0$ then $\mathbf{b}^\top := \mathbf{r}^\top \mathbf{A} + \bar{\mathbf{e}}^\top \bmod q$	assert $\mathbf{z} \in \mathcal{L} \subseteq \mathcal{R}^m$
if $b = 1$ then $\mathbf{b} \leftarrow \mathcal{R}_q^m$	$e \leftarrow \mathcal{D}_{\mathcal{R}, \chi}; \ell := \bar{\mathbf{e}}^\top \cdot \mathbf{z} + e$
$\mathbf{Z} \leftarrow \mathcal{A}(\mathbf{A})$ assert $\mathbf{Z} \in \mathcal{L} \subseteq \mathcal{R}^{m \times Q}$ $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^Q; \mathbf{l}^\top := \bar{\mathbf{e}}^\top \cdot \mathbf{Z} + \mathbf{e}^\top$ return $\mathcal{A}(\mathbf{b}, \mathbf{l})$	return ℓ
$i := 0$ return $\mathcal{A}^{\text{LeakO}}(\mathbf{A}, \mathbf{b})$	

Fig. 4: Experiments for Leaky LWE (Definition 8), [dashed box] for semi-adaptive, [solid box] for adaptive.

Lemma 16 (Special case of [LSW25, Theorem 3]). Let $\mathcal{R}, n, m, Q, q, \epsilon, \alpha, \bar{\chi}, \chi, \chi^*, \mathcal{L}, \mathcal{D}$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$, $n, m, Q, q \in \mathbb{N}$, $\mathcal{D} \sim \mathcal{R}_q^{n \times m}$, $0 < \epsilon \leq 1/2$, $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$, $\bar{\chi} > 0$, $\chi > 0$, and $\mathcal{L} = \left\{ \mathbf{Z} \in \mathcal{R}^{m \times Q} : \|\mathbf{Z}^\top\|_{2,2} \leq \alpha \right\}$, where $\alpha = \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}$.
Denote

$$\text{par} = (\mathcal{R}, n, m, q, \chi^*, \mathcal{D}), \quad \text{par}_L = (\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}, \mathcal{D}).$$

$\forall$ PPT $\mathcal{A}$, $\exists$ PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}_L, \mathcal{A}}^{\text{LLWE}}(\lambda) \leq \text{Adv}_{\text{par}, \mathcal{B}}^{\text{LWE}}(\lambda) + 6 \cdot \epsilon / (1 - \epsilon).$$

In particular, if $\epsilon \in \text{negl}(\lambda)$ and the LWE_{par} assumption holds, then the $\text{LLWE}_{\text{par}_L}$ assumption holds.

The following is a special case of the Gaussian convolution lemma in [GMPW20], needed for Lemma 18.

Lemma 17 (Special case of [GMPW20, Theorem 3.1 (eprint)]). Let $\mathcal{R} = \mathbb{Z}[\zeta]$ have ring degree φ and coefficient embedding ψ , let $m \in \mathbb{N}$, $0 < \epsilon \leq 1/2$, and $s, \kappa > 0$ with $s > \eta_\epsilon(\mathcal{R}^m)$. Let $\mathbf{z} \in \mathcal{R}^m$ be arbitrary satisfying

$$\|\Psi(\mathbf{z}^\top)\|_{2,2} \leq \kappa \cdot \sqrt{1 - \eta_\epsilon(\mathcal{R}^m)^2 / s^2} \quad (26)$$

and let $\Sigma_{\mathbf{z}} := s^2(\kappa^2 \mathbf{I}_\varphi - \Psi(\mathbf{z}^\top)\Psi(\mathbf{z}^\top)^\top)$. Then the statistical distance between

$$\{\mathbf{z}^\top \mathbf{y} + f \mid \mathbf{y} \leftarrow \mathcal{D}_{\mathcal{R}, s}^m, f \leftarrow \mathcal{D}_{\mathcal{R}, \sqrt{\Sigma_{\mathbf{z}}}}\} \quad \text{and} \quad \mathcal{D}_{\mathcal{R}, \kappa s}$$

is at most $\epsilon / (1 - \epsilon) \leq 2\epsilon$.

Proof. In this proof we use the shorthands

$$\mathbf{M} := \Psi(\mathbf{z}^\top), \quad \mu := \|\mathbf{M}\|_{2,2}, \quad \Sigma := \Sigma_{\mathbf{z}}, \quad \eta := \eta_\epsilon(\mathcal{R}^m) = \eta_\epsilon(\mathbb{Z}^{\varphi m}).$$

The bijection ψ preserves statistical distances and maps the distributions

$$\mathcal{D}_{\mathcal{R}, s}^m, \mathcal{D}_{\mathcal{R}, \sqrt{\Sigma}}, \mathcal{D}_{\mathcal{R}, \kappa s}, \mathbf{z}^\top \mathbf{y} \quad \text{to} \quad \mathcal{D}_{\mathbb{Z}^{\varphi m}, s}, \mathcal{D}_{\mathbb{Z}^{\varphi}, \sqrt{\Sigma}}, \mathcal{D}_{\mathbb{Z}^{\varphi}, \kappa s}, \mathbf{M} \cdot \psi(\mathbf{y})$$

respectively, so it suffices to bound the distance between $\mathcal{D} := \{\mathbf{M}\mathbf{y} + \mathbf{f} \mid \mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^{\varphi m}, s}, \mathbf{f} \leftarrow \mathcal{D}_{\mathbb{Z}^{\varphi}, \sqrt{\Sigma}}\}$ and $\mathcal{D}_{\mathbb{Z}^{\varphi}, \kappa s}$.

We apply [GMPW20, Theorem 3.1 (eprint)], for which we express the distribution $\mathcal{D}$ as a linear image of one discrete Gaussian vector. Let $\Lambda := \mathbb{Z}^{\varphi m + \varphi}$, $\mathbf{S} := \text{diag}(s \cdot \mathbf{I}_{\varphi m}, \sqrt{\Sigma})$ and $\mathbf{T} := (\mathbf{M} \mid \mathbf{I}_\varphi) \in \mathbb{Z}^{\varphi \times (\varphi m + \varphi)}$.

Since $\mathbf{S}\mathbf{S}^\top = \text{diag}(s^2\mathbf{I}_{\varphi m}, \Sigma)$ is block diagonal, a sample $\mathbf{x} \leftarrow D_{\Lambda, \mathbf{S}}$ is $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{f} \end{bmatrix}$ with independent $\mathbf{y} \leftarrow D_{\mathbb{Z}^{\varphi m}, s}$ and $\mathbf{f} \leftarrow D_{\mathbb{Z}^\varphi, \sqrt{\Sigma}}$, and $\mathbf{T}\mathbf{x} = \mathbf{M}\mathbf{y} + \mathbf{f}$. Hence $\mathbf{T} \cdot D_{\Lambda, \mathbf{S}}$ is exactly the distribution of $\mathbf{M}\mathbf{y} + \mathbf{f}$ in $\mathcal{D}$, and it remains to check the requirements of [GMPW20, Theorem 3.1 (eprint)]:

- $\mathbf{S}$ has full column rank and $\Lambda \subseteq \text{span}(\mathbf{S}) = \mathbb{R}^{\varphi m + \varphi}$.
- $\ker(\mathbf{T}) = \{(\mathbf{u}, -\mathbf{M}\mathbf{u}) : \mathbf{u} \in \mathbb{R}^{\varphi m}\}$ is spanned by $\Lambda \cap \ker(\mathbf{T}) = \{(\mathbf{u}, -\mathbf{M}\mathbf{u}) : \mathbf{u} \in \mathbb{Z}^{\varphi m}\}$, using $\mathbf{M} \in \mathbb{Z}^{\varphi \times \varphi m}$, i.e. $\ker(\mathbf{T})$ is a Λ -subspace.
- $\eta_\epsilon(\Lambda \cap \ker(\mathbf{T})) \leq \mathbf{S}$. Recall that for matrices $\mathbf{S}' \leq \mathbf{T}'$ means $\mathbf{S}'\mathbf{S}'^\top \preceq \mathbf{T}'\mathbf{T}'^\top$ [GMPW20]. Since $\Lambda \cap \ker(\mathbf{T}) = \mathbf{C}\mathbb{Z}^{\varphi m}$ for $\mathbf{C} := \begin{bmatrix} \mathbf{I}_{\varphi m} \\ -\mathbf{M} \end{bmatrix}$, by [GMPW20, Definition 2.4 (eprint)] the condition is same as

$$\eta_\epsilon(\mathbf{B}\mathbb{Z}^{\varphi m}) \leq 1 \quad \text{where} \quad \mathbf{B} := \mathbf{S}^{-1}\mathbf{C} = \begin{bmatrix} s^{-1}\mathbf{I}_{\varphi m} \\ -\sqrt{\Sigma}^{-1}\mathbf{M} \end{bmatrix}.$$

As $\eta_\epsilon(\mathbb{Z}^{\varphi m}) = \eta$, [GMPW20, Lemma 2.5 (eprint)] gives $\eta_\epsilon(\mathbf{B}\mathbb{Z}^{\varphi m}) \leq \eta\mathbf{B}$, so it suffices to show $\eta\mathbf{B} \leq \mathbf{I}_{\varphi m + \varphi}$, i.e. $\eta^2\mathbf{B}^\top\mathbf{B} \preceq \mathbf{I}_{\varphi m}$. Write $\mathbf{X} := \mathbf{M}^\top\mathbf{M}$, whose eigenvalues lie in $[0, \mu^2]$, with $\mu < \kappa$ by (26). Using $\Sigma^{-1} = s^{-2}(\kappa^2\mathbf{I}_\varphi - \mathbf{M}\mathbf{M}^\top)^{-1}$ and $\mathbf{M}^\top(\kappa^2\mathbf{I}_\varphi - \mathbf{M}\mathbf{M}^\top)^{-1} = (\kappa^2\mathbf{I}_{\varphi m} - \mathbf{X})^{-1}\mathbf{M}^\top$ (the “push-through identity”),

$$\begin{aligned} \mathbf{B}^\top\mathbf{B} &= s^{-2}\mathbf{I}_{\varphi m} + \mathbf{M}^\top\Sigma^{-1}\mathbf{M} = s^{-2}(\mathbf{I}_{\varphi m} + (\kappa^2\mathbf{I}_{\varphi m} - \mathbf{X})^{-1}\mathbf{X}) \\ &= \frac{\kappa^2}{s^2}(\kappa^2\mathbf{I}_{\varphi m} - \mathbf{X})^{-1} \preceq \frac{\kappa^2}{s^2(\kappa^2 - \mu^2)}\mathbf{I}_{\varphi m} \preceq \eta^{-2}\mathbf{I}_{\varphi m}, \end{aligned}$$

where the last step is (26) squared and rearranged.

Therefore $\mathbf{T} \cdot D_{\Lambda, \mathbf{S}}$ and $D_{\mathbf{T}\Lambda, \mathbf{T}\mathbf{S}}$ are within relative error $2\epsilon/(1 - \epsilon)$, hence within statistical distance $\epsilon/(1 - \epsilon) \leq 2\epsilon$ using $\epsilon \leq 1/2$. Finally $\mathbf{T}\Lambda = \mathbb{Z}^\varphi$ and $(\mathbf{T}\mathbf{S})(\mathbf{T}\mathbf{S})^\top = s^2\mathbf{M}\mathbf{M}^\top + \Sigma = \kappa^2s^2\mathbf{I}_\varphi$, so $D_{\mathbf{T}\Lambda, \mathbf{T}\mathbf{S}} = D_{\mathbb{Z}^\varphi, \kappa s}$. $\square$

The next lemma, adapted from [HKM⁺26], shows that fully-adaptive Leaky LWE is implied by the semi-adaptive counterpart.

Lemma 18 (Adapted from proof of [HKM⁺26, Theorem 1]). *Let $\mathcal{R}, n, m, Q, q, \epsilon, \alpha, \kappa, \bar{\chi}, \chi, \mathcal{D}$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ and field expansion factor $\mathfrak{c}_2$ w.r.t. the coefficient ℓ_2 -norm, and $n, m, Q, q \in \mathbb{N}$, $\mathcal{D} \sim \mathcal{R}_q^{n \times m}$, $0 < \epsilon \leq 1/2$, and $\alpha \geq 1$, $\kappa, \bar{\chi}, \chi > 0$ satisfy*

$$\mathfrak{c}_2 \cdot \alpha \leq \kappa \cdot \sqrt{1 - \kappa^2 \cdot \eta_\epsilon(\mathcal{R}^m)^2 / \chi^2}. \quad (27)$$

Define the parameters

$$\text{par}_L = (\mathcal{R}, n, m, mQ, q, \bar{\chi}, \chi/\kappa, \mathcal{L}_L, \mathcal{D}) \quad \text{and} \quad \text{par}_A = (\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}_A, \mathcal{D})$$

where $\mathcal{L}_L = \{\mathbf{Z} \in \mathcal{R}^{m \times mQ} : \|\mathbf{Z}^\top\|_{2,2} \leq \sqrt{Q}\}$ and $\mathcal{L}_A = \{\mathbf{z} \in \mathcal{R}^m : \|\mathbf{z}\|_2 \leq \alpha\}$. $\forall$ PPT $\mathcal{A}$, $\exists$ PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}_A, \mathcal{A}}^{\text{AdpLLWE}}(\lambda) \leq \text{Adv}_{\text{par}_L, \mathcal{B}}^{\text{LLWE}}(\lambda) + 4 \cdot Q \cdot \epsilon.$$

In particular, if $Q \in \text{poly}(\lambda)$, $\epsilon \in \text{negl}(\lambda)$, and the $\text{LLWE}_{\mathcal{R}, n, m, mQ, q, \bar{\chi}, \chi/\kappa, \mathcal{L}_L, \mathcal{D}}$ assumption holds for some κ satisfying Eq. (27), then the $\text{AdpLLWE}_{\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}_A, \mathcal{D}}$ assumption holds.

Proof. Consider the following hybrid experiments, where $\Sigma_{\mathbf{z}} := (\chi/\kappa)^2(\kappa^2\mathbf{I}_\varphi - \Psi(\mathbf{z}^\top)\Psi(\mathbf{z}^\top)^\top)$:

Hyb₀: The fully-adaptive Leaky LWE experiment (Fig. 4, solid box) with challenge bit $b = 0$.

Hyb₁: Almost identical to **Hyb₀**, except that the challenger additionally samples $\mathbf{e}_1, \dots, \mathbf{e}_Q \leftarrow D_{\mathcal{R}, \chi/\kappa}^m$, and it answers the i -th oracle query $\text{LeakO}(\mathbf{z})$ with $(\bar{\mathbf{e}} + \mathbf{e}_i)^\top \cdot \mathbf{z} + f$, where $f \leftarrow D_{\mathcal{R}, \sqrt{\Sigma_{\mathbf{z}}}}$ is sampled freshly and dependent on the query $\mathbf{z}$.

Hyb₂: Almost identical to **Hyb₁**, except that the challenger samples the vector $\mathbf{b} \leftarrow \mathcal{R}_q^m$ uniformly randomly.

Hyb_3 : Almost identical to Hyb_2 , except that challenger undoes the change in Hyb_1 , i.e. each oracle query $\text{LeakO}(\mathbf{z})$ is answered with $\bar{\mathbf{e}}^\top \cdot \mathbf{z} + e$ where $e \leftarrow \$ D_{\mathcal{R}, \chi}$ is a fresh error, as in Hyb_0 .

We have the following:

- By Lemma 19, the advantage in distinguishing Hyb_0 and Hyb_1 is at most $2Q\epsilon$.
- By Lemma 20, the advantage in distinguishing Hyb_1 and Hyb_2 equals $\text{Adv}_{\text{par}_L, \mathcal{B}}^{\text{LLWE}}(\lambda)$ for some PPT $\mathcal{B}$.
- By the same argument as for Hyb_0 and Hyb_1 , the advantage in distinguishing Hyb_2 and Hyb_3 is again at most $2Q\epsilon$.

The proof is completed by noting that Hyb_3 is identical to the fully-adaptive Leaky LWE experiment with challenge bit $b = 1$, and the claimed bound follows from the triangle inequality.

Lemma 19. *For any (computationally unbounded) $\mathcal{A}$, the advantage in distinguishing Hyb_0 and Hyb_1 is at most $2Q\epsilon$.*

Proof. For $i \in \{0, 1, \dots, Q\}$, let $\text{Hyb}_{0,i}$ be identical to Hyb_0 except that the first i oracle queries are answered as in Hyb_1 , so that $\text{Hyb}_{0,0} = \text{Hyb}_0$ and $\text{Hyb}_{0,Q} = \text{Hyb}_1$. Fix any $i \in [Q]$. The answer to the i -th LeakO query $\mathbf{z}$ in $\text{Hyb}_{0,i}$ is

$$\hat{\mathbf{e}}_i^\top \cdot \mathbf{z} + f = \bar{\mathbf{e}}^\top \cdot \mathbf{z} + (\mathbf{e}_i^\top \cdot \mathbf{z} + f).$$

By Lemma 17, applied with $\mathbf{y} = \mathbf{e}_i$ and using $\kappa s = \chi$, the summand $\mathbf{e}_i^\top \cdot \mathbf{z} + f$ is within statistical distance 2ϵ to $D_{\mathcal{R}, \chi}$, which is the distribution of the corresponding error in $\text{Hyb}_{0,i-1}$. As $\text{Hyb}_{0,i-1}$ and $\text{Hyb}_{0,i}$ are otherwise identical, their statistical distance is at most 2ϵ . Summing over $i \in [Q]$ yields the claim.

In the rest we show that the requirements in Lemma 17 are satisfied, so that it can indeed be applied. Let $\mathbf{z} \in \mathcal{L}_A$ be any admissible query and write $\mathbf{z} = (z_1, \dots, z_m)$. For any $\mathbf{x} = (x_1, \dots, x_m) \in \mathcal{K}^m$ we have

$$\|\mathbf{z}^\top \mathbf{x}\|_2 \leq \mathbf{c}_2 \cdot \sum_{j \in [m]} \|z_j\|_2 \cdot \|x_j\|_2 \leq \mathbf{c}_2 \cdot \|\mathbf{z}\|_2 \cdot \|\mathbf{x}\|_2$$

by the triangle inequality and Cauchy–Schwarz, hence

$$\|\mathbf{z}^\top\|_{2,2} = \|\Psi(\mathbf{z}^\top)\|_{2,2} \leq \mathbf{c}_2 \cdot \|\mathbf{z}\|_2 \leq \mathbf{c}_2 \cdot \alpha.$$

The above together with Eq. (27) imply Eq. (26) for $s = \chi/\kappa$, i.e.

$$\|\Psi(\mathbf{z}^\top)\|_{2,2} \leq \kappa \cdot \sqrt{1 - \kappa^2 \cdot \eta_\epsilon(\mathcal{R}^m)^2 / \chi^2}$$

holds. Clearly Eq. (27) also implies the other requirement of $\kappa \cdot \eta_\epsilon(\mathcal{R}^m) / \chi < 1$. $\square$

Lemma 20. *For any PPT $\mathcal{A}$ there exists a PPT $\mathcal{B}$ such that the advantage of $\mathcal{A}$ in distinguishing Hyb_1 and Hyb_2 is equal to $\text{Adv}_{\text{par}_L, \mathcal{B}}^{\text{LLWE}}(\lambda)$.*

Proof. Given a PPT $\mathcal{A}$ distinguishing Hyb_1 and Hyb_2 , we construct a PPT reduction $\mathcal{B}$ against $\text{LLWE}_{\text{par}_L}$, as follows:

- Receive $\mathbf{A}$ from the LLWE challenger, reply with the leakage matrix $\mathbf{Z} := (\mathbf{I}_m | \dots | \mathbf{I}_m) \in \mathcal{R}^{m \times mQ}$.
- Receive $(\mathbf{b}, \mathbf{l})$ from the LLWE challenger, pass $(\mathbf{A}, \mathbf{b})$ to $\mathcal{A}$. Also, write $\mathbf{l}^\top = (\mathbf{l}_1^\top, \dots, \mathbf{l}_Q^\top)$ with each $\mathbf{l}_i \in \mathcal{R}^m$.
- Maintain a counter $i := 0$. Answer $\mathcal{A}$'s oracle queries $\text{LeakO}(\mathbf{z})$ as follows:
 - Assert $i < Q$ and $\mathbf{z} \in \mathcal{L}_A$, and set $i := i + 1$.
 - Sample $f \leftarrow \$ D_{\mathcal{R}, \sqrt{\Sigma_{\mathbf{z}}}}$ and answer with $\mathbf{l}_i^\top \cdot \mathbf{z} + f$.
- Eventually receive a bit b' from $\mathcal{A}$, return b' .

The reduction $\mathcal{B}$ is admissible as an $\text{LLWE}_{\text{par}_L}$ adversary, since for any $\mathbf{y} \in \mathcal{K}^m$ we have $\mathbf{Z}^\top \cdot \mathbf{y} = (\mathbf{y}^\top, \dots, \mathbf{y}^\top)^\top \in \mathcal{K}^{mQ}$, which implies $\|\mathbf{Z}^\top \cdot \mathbf{y}\|_2 = \sqrt{Q} \cdot \|\mathbf{y}\|_2$, so $\|\mathbf{Z}^\top\|_{2,2} = \sqrt{Q}$ and $\mathbf{Z} \in \mathcal{L}_L$.

The simulation is perfect: in the $\text{LLWE}_{\text{par}_L}$ experiment with challenge bit b , the matrix $\mathbf{A}$ and the error $\bar{\mathbf{e}} \leftarrow \$ D_{\mathcal{R}, \bar{\chi}}^m$ are distributed as in Hyb_1 and Hyb_2 , the vector $\mathbf{b}$ is as in Hyb_1 if $b = 0$ and as in Hyb_2 if $b = 1$. In either game, by the choice of $\mathbf{Z}$,

$$\mathbf{l}^\top = \bar{\mathbf{e}}^\top \cdot \mathbf{Z} + \mathbf{e}^\top \quad // \quad \mathbf{e} \leftarrow \$ D_{\mathcal{R}, \chi/\kappa}^{mQ}$$

$$\begin{aligned}
&= \bar{\mathbf{e}}^T \cdot (\mathbf{I}_m | \dots | \mathbf{I}_m) + (\mathbf{e}_1^T | \dots | \mathbf{e}_Q^T) \quad // \quad (\mathbf{e}_1^T | \dots | \mathbf{e}_Q^T) := \mathbf{e}^T \\
&= (\bar{\mathbf{e}}^T + \mathbf{e}_1^T | \dots | \bar{\mathbf{e}}^T + \mathbf{e}_Q^T) \\
\iff &\mathbf{l}_i = \bar{\mathbf{e}} + \mathbf{e}_i \quad \forall i \in [Q], \quad // \quad \mathbf{e}_i \leftarrow \$ D_{\mathcal{R}, \chi/\kappa}^m
\end{aligned}$$

hence each $\text{LeakO}(\mathbf{z})$ query answer is $(\bar{\mathbf{e}} + \mathbf{e}_i)^T \cdot \mathbf{z} + f$, as wanted. Overall, $\mathcal{B}$'s answers are exactly those of Hyb_1 , resp. Hyb_2 . $\square$

By Lemmas 16 and 18, standard LWE thus implies fully-adaptive Leaky LWE, summarised below.

Corollary 2 (Adapted from [HKM⁺26, Theorem 1]). *Let $\mathcal{R}, n, m, Q, q, \epsilon, \alpha, \kappa, \bar{\chi}, \chi, \chi^*, \mathcal{D}$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ and field expansion factor $\mathfrak{c}_2$ w.r.t. the coefficient ℓ_2 -norm, $n, m, Q, q \in \mathbb{N}$, $\mathcal{D} \sim \mathcal{R}_q^{n \times m}$, $0 < \epsilon \leq 1/2$, $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$, and $\alpha \geq 1$, $\kappa, \bar{\chi}, \chi > 0$ satisfy both*

$$\mathfrak{c}_2 \cdot \alpha \leq \kappa \cdot \sqrt{1 - \kappa^2 \cdot \eta_\epsilon(\mathcal{R}^m)^2 / \chi^2}, \quad (28)$$

$$\sqrt{Q} \leq \frac{\chi}{\kappa} \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}. \quad (29)$$

Denote

$$\text{par} = (\mathcal{R}, n, m, q, \chi^*, \mathcal{D}), \quad \text{par}_A = (\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}_A, \mathcal{D}),$$

where $\mathcal{L}_A = \{\mathbf{z} \in \mathcal{R}^m : \|\mathbf{z}\|_2 \leq \alpha\}$. $\forall$ PPT $\mathcal{A}$, $\exists$ PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}_A, \mathcal{A}}^{\text{AdpLLWE}}(\lambda) \leq \text{Adv}_{\text{par}, \mathcal{B}}^{\text{LWE}}(\lambda) + 4 \cdot Q \cdot \epsilon + 6 \cdot \epsilon / (1 - \epsilon).$$

In particular, if $Q \in \text{poly}(\lambda)$, $\epsilon \in \text{negl}(\lambda)$ and the LWE_{par} assumption holds, then the $\text{AdpLLWE}_{\text{par}_A}$ assumption holds.

Proof. Follows from putting Lemmas 16 and 18 together. In particular, we invoke Lemma 16 with LLWE parameters $\text{par}' = (\mathcal{R}, n, m, mQ, q, \bar{\chi}, \chi/\kappa, \mathcal{L}', \mathcal{D})$, where $\mathcal{L}' = \{\mathbf{Z} \in \mathcal{R}^{m \times mQ} : \|\mathbf{Z}^T\|_{2,2} \leq \alpha'\}$ with $\alpha' = \frac{\chi}{\kappa} \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}$.

Since $\mathcal{L}' \supseteq \mathcal{L}_L := \{\mathbf{Z} \in \mathcal{R}^{m \times mQ} : \|\mathbf{Z}^T\|_{2,2} \leq \sqrt{Q}\}$ by Eq. (29), LLWE with par' implies LLWE with $\text{par}_L = (\mathcal{R}, n, m, mQ, q, \bar{\chi}, \chi/\kappa, \mathcal{L}_L, \mathcal{D})$, the latter being the same as that used in Lemma 18.

Eq. (28) is identical to Eq. (27) required by Lemma 18. $\square$

Remark 9 (On choosing κ for Lemma 18 and Corollary 2). Keeping everything else unchanged, a smaller κ in Lemma 18 and Corollary 2 is desired, since it leads to a larger error parameter χ/κ for LLWE in Lemma 18, i.e. a harder LLWE problem, and it makes Eq. (29) in Corollary 2 more easily satisfied.

Writing $\eta := \eta_\epsilon(\mathcal{R}^m)$, a κ satisfying Eq. (28) exists if and only if

$$\chi \geq 2 \cdot \mathfrak{c}_2 \cdot \alpha \cdot \eta, \quad (30)$$

in which case any κ such that $\kappa^2 \in [\kappa_-^2, \kappa_+^2]$ is admissible, where $\kappa_\pm^2 := \frac{\chi^2}{2\eta^2} \left(1 \pm \sqrt{1 - (2 \cdot \mathfrak{c}_2 \cdot \alpha \cdot \eta / \chi)^2}\right)$.¹⁸ For the minimum κ_- , it holds that

$$\mathfrak{c}_2 \cdot \alpha \leq \kappa_- \leq \sqrt{2} \cdot \mathfrak{c}_2 \cdot \alpha,$$

which follows from $\kappa_-^2 \cdot (1 - \kappa_-^2 \cdot \eta^2 / \chi^2) = \mathfrak{c}_2^2 \cdot \alpha^2$ and from $1 - \sqrt{1-t} \leq t$ for $t \in [0, 1]$, the latter giving $\kappa_-^2 \leq \frac{\chi^2}{2\eta^2} \cdot \frac{4 \cdot \mathfrak{c}_2^2 \cdot \alpha^2 \cdot \eta^2}{\chi^2}$.

A sufficient condition for Eq. (29) at $\kappa = \kappa_-$ is thus

$$\sqrt{2Q} \cdot \mathfrak{c}_2 \cdot \alpha \leq \chi \cdot \sqrt{(2\eta^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}. \quad (31)$$

In the latter sections where we invoke Corollary 2, we will always directly require Eq. (31) instead of Eqs. (28) and (29). This is sufficient, since Eq. (31) and the condition $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$ jointly imply Eq. (30)¹⁹, and Eqs. (30) and (31) in turn imply that Eqs. (28) and (29) hold for the choice $\kappa = \kappa_-$ by the above.

¹⁸ Squaring Eq. (28) gives $\kappa^4 \cdot \eta^2 / \chi^2 - \kappa^2 + \mathfrak{c}_2^2 \cdot \alpha^2 \leq 0$, a quadratic in κ^2 whose roots are the stated $\kappa_\pm^2$. The latter are real if and only if $2 \cdot \mathfrak{c}_2 \cdot \alpha \cdot \eta / \chi \leq 1$.

¹⁹ Since $\chi^* \geq \sqrt{2} \cdot \eta$, the radicand in Eq. (31) is at most $(2\eta^2 + 2\eta^2)^{-1} = (4\eta^2)^{-1}$, so the right-hand side of Eq. (31) is at most $\chi / (2\eta)$. Hence Eq. (31) gives $\chi \geq 2 \cdot \sqrt{2Q} \cdot \mathfrak{c}_2 \cdot \alpha \cdot \eta$, which implies Eq. (30) since $Q \geq 1$.

$\text{Real}_{\text{par}, \mathcal{A}}(1^\lambda)$	$\text{Ideal}_{\text{par}, \mathcal{A}}(1^\lambda)$
$L_{\text{Share}} := \emptyset; \quad L_{\text{Query}} := \emptyset$ $\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$ $(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{ assert } \mathcal{C} < t$ $\mathbf{R} \leftarrow \mathcal{R}_q^{t \times n}$ $\mathbf{S}_{\mathcal{C}} := \mathbf{V}_{\mathcal{C}} \cdot \mathbf{R} \bmod q \in \mathcal{R}_q^{C \times n}$ $\mathbf{r} := \mathbf{v}_0^T \cdot \mathbf{R} \bmod q$ $\mathbf{e} \leftarrow D_{\mathcal{R}, \bar{\chi}}^m$ $\mathbf{b}^T := \mathbf{r}^T \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q$ $b' \leftarrow \mathcal{A}^{\text{GenO}, \text{ShareO}_0}(\mathbf{b}, \xi \cdot \mathbf{S}_{\mathcal{C}})$ $\text{ assert } L_{\text{Share}} \leq Q \quad // \text{ at most } Q \text{ shared } \mathbf{Ax}$ $\text{ return } b'$	$L_{\text{Share}} := \emptyset; \quad L_{\text{Query}} := \emptyset$ $\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$ $(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{ assert } \mathcal{C} < t$ $\mathbf{S}_{\mathcal{C}} \leftarrow \mathcal{R}_q^{C \times n}$ $\mathbf{b} \leftarrow \mathcal{R}_q^m$ $\text{st} \leftarrow \mathcal{S}_{\text{Init}}(\mathbf{A}, \mathbf{b}, \mathbf{S}_{\mathcal{C}}, \mathbf{z}) \quad // \text{ State of simulator } \mathcal{S}$ $b' \leftarrow \mathcal{A}^{\text{GenO}, \text{ShareO}_1}(\mathbf{b}, \xi \cdot \mathbf{S}_{\mathcal{C}})$ $\text{ assert } L_{\text{Share}} \leq Q \quad // \text{ at most } Q \text{ shared } \mathbf{Ax}$ $\text{ return } b'$
$\text{GenO}(\text{id}, \tilde{\mathbf{c}}) \quad // \tilde{\mathbf{c}} \in \mathcal{R}^{(m-m_t)}$ $\text{ if } L_{\text{Query}}[\text{id}] = \emptyset \text{ then}$ $\quad \text{ assert } \ \tilde{\mathbf{c}}\ \leq B_c$ $\quad \tilde{\mathbf{t}} \leftarrow \mathcal{D}_{\mathbf{t}}; \quad \mathbf{x} := \begin{bmatrix} \tilde{\mathbf{c}} \\ \tilde{\mathbf{t}} \end{bmatrix}$ $L_{\text{Query}}[\text{id}] := \mathbf{x}$ $\text{ return } \mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$	$\mathcal{S}_{\text{Init}}(\mathbf{A}, \mathbf{b}, \mathbf{S}_{\mathcal{S}}, \mathbf{z})$ $\mathbf{e} \leftarrow D_{\mathcal{R}, \bar{\chi}}^m; \quad \hat{\mathbf{b}} := \mathbf{b} - \mathbf{e} - \mathbf{z}$ $\text{ return } \text{st} := (\mathbf{A}, \mathbf{b}, \mathbf{S}_{\mathcal{S}}, \hat{\mathbf{b}})$
$\mathcal{S}_{\text{Share}}(\mathbf{x}, \text{st})$ $\mathbf{e}_{\mathbf{x}} \leftarrow D_{\mathcal{R}, \chi}^K$ $\mathbf{w}_{\mathbf{x}} \leftarrow \mathcal{R}_q^{t- \mathcal{C} -1}$	$\mathbf{d} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_{\mathcal{C}} \\ \mathbf{I}_{\mathcal{C}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}^T \mathbf{x} \\ \mathbf{w}_{\mathbf{x}} \\ \mathbf{S}_{\mathcal{C}} \mathbf{Ax} \end{bmatrix} + \mathbf{e}_{\mathbf{x}} \bmod q$ $\text{ return } (\mathbf{d}, \text{st})$
$\text{ShareO}_b(\text{id}, k \in [K])$ $\text{ if } L_{\text{Share}}[\text{id}] = \emptyset \text{ then}$ $\quad \text{ assert } L_{\text{Query}}[\text{id}] \neq \emptyset; \text{ parse } \mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$ $\quad \text{ if } b = 0 \text{ then}$ $\quad \quad \mathbf{e}_{\mathbf{x}} \leftarrow D_{\mathcal{R}, \chi}^K; \quad \mathbf{d} := \xi \cdot \mathbf{VRAx} + \mathbf{e}_{\mathbf{x}} \bmod q$ $\quad \text{ if } b = 1 \text{ then}$ $\quad \quad (\mathbf{d}, \text{st}) \leftarrow \mathcal{S}_{\text{Share}}(\mathbf{x}, \text{st})$ $\quad L_{\text{Share}}[\text{id}] := \mathbf{d}$ $\text{ parse } \mathbf{d} \leftarrow L_{\text{Share}}[\text{id}] \in \mathcal{R}_q^K$ $\text{ return } d_k$	

Fig. 5: $\text{Real}_{\text{par}, \mathcal{A}}$ and $\text{Ideal}_{\text{par}, \mathcal{A}}$ with $\text{par} = ((\mathcal{R}, n, m, q, Q, \bar{\chi}, \chi, \mathcal{D}_{\mathbf{A}}, \mathcal{D}_{\mathbf{t}}, B_c), (t, K, \Xi))$.

Remark 10. The semi-adaptive Leaky LWE in [LSW25] more generally supports bounded-norm leakage also on the LWE secret $\mathbf{r}$, and admits a reduction from LWE when $\mathbf{r}$ is distributed as Gaussian. The adaptive version can be generalised in an analogous way, where the oracle LeakO returns leakages of the form $(\mathbf{r}^T, \mathbf{e}^T) \cdot \mathbf{z} + e$. When $\mathbf{r}$ is Gaussian vector, Lemma 18 and Corollary 2 can be generalised analogously.

G.2 Utility Lemma

This subsection is devoted to proving that the two experiments Real and Ideal in Fig. 5 are computationally indistinguishable under two LWE assumptions, one with respect to the tag distribution $\mathcal{D}_{\mathbf{t}}$ and one with respect to the public matrix distribution $\mathcal{D}_{\mathbf{A}}$. This is formally given in Lemma 21. The majority of its proof is a generalisation of [CLW25, Lemma 10, Eprint].

Lemma 21. *Let $\text{par} = ((\mathcal{R}, n, m, q, Q, \bar{\chi}, \chi, \mathcal{D}_{\mathbf{A}}, \mathcal{D}_{\mathbf{t}}, B_c), (t, K, \Xi))$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ and field expansion factor c_p w.r.t. the coefficient ℓ_p -norm, Ξ is a size- K ξ -subtractive*

set with quality measurements $(\rho, \gamma, \theta) = (\rho(t), \gamma(t), \theta(t))$ (Definition 1), $\mathcal{D}_\mathbf{A}$ and $\mathcal{D}_\mathbf{t}$ are distributions over $\mathcal{R}_q^{n \times m}$ and $\mathcal{R}^{m_t}$ respectively with $m_t < m$, and $B_c > 0$. Let $B_t > 0$ be such that $\Pr(\|\tilde{\mathbf{t}}\| \leq B_t | \tilde{\mathbf{t}} \leftarrow \mathcal{D}_\mathbf{t}) = 1$. Consider the experiments $\text{Real}_{\text{par}, \mathcal{A}}$ and $\text{Ideal}_{\text{par}, \mathcal{A}}$ in Fig. 5, and define the advantage bound

$$\text{Adv}_{\text{par}, \mathcal{A}}^{\text{RealIdeal}}(\lambda) := |\Pr[\text{Real}_{\text{par}, \mathcal{A}}(1^\lambda) = 1] - \Pr[\text{Ideal}_{\text{par}, \mathcal{A}}(1^\lambda) = 1]|.$$

Suppose q and ξ are coprime, $0 < \epsilon \leq 1/2$, and the noise parameters χ_1^*, χ_2^* satisfy both of the following:

- $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$ is such that there exists $\hat{\chi} > 0$ satisfying

$$\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2)^{-1} - \hat{\chi}^{-2}};$$

- $\chi_2^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$, $\gamma \cdot \sqrt{\varphi m} \cdot \max(B_c, B_t) \geq 1$, and, with $\mathbf{c}_t^\otimes := 1$ if t is a power of two and $\mathbf{c}_t^\otimes := \sqrt{t}$ otherwise,

$$\sqrt{2QK \cdot \varphi m} \cdot \mathbf{c}_2 \cdot \mathbf{c}_t^\otimes \cdot \gamma \cdot \max(B_c, B_t) \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi_2^*)^2)^{-1} - \bar{\chi}^{-2}}. \quad (32)$$

Then, for any PPT $\mathcal{A}$, there exist PPT $\mathcal{B}_1, \mathcal{B}_2$ such that

$$\begin{aligned} \frac{1}{t} \cdot \text{Adv}_{\text{par}, \mathcal{A}}^{\text{RealIdeal}}(\lambda) &\leq \text{Adv}_{(\mathcal{R}, m_t, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_\mathbf{t}), \mathcal{B}_1}^{\text{LWE}}(\lambda) + \text{Adv}_{(\mathcal{R}, n, m, q, \chi_2^*, \mathcal{D}_\mathbf{A}), \mathcal{B}_2}^{\text{LWE}}(\lambda) \\ &\quad + \frac{6 \cdot Q \cdot \epsilon}{t(1 - \epsilon)} + 4 \cdot QK \cdot \epsilon + \frac{6 \cdot \epsilon}{1 - \epsilon}. \end{aligned}$$

In particular, if $QK \in \text{poly}(\lambda)$, $\epsilon \in \text{negl}(\lambda)$, and the $\text{LWE}_{(\mathcal{R}, m_t, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_\mathbf{t})}$ and $\text{LWE}_{(\mathcal{R}, n, m, q, \chi_2^*, \mathcal{D}_\mathbf{A})}$ assumptions hold, then the experiments $\text{Real}_{\text{par}, \mathcal{A}}$ and $\text{Ideal}_{\text{par}, \mathcal{A}}$ are computationally indistinguishable.

Proof. Let the experiment $\text{Real}_{\text{par}, \mathcal{A}}$ be Hyb_0 , and we define experiments $\text{Hyb}_1, \text{Hyb}_2, \text{Hyb}_3$ and Hyb_4 in Fig. 6. Below we argue that Hyb_i is indistinguishable from Hyb_{i+1} for each $i \in \{0, 1, 2, 3\}$. The proof is then completed by noting that Hyb_4 is identical to the experiment $\text{Ideal}_{\text{par}, \mathcal{A}}$.

Hyb_1 . This is functionally equivalent to $\text{Hyb}_0 = \text{Real}_{\text{par}, \mathcal{A}}$. Notice that $\mathbf{V} = \begin{bmatrix} \mathbf{V}_\mathcal{H} \\ \mathbf{V}_\mathcal{C} \end{bmatrix}$ by definition, thus the

only differences in Hyb_1 are that we sample uniformly random $\begin{bmatrix} \mathbf{r}^\mathbf{T} \\ \mathbf{W} \\ \mathbf{S}_\mathcal{C} \end{bmatrix}$, and we swap

$$\begin{bmatrix} \mathbf{v}_0^\mathbf{T} \\ \xi \cdot \mathbf{V} \end{bmatrix} \mathbf{R} = \begin{bmatrix} \mathbf{v}_0^\mathbf{T} \\ \xi \cdot \mathbf{V}_\mathcal{H} \\ \xi \cdot \mathbf{V}_\mathcal{C} \end{bmatrix} \mathbf{R} \quad \text{to} \quad \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}^\mathbf{T} \\ \mathbf{W} \\ \mathbf{S}_\mathcal{C} \end{bmatrix} = \begin{bmatrix} 1 & | & | \\ \mathbf{V}_\mathcal{H} \cdot \mathbf{T}_\mathcal{C} & | & | \\ | & | & \xi \mathbf{I}_\mathcal{C} \end{bmatrix} \begin{bmatrix} \mathbf{r}^\mathbf{T} \\ \mathbf{W} \\ \mathbf{S}_\mathcal{C} \end{bmatrix}.$$

We recall $\mathbf{T}_\mathcal{C} \in \mathcal{R}^{t \times t}$ is such that it holds $\xi^{-1} \cdot \mathbf{T}_\mathcal{C} \cdot \mathbf{V}_{\mathcal{C}^*} = \mathbf{I}_t$ for some size- t set $\mathcal{C}^*$ satisfying $(\{0\} \cup \mathcal{C}) \subseteq \mathcal{C}^* \subseteq [0, K]$. To see that the two terms are functionally equivalent, we denote $\bar{\mathcal{C}} := \mathcal{C}^* \setminus (\{0\} \cup \mathcal{C})$ and notice

$$\begin{bmatrix} \mathbf{v}_0^\mathbf{T} \\ \xi \cdot \mathbf{V} \end{bmatrix} \mathbf{R} = \begin{bmatrix} \mathbf{v}_0^\mathbf{T} \\ \xi \cdot \mathbf{V}_\mathcal{H} \\ \xi \cdot \mathbf{V}_\mathcal{C} \end{bmatrix} \mathbf{R} = \begin{bmatrix} 1 \cdot \mathbf{v}_0^\mathbf{T} \\ \xi \cdot \xi^{-1} \cdot \mathbf{V}_\mathcal{H} \cdot \mathbf{T}_\mathcal{C} \cdot \mathbf{V}_{\mathcal{C}^*} \\ \xi \cdot \mathbf{I}_\mathcal{C} \cdot \mathbf{V}_\mathcal{C} \end{bmatrix} \mathbf{R} = \begin{bmatrix} 1 & | & | \\ \mathbf{V}_\mathcal{H} \cdot \mathbf{T}_\mathcal{C} & | & | \\ | & | & \xi \mathbf{I}_\mathcal{C} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{v}_0^\mathbf{T} \\ \mathbf{V}_{\bar{\mathcal{C}}} \\ \mathbf{V}_\mathcal{C} \end{bmatrix}}_{=\mathbf{V}_{\mathcal{C}^*}} \mathbf{R} \equiv \begin{bmatrix} 1 & | & | \\ \mathbf{V}_\mathcal{H} \cdot \mathbf{T}_\mathcal{C} & | & | \\ | & | & \xi \mathbf{I}_\mathcal{C} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{r}^\mathbf{T} \\ \mathbf{W} \\ \mathbf{S}_\mathcal{C} \end{bmatrix}}_{\text{uniform}} \text{mod } q,$$

where the first $=$ is by definition of $\mathbf{V}$, the second $=$ holds since $\xi^{-1} \cdot \mathbf{T}_\mathcal{C} \cdot \mathbf{V}_{\mathcal{C}^*} = \mathbf{I}_t$, and the last $\equiv$ holds because ξ and q are coprime and thus $\mathbf{V}_{\mathcal{C}^*}^{-1}$ exists over $\mathcal{R}_q$.²⁰

²⁰ By definition of Ξ , $\xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \in \mathcal{R}^{t \times t}$. Since ξ and q are coprime, there exists $a \in \mathcal{R}$ s.t. $a \cdot \xi = 1 + q \cdot h$ for some $h \in \mathbb{Z}$. We have $a \cdot \xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \cdot \mathbf{V}_{\mathcal{C}^*} = (1 + q \cdot h) \cdot \mathbf{I}$, and $a \cdot \xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \in \mathcal{R}^{t \times t}$. Therefore, $a \cdot \xi \cdot \mathbf{V}_{\mathcal{C}^*}^{-1} \text{mod } q$ is the inverse of $\mathbf{V}_{\mathcal{C}^*}$ over $\mathcal{R}_q$.

Hyb ₁ (1 ^λ)	Hyb ₂ (1 ^λ)	Hyb ₃ (1 ^λ)	Hyb ₄ (1 ^λ)
$L_{\text{Share}}[\cdot] := \emptyset$	$L_{\text{Share}}[\cdot] := \emptyset$	$L_{\text{Share}}[\cdot] := \emptyset$	$L_{\text{Share}}[\cdot] := \emptyset$
$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$	$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$	$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$	$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$
$(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{assert } \mathcal{C} < t$	$(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{assert } \mathcal{C} < t$	$(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{assert } \mathcal{C} < t$	$(\mathcal{C}, \mathbf{z}) \leftarrow \mathcal{A}(\mathbf{A}); \text{assert } \mathcal{C} < t$
$\begin{bmatrix} \mathbf{r}^T \\ \mathbf{W} \\ \mathbf{S}_C \end{bmatrix} \leftarrow \mathcal{R}_q^{t \times n}$	$\begin{bmatrix} \mathbf{r}_m^T \\ \mathbf{W}_m \end{bmatrix} \leftarrow \mathcal{R}_q^{(t- \mathcal{C}) \times m}$	$\mathbf{S}_C \leftarrow \mathcal{R}_q^{C \times n}$ $\mathbf{r}_m^T \leftarrow \mathcal{R}_q^{1 \times m}$	$\mathbf{S}_C \leftarrow \mathcal{R}_q^{C \times n}$
$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^m$	$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^m$	$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^m$	$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^m$
$\mathbf{b}^T := \mathbf{r}^T \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q$	$\mathbf{b}^T := \mathbf{r}_m^T + \mathbf{e}^T + \mathbf{z}^T \bmod q$	$\mathbf{b}^T := \mathbf{r}_m^T + \mathbf{e}^T + \mathbf{z}^T \bmod q$	$\mathbf{b}^T \leftarrow \mathcal{R}_q^m$
$b' \leftarrow \mathcal{A}^{\text{ShareOHyb1}}(\mathbf{b}, \xi \cdot \mathbf{S}_C)$	$b' \leftarrow \mathcal{A}^{\text{ShareOHyb2}}(\mathbf{b}, \xi \cdot \mathbf{S}_C)$	$b' \leftarrow \mathcal{A}^{\text{ShareOHyb3}}(\mathbf{b}, \xi \cdot \mathbf{S}_C)$	$b' \leftarrow \mathcal{A}^{\text{ShareOHyb4}}(\mathbf{b}, \xi \cdot \mathbf{S}_C)$
assert $ L_{\text{Share}} \leq Q$	assert $ L_{\text{Share}} \leq Q$	assert $ L_{\text{Share}} \leq Q$	assert $ L_{\text{Share}} \leq Q$
return b'	return b'	return b'	return b'
ShareOHyb1(id, $k \in [K]$)	ShareOHyb2(id, $k \in [K]$)	ShareOHyb3(id, $k \in [K]$)	ShareOHyb4(id, $k \in [K]$)
if $L_{\text{Share}}[\text{id}] = \emptyset$ then	if $L_{\text{Share}}[\text{id}] = \emptyset$ then	if $L_{\text{Share}}[\text{id}] = \emptyset$ then	if $L_{\text{Share}}[\text{id}] = \emptyset$ then
assert $L_{\text{Query}}[\text{id}] \neq \emptyset$	assert $L_{\text{Query}}[\text{id}] \neq \emptyset$	assert $L_{\text{Query}}[\text{id}] \neq \emptyset$	assert $L_{\text{Query}}[\text{id}] \neq \emptyset$
parse $\mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$	parse $\mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$	parse $\mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$	parse $\mathbf{x} \leftarrow L_{\text{Query}}[\text{id}]$
$\mathbf{e}_x \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$	$\mathbf{e}_x \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$	$\mathbf{e}_x \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$	$\mathbf{e}_x \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$
$\mathbf{M} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C \\ \mathbf{r}^T \\ \xi \mathbf{I}_C \end{bmatrix}$	$\mathbf{M} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C \\ \mathbf{r}_m^T \\ \xi \mathbf{I}_C \end{bmatrix}$	$\mathbf{M} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C \\ \mathbf{r}_m^T \\ \xi \mathbf{I}_C \end{bmatrix}$	$\mathbf{M} := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C \\ \mathbf{r}_m^T \\ \xi \mathbf{I}_C \end{bmatrix}$
$\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}^T \\ \mathbf{W} \\ \mathbf{S}_C \end{bmatrix} \mathbf{A} \mathbf{x} + \mathbf{e}_x \bmod q$	$\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}_m^T \\ \mathbf{W}_m \\ \mathbf{S}_C \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e}_x \bmod q$	$\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}_m^T \mathbf{x} \\ \mathbf{w}_x \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_x \bmod q$	$\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} (\mathbf{b} - \mathbf{e} - \mathbf{z})^T \mathbf{x} \\ \mathbf{w}_x \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_x \bmod q$
$L_{\text{Share}}[\text{id}] := \mathbf{d} \in \mathcal{R}_q^K$	$L_{\text{Share}}[\text{id}] := \mathbf{d} \in \mathcal{R}_q^K$	$L_{\text{Share}}[\text{id}] := \mathbf{d} \in \mathcal{R}_q^K$	$L_{\text{Share}}[\text{id}] := \mathbf{d} \in \mathcal{R}_q^K$
parse $\mathbf{d} \leftarrow L_{\text{Share}}[\text{id}]$	parse $\mathbf{d} \leftarrow L_{\text{Share}}[\text{id}]$	parse $\mathbf{d} \leftarrow L_{\text{Share}}[\text{id}]$	parse $\mathbf{d} \leftarrow L_{\text{Share}}[\text{id}]$
return d_k	return d_k	return d_k	return d_k

Fig. 6: Hybrid distributions Hyb₁ to Hyb₄ for proof of Lemma 21. For each, the changes relative to the previous hybrid are marked in red. The oracle GenO is unchanged and omitted.

Hyb₂. We swap the vector $\mathbf{b}^T$ and each vector $\mathbf{d}$ from

$$\begin{aligned}
 & \mathbf{r}^T \mathbf{A} + \mathbf{e}^T + \mathbf{z}^T \bmod q & \text{and} & & \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}^T \\ \mathbf{W} \\ \mathbf{S}_C \end{bmatrix} \mathbf{A} \mathbf{x} + \mathbf{e}_x \bmod q \\
 & \text{to} \quad \mathbf{r}_m^T + \mathbf{e}^T + \mathbf{z}^T \bmod q & \text{and} & & \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}_m^T \\ \mathbf{W}_m \\ \mathbf{S}_C \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e}_x \bmod q
 \end{aligned}$$

respectively, where $\mathbf{r}_m, \mathbf{W}_m$ are uniformly random. For this, we will invoke Theorem 5, instantiating its parameters (t, t_2) with $(t - |\mathcal{C}|, |\mathcal{C}|)$ and its matrix with

$$\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2) := \begin{bmatrix} \mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C \\ \mathbf{r}_m^T \\ \xi \mathbf{I}_C \end{bmatrix} \in \mathcal{R}^{K \times ((t-|\mathcal{C}|) + |\mathcal{C}|)},$$

so that its secret matrices are $\mathbf{S}_1 = \begin{bmatrix} \mathbf{r}^T \\ \mathbf{W} \end{bmatrix}$ and $\mathbf{S}_2 = \mathbf{S}_C$, and its unit vector $\mathbf{u}$ selects the row $\mathbf{r}^T$ of $\mathbf{S}_1$,

whose noisy image is $\mathbf{b}$. We bound the ℓ_2 -induced norm of $\mathbf{M}_1$ and the norm of $\mathbf{x} := \begin{bmatrix} \tilde{\mathbf{c}} \\ \tilde{\mathbf{t}} \end{bmatrix}$:

$$\begin{aligned}
 \|\mathbf{M}_1\|_{2,2} &\leq \|\mathbf{V}_{\mathcal{H}} \cdot \mathbf{T}_C\|_{2,2} = \|\xi \cdot \mathbf{V}_{\mathcal{H}} \cdot \mathbf{V}_C^{-1}\|_{2,2} \leq \gamma, \\
 \|\mathbf{x}\|_{\infty} &\leq \max(B_c, B_t),
 \end{aligned}$$

where the first inequality holds since $\mathbf{M}_1$ consists of the first $t - |\mathcal{C}|$ columns of $\mathbf{M}$, the block $\xi \mathbf{I}_C$ contributing to $\mathbf{M}_2$ only, and where we recall γ is a quality measurement of Ξ (ref. Eq. (4)): the last inequality holds by taking $T^* = \mathcal{C}^* \setminus \{0\}$ in Eq. (4), since the rows of $\mathbf{V}_{\mathcal{H}}$ are a subset of the rows of $\mathbf{V}_{\Xi}$

and removing rows does not increase the ℓ_2 -induced norm. The remaining requirements of Theorem 5 are satisfied by assumptions in this lemma: We set the noise parameter $\chi^* = \chi_2^*$, the norm bound $\beta = \max(B_c, B_t)$. The two constraints in Theorem 5, at its instantiated threshold $t - |\mathcal{C}|$, follow from $\gamma \cdot \sqrt{\varphi m} \cdot \max(B_c, B_t) \geq 1$ and Eq. (32), since $1 \leq \mathbf{c}_{t-|\mathcal{C}|}^\otimes \leq \mathbf{c}_t^\otimes$. Then, by Theorem 5, whose advantage bound carries the multiplicative factor $t - |\mathcal{C}| \leq t$, the advantage of any PPT algorithm $\mathcal{B}_2$ in distinguishing Hyb_1 and Hyb_2 is upper-bounded by

$$t \cdot \left(\text{Adv}_{(\mathcal{R}, n, m, q, \chi_2^*, \mathcal{D}_A), \mathcal{B}_2}^{\text{LWE}}(\lambda) + 4 \cdot QK \cdot \epsilon + \frac{6 \cdot \epsilon}{1 - \epsilon} \right).$$

(To use Theorem 5, the trivial reduction forwards $\mathbf{A}$ to $\mathcal{A}$, adds $\mathbf{z}^\top$ to the vector $\mathbf{b}^\top$ received from the experiment Exp2^b , passes $\xi \mathbf{S}_2$ in place of $\mathbf{S}_2$, and answers the first ShareO query on each id by querying $\mathcal{O}_b$ once on the vector $\mathbf{x}$ recorded in $L_{\text{Query}}[\text{id}]$ and storing the answer under id , and every later query on the same id from that stored answer.)

Hyb_3 . We swap each vector $\mathbf{d}$ from

$$\mathbf{M} \cdot \begin{bmatrix} \mathbf{r}_m^\top \\ \mathbf{W}_m \\ \mathbf{S}_C \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e}_\mathbf{x} \bmod q \quad \text{to} \quad \mathbf{M} \cdot \begin{bmatrix} \mathbf{r}_m^\top \mathbf{x} \\ \mathbf{w}_\mathbf{x} \\ \mathbf{S}_C \mathbf{A} \mathbf{x} \end{bmatrix} + \mathbf{e}_\mathbf{x} \bmod q$$

where $\mathbf{w}_\mathbf{x}$ is uniformly random. For this, we will invoke Theorem 4, for which we again write $\mathbf{M} = (\mathbf{M}_0, \mathbf{M}_1, \mathbf{M}_2)$; here $\mathbf{M}_1$ (width $t_1 = t - |\mathcal{C}| - 1$) is the block swapped to uniform, while $\mathbf{M}_0, \mathbf{M}_2$ (total width $1 + |\mathcal{C}|$) are left unchanged and together play the role of Theorem 4's second block, so we instantiate Theorem 4 with $t_1 = t - |\mathcal{C}| - 1$ and $t_2 = 1 + |\mathcal{C}|$. We bound the norm of $\mathbf{M}_1$:

$$\|\mathbf{M}_1\|_{2,2} \leq \|\mathbf{V}_\mathcal{H} \cdot \mathbf{T}_C\|_{2,2} = \|\xi \cdot \mathbf{V}_\mathcal{H} \cdot \mathbf{V}_C^{-1}\|_{2,2} \leq \gamma,$$

where we recall γ is a quality measurement of Ξ (ref. Eq. (4)), and the last inequality holds for the same reason as in the Hyb_2 step. Then, by Theorem 4, whose advantage bound carries the multiplicative factor $t_1 = t - |\mathcal{C}| - 1 \leq t$, the advantage of any PPT algorithm $\mathcal{B}_1$ in distinguishing Hyb_2 and Hyb_3 is upper-bounded by

$$t \text{Adv}_{(\mathcal{R}, m_t, \text{poly}(\lambda), q, \chi_1^*, \mathcal{D}_t), \mathcal{B}_1}^{\text{LWE}}(\lambda) + \frac{6 \cdot Q \cdot \epsilon}{1 - \epsilon},$$

where $\chi_1^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^t)$ is such that there exists $\hat{\chi} > 0$ satisfying

$$\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^t)^2 + (\chi_1^*)^2)^{-1} - \hat{\chi}^{-2}},$$

as assumed in the statement of this lemma. (To use Theorem 4, the trivial reduction queries TagO from the experiments in Theorem 4 to obtain $\tilde{\mathbf{t}}$ to answer GenO queries, simply forwards the ShareO queries to the oracle $\mathcal{O}$, and generates the rest itself.)

Hyb_4 . This is functionally equivalent to Hyb_3 . We rewrite the vectors $\mathbf{b} = \mathbf{r}_m + \mathbf{e} + \mathbf{z} \bmod q$ and $\mathbf{r}_m \leftarrow \mathcal{R}_q^m$ in Hyb_3 as $\mathbf{b} \leftarrow \mathcal{R}_q^m$ and $(\mathbf{b} - \mathbf{e} - \mathbf{z})$ in Hyb_4 . This is a simple change of variables of uniformly random vectors, and clearly the two hybrids are identically distributed. $\square$

We provide Theorems 4 and 5 involved in Lemma 21 in the remaining subsections.

G.3 Theorem 4 and proofs

Theorem 4. Let $\mathcal{R}, m, Q, q, m_c, K, t_1, t_2, \chi^*, \chi, \gamma, \mathcal{D}_t$ and $\epsilon \in (0, 1)$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ , $m, Q, m_c, K, t_1, t_2 \in \text{poly}(\lambda)$ and $\mathcal{D}_t$ is a distribution over $\mathcal{R}_q^m$. Suppose χ^*, χ are such that $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^{t_1})$, and there exists $\hat{\chi} > 0$ such that $\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^{t_1})^2 + (\chi^*)^2)^{-1} - \hat{\chi}^{-2}}$. Denote

$$\text{par} = (\mathcal{R}, m, Q, q, m_c, K, t_1, t_2, \chi, \gamma, \mathcal{D}_t).$$

For any PPT $\mathcal{A}$, define the following experiment $\text{Exp1}_{\text{par}, \mathcal{A}}^b$:

$\text{Exp1}_{\text{par},\mathcal{A}}^b(1^\lambda)$	$\text{O}_b(\mathbf{c} \in \mathcal{R}^{m_c}, i)$
$L_{\text{Query}}[\cdot] := \emptyset; L_t[\cdot] := \emptyset$ $\left(\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2), \mathbf{S}_2 \right) \leftarrow \mathcal{A}(1^\lambda)$ $\text{assert } \ \mathbf{M}_1\ _{2,2} \leq \gamma$ $\mathbf{S}_1 \leftarrow \mathcal{R}_q^{t_1 \times (m_c + m)}$ $b' \leftarrow \mathcal{A}^{\text{TagO}, \text{O}_b}(1^\lambda)$ return b'	$\text{assert } L_{\text{Query}} < Q$ if $L_{\text{Query}}[i] = \emptyset$ then $\text{assert } L_t[i] \neq \emptyset; \text{ parse } \mathbf{t}_i \leftarrow L_t[i]$ $\mathbf{x} := \begin{bmatrix} \mathbf{c} \\ \mathbf{t}_i \end{bmatrix}; \mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$ if $b = 0$ then $\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \mathbf{x} + \mathbf{e}$ if $b = 1$ then $\mathbf{w} \leftarrow \mathcal{R}_q^{t_1}; \mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{w} \\ \mathbf{S}_2 \mathbf{x} \end{bmatrix} + \mathbf{e}$ $L_{\text{Query}}[i] := \mathbf{d}$ return $\mathbf{d} \leftarrow L_{\text{Query}}[i]$
$\text{TagO}()$	
$i := L_t + 1; \mathbf{t}_i \leftarrow \mathcal{D}_t$ $L_t[i] := \mathbf{t}_i$ return $(i, \mathbf{t}_i)$	

and denote the advantage of $\mathcal{A}$ in distinguishing the above experiments by

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp1}} := |\Pr[\text{Exp1}_{\text{par},\mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp1}_{\text{par},\mathcal{A}}^1(1^\lambda) = 1]|. \quad (33)$$

For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp1}}(\lambda) \leq t_1 \cdot \text{Adv}_{(\mathcal{R}, m, \text{poly}(\lambda), q, \chi^*, \mathcal{D}_t), \mathcal{B}}^{\text{LWE}}(\lambda) + 2Q \cdot \frac{3\epsilon}{1 - \epsilon}.$$

In particular, if the $\text{LWE}_{\mathcal{R}, m, \text{poly}(\lambda), q, \chi^*, \mathcal{D}_t}$ assumption holds, then the experiments $\text{Exp1}_{\text{par},\mathcal{A}}^0$ and $\text{Exp1}_{\text{par},\mathcal{A}}^1$ are computationally indistinguishable.

Proof. Follows directly by putting Lemmas 23 and 24 together. (We rename $\bar{\chi}$ in Lemma 23 to $\hat{\chi}$.) $\square$

We will use the following statistical lemma from [LSW25] to prove Lemma 23. The proof of Lemma 23 is essentially an adaption of [LSW25, Thm.3].

Lemma 22 (Special case of [LSW25, Thm. 2]). *Let $m_1, m_2 \in \mathbb{N}$ and $s_1, s_2, \chi^* > 0$. Fix any $\mathbf{Z} \in \mathbb{Z}^{m_1 \times m_2}$. Define the following matrices (dependent on $s_1, s_2, \mathbf{Z}$):*

$$\begin{aligned} \mathbf{\Gamma} &:= s_2^{-2} \cdot \mathbf{Z} \in \mathbb{R}^{m_1 \times m_2}, \\ \mathbf{\Theta}^{-1} &:= s_1^{-2} \cdot \mathbf{I}_{m_1} + s_2^{-2} \cdot \mathbf{Z}\mathbf{Z}^T \in \mathbb{R}^{m_1 \times m_1}, \\ \mathbf{\Phi}^{-1} &:= s_2^{-2} \cdot \mathbf{I}_{m_2} - \mathbf{\Gamma}^T \cdot \mathbf{\Theta} \cdot \mathbf{\Gamma} \in \mathbb{R}^{m_2 \times m_2}. \end{aligned} \quad (34)$$

Let $\epsilon \in (0, 1)$ and $v^{-1} := s_1^{-2} + s_{\max}(\mathbf{Z}\mathbf{Z}^T) \cdot s_2^{-2}$. Suppose $(\chi^*)^2 < v$ and $((\chi^*)^{-2} + (v - (\chi^*)^2)^{-1})^{-1/2} \geq \eta_\epsilon(\mathbb{Z}^{m_1})$, which implies $\sqrt{v} \geq \eta_\epsilon(\mathbb{Z}^{m_1})$. Then, the statistical distance between the distributions

$$\left\{ \left(\mathbf{e}_1^T, \mathbf{e}_1^T \mathbf{Z} + \mathbf{e}_2^T \right) \left| \begin{array}{l} \mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_1}, s_1} \\ \mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_2}, s_2} \end{array} \right. \right\} \quad \text{and} \quad \left\{ \left(\mathbf{e}^T + \mathbf{f}_1^T, \mathbf{f}_2^T \right) \left| \begin{array}{l} \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^{m_1}, \chi^*}; \mathbf{f}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_2}, \sqrt{\mathbf{\Phi}}} \\ \mathbf{c} := \mathbf{\Theta} \cdot \mathbf{\Gamma} \cdot \mathbf{f}_2 \\ \mathbf{f}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{m_1}, \sqrt{\mathbf{\Theta} - (\chi^*)^2 \mathbf{I}_{m_1}}, \mathbf{c}}} \end{array} \right. \right\}$$

is at most $3\epsilon/(1 - \epsilon)$.

Lemma 23. *Let $\mathcal{R}, m, Q, q, t_1, K, \bar{\chi}, \chi^*, \chi, \gamma, \mathcal{D}_t$ and $\epsilon \in (0, 1)$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ , $m, Q, t_1 \in \text{poly}(\lambda)$ and $\mathcal{D}_t$ is a distribution over $\mathcal{R}_q^m$. Let $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^{t_1})$, and $\gamma \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^{t_1})^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}$. Denote*

$$\text{par} = (\mathcal{R}, m, Q, q, t_1, K, \bar{\chi}, \chi, \gamma, \mathcal{D}_t).$$

For any PPT $\mathcal{A}$, define the following experiment $\text{Exp}_{\text{par},\mathcal{A}}^b$:

$\text{Exp}_{\text{par},\mathcal{A}}^b(1^\lambda)$	$\text{leakO}_b(i)$
$L_{\text{Query}}[\cdot] := \emptyset; L_t[\cdot] := \emptyset \quad // \text{ initial lists}$	assert $ L_{\text{Query}} < Q$
$\mathbf{Z} \in \mathcal{R}^{t_1 \times K} \leftarrow \mathcal{A}(1^\lambda); \text{assert } \ \mathbf{Z}^T\ _{2,2} \leq \gamma$	if $L_{\text{Query}}[i] = \emptyset$ then
$\mathbf{S} \leftarrow \mathcal{R}_q^{t_1 \times m}$	assert $L_t[i] \neq \emptyset; \text{parse } \mathbf{t}_i \leftarrow L_t[i]$
$b' \leftarrow \mathcal{A}^{\text{TagO}, \text{leakO}_b}(\cdot)$	$\bar{\mathbf{e}}_i \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^{t_1}; \mathbf{e}_i \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$
return b'	if $b = 0$ then $\mathbf{b}_i := \mathbf{S} \cdot \mathbf{t}_i + \bar{\mathbf{e}}_i \bmod q$
	if $b = 1$ then $\mathbf{b}_i \leftarrow \mathcal{R}_q^{t_1}$
	$\mathbf{l}_i^T := \bar{\mathbf{e}}_i^T \cdot \mathbf{Z} + \mathbf{e}_i^T \bmod q$
$\text{TagO}()$	$L_{\text{Query}}[i] := (\mathbf{b}_i, \mathbf{l}_i)$
$i := L_t + 1; \mathbf{t}_i \leftarrow \mathcal{D}_t \quad // \mathbf{t}_i \in \mathcal{R}_q^m$	return $(\mathbf{b}_i, \mathbf{l}_i) \leftarrow L_{\text{Query}}[i]$
$L_t[i] := \mathbf{t}_i$	
return $(i, \mathbf{t}_i)$	

and denote the advantage of $\mathcal{A}$ in distinguishing the above experiments by

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp}} := |\Pr[\text{Exp}_{\text{par},\mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp}_{\text{par},\mathcal{A}}^1(1^\lambda) = 1]|.$$

For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp}}(\lambda) \leq t_1 \cdot \text{Adv}_{(\mathcal{R}, m, \text{poly}(\lambda), q, \chi^*, \mathcal{D}_t), \mathcal{B}}^{\text{LWE}}(\lambda) + 2Q \cdot \frac{3\epsilon}{1 - \epsilon}.$$

In particular, if the $\text{LWE}_{\mathcal{R}, m, \text{poly}(\lambda), q, \chi^*, \mathcal{D}_t}$ assumption holds, then the experiments $\text{Exp}_{\text{par},\mathcal{A}}^0$ and $\text{Exp}_{\text{par},\mathcal{A}}^1$ are computationally indistinguishable.

Proof. We will rely on that LWE with a uniformly random secret matrix $\mathbf{S} \leftarrow \mathcal{R}_q^{t_1 \times m}$ is no easy than LWE with a uniformly random secret row vector. This is stated in the following proposition.

Proposition 4. Let $\mathcal{R}, m, q, t_1, \chi^*, \mathcal{D}_A, \mathcal{D}_t$ be parametrised by λ , where $t_1 \in \text{poly}(\lambda)$ and $\mathcal{D}_t$ is a distribution over $\mathcal{R}_q^m$. For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{(\mathcal{R}, m, \text{poly}(\lambda), t_1, q, \chi^*, \mathcal{D}_t), \mathcal{A}}^{\text{LWE}}(\lambda) \leq t_1 \cdot \text{Adv}_{(\mathcal{R}, m, \text{poly}(\lambda), q, \chi^*, \mathcal{D}_t), \mathcal{B}}^{\text{LWE}}(\lambda),$$

The above follows directly from standard hybrid argument, its proof omitted.

In the rest we show that, given a PPT $\mathcal{A}$ distinguishing Exp^0 and Exp^1 , we construct a PPT $\mathcal{B}$ against the $\text{LWE}_{\mathcal{R}, m, \text{poly}(\lambda), t_1, q, \chi^*, \mathcal{D}_t}$ assumption. We will show that if $\mathcal{A}$ distinguishes the two experiments with advantage $\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp}}(\lambda)$, then $\mathcal{B}$ breaks the $\text{LWE}_{\mathcal{R}, m, \text{poly}(\lambda), t_1, q, \chi^*, \mathcal{D}_t}$ assumption with advantage at least

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp}}(\lambda) - 2Q \cdot \frac{3\epsilon}{1 - \epsilon}. \quad (35)$$

The lemma then follows from combining with the proposition above.

The PPT $\mathcal{B}$ works as follows:

- Receive for $\mathcal{A}$ the matrix $\mathbf{Z}$, and assert $\|\mathbf{Z}^T\|_{2,2} \leq \gamma$.
- Internally maintain the lists L_{Query} and L_t . Also, let

$$\mathbf{\Gamma} = \chi^{-2} \cdot \Psi(\mathbf{Z}^T)^T, \quad \mathbf{\Theta} = (\bar{\chi}^{-2} \cdot \mathbf{I}_{\varphi t_1} + \chi^{-2} \cdot \Psi(\mathbf{Z}^T)^T \Psi(\mathbf{Z}^T))^{-1}, \quad \mathbf{\Phi} = (\chi^{-2} \cdot \mathbf{I}_{\varphi K} - \mathbf{\Gamma}^T \mathbf{\Theta} \mathbf{\Gamma})^{-1},$$

where we recall that $\psi: \mathcal{K} \rightarrow \mathbb{Z}^\varphi$ is the coefficient embedding w.r.t. the power basis (Section 3), and $\Psi(\mathbf{M})$ is the matrix of $\mathbf{x} \mapsto \mathbf{M}\mathbf{x}$ in the coefficient embedding, so that $\psi(\mathbf{M}\mathbf{x}) = \Psi(\mathbf{M}) \cdot \psi(\mathbf{x})$. In particular $\psi((\bar{\mathbf{e}}^T \mathbf{Z})^T) = \psi(\mathbf{Z}^T \bar{\mathbf{e}}) = \Psi(\mathbf{Z}^T) \cdot \psi(\bar{\mathbf{e}})$, so the leakage $\bar{\mathbf{e}}^T \mathbf{Z} + \mathbf{e}^T$ is, in coefficients, $\psi(\bar{\mathbf{e}})^T \cdot \Psi(\mathbf{Z}^T)^T + \psi(\mathbf{e})^T$, and Lemma 22 is applied with the integer matrix $\Psi(\mathbf{Z}^T)^T \in \mathbb{Z}^{\varphi t_1 \times \varphi K}$.

- For any query to TagO , do the following:
 - Query the LWE challenger to obtain an LWE sample $(\mathbf{t}_i, \hat{\mathbf{b}}_i)$.
 - Store $L_t[i] := (\mathbf{t}_i, \hat{\mathbf{b}}_i)$, i.e. including also the LWE sample $\hat{\mathbf{b}}_i$, and answer with $\mathbf{t}_i$.
- For any query i to LeakO , do the following:
 - Assert $|L_{\text{Query}}| < Q$, i.e. there has been fewer than Q distinct queries.

- If $L_{\text{Query}}[i] = \emptyset$, assert $L_t[i] \neq \emptyset$ and parse $(\mathbf{t}_i, \hat{\mathbf{b}}_i) \leftarrow L_t[i]$.
 - Sample $\mathbf{l}_i \leftarrow \mathcal{D}_{\mathcal{R}^K, \sqrt{\Phi}}$, set $\mathbf{c}_i := \psi^{-1}(\Theta \cdot \Gamma \cdot \psi(\mathbf{l}_i))$ ²¹ and $\mathbf{F} := \Theta - (\chi^*)^2 \mathbf{I}_{\varphi t_1}$, and sample $\mathbf{f}_i \leftarrow \mathcal{D}_{\mathcal{R}^{t_1}, \sqrt{\mathbf{F}}, \mathbf{c}_i}$. Set $\mathbf{b}_i := \hat{\mathbf{b}}_i + \mathbf{f}_i \bmod q$ and $L_{\text{Query}}[i] := (\mathbf{b}_i, \mathbf{l}_i)$.
 - Answer with $(\mathbf{b}_i, \mathbf{l}_i) \leftarrow L_{\text{Query}}[i]$.
- Eventually receive a bit b' from $\mathcal{A}$, return b' .

We analyse the output of $\mathcal{B}$.

If $\mathcal{B}$ interacts with the LWE experiment 0: In this case, each sample $(\mathbf{t}_i, \hat{\mathbf{b}}_i)$ from the LWE challenger is such that $\hat{\mathbf{b}}_i = \mathbf{S} \cdot \mathbf{t}_i + \hat{\mathbf{e}}_i \bmod q$, where $\hat{\mathbf{e}}_i \leftarrow \mathcal{D}_{\mathcal{R}, \chi^*}^{t_1}$. The simulated leakO answer is thus

$$(\mathbf{b}_i, \mathbf{l}_i) = (\mathbf{S} \cdot \mathbf{t}_i + \hat{\mathbf{e}}_i + \mathbf{f}_i, \mathbf{l}_i) \approx_s (\mathbf{S} \cdot \mathbf{t}_i + \bar{\mathbf{e}}_i, \bar{\mathbf{e}}_i^T \cdot \mathbf{Z} + \mathbf{e}_i^T),$$

where the last line follows from Lemma 22, with statistical distance at most $\frac{3\epsilon}{1-\epsilon}$. Since there are at most Q many queries, the overall view of $\mathcal{A}$ is close to that from Exp^0 with statistical distance at most $Q \cdot \frac{3\epsilon}{1-\epsilon}$.

If $\mathcal{B}$ interacts with the LWE experiment 1: In this case, each sample $(\mathbf{t}_i, \hat{\mathbf{b}}_i)$ from the LWE challenger is such that $\hat{\mathbf{b}}_i \leftarrow \mathcal{R}_q^{t_1}$ is uniformly random. The simulated leakO answer is thus

$$(\mathbf{b}_i, \mathbf{l}_i) = (\hat{\mathbf{b}}_i + \mathbf{f}_i, \mathbf{l}_i) = (\mathcal{U}(\mathcal{R}_q^{t_1}), \mathbf{l}_i) \approx_s (\mathcal{U}(\mathcal{R}_q^{t_1}), \bar{\mathbf{e}}_i^T \cdot \mathbf{Z} + \mathbf{e}_i^T),$$

where the last line again follows from Lemma 22, with statistical distance at most $\frac{3\epsilon}{1-\epsilon}$. Since there are at most Q many queries, the overall view of $\mathcal{A}$ is close to that from Exp^1 with statistical distance at most $Q \cdot \frac{3\epsilon}{1-\epsilon}$.

Combining the two cases yields Eq. (35).

It remains to see that Lemma 22 can indeed be invoked. Write $v^{-1} := \bar{\chi}^{-2} + s_{\max}(\Psi(\mathbf{Z}^T)^T \cdot \Psi(\mathbf{Z}^T)) \cdot \chi^{-2}$. By the lemma condition,

$$s_{\max}(\Psi(\mathbf{Z}^T)^T \cdot \Psi(\mathbf{Z}^T)) = \|\mathbf{Z}^T\|_{2,2}^2 \leq \gamma^2 \leq \chi^2((2\eta_\epsilon(\mathcal{R}^{t_1})^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}),$$

which implies $v^{-1} \leq (2\eta_\epsilon(\mathcal{R}^{t_1})^2 + (\chi^*)^2)^{-1}$, and therefore $v - (\chi^*)^2 \geq 2\eta_\epsilon(\mathcal{R}^{t_1})^2$. Together with the other condition of $(\chi^*)^2 \geq 2\eta_\epsilon(\mathcal{R}^{t_1})^2$, we have $(\chi^*)^2 < v$ and $((\chi^*)^{-2} + (v - (\chi^*)^2)^{-1})^{-1/2} \geq \eta_\epsilon(\mathcal{R}^{t_1})$, hence the requirements of Lemma 22 are satisfied. $\square$

Lemma 24. *Let $\mathcal{R}, m, Q, q, m_c, K, t_1, t_2, \hat{\chi}, \chi, \gamma, \mathcal{D}_t$ be parametrised by λ , where $\mathcal{D}_t$ is a distribution over $\mathcal{R}_q^m$. Denote*

$$\text{par} = (\mathcal{R}, m, Q, q, m_c, K, t_1, t_2, \chi, \gamma, \mathcal{D}_t), \quad \text{par}_E = (\mathcal{R}, m, Q, q, t_1, K, \hat{\chi}, \chi, \gamma, \mathcal{D}_t).$$

Consider the experiment $\text{Exp1}_{\text{par}, \mathcal{A}}^b$ defined in Theorem 4 and the advantage $\text{Adv}_{\text{par}, \mathcal{A}}^{\text{Exp1}}$ defined in Eq. (33), and let Exp denote the experiment of Lemma 23 with parameters par_E . For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}, \mathcal{A}}^{\text{Exp1}}(\lambda) = \text{Adv}_{\text{par}_E, \mathcal{B}}^{\text{Exp}}(\lambda).$$

Proof. Given a PPT $\mathcal{A}$ distinguishing Exp1^0 and Exp1^1 , we construct a PPT $\mathcal{B}$ distinguishing Exp^0 and Exp^1 in Lemma 23, as follows:

- Receive from $\mathcal{A}(1^\lambda)$ the matrices $(\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2), \mathbf{S}_2)$, and assert $\|\mathbf{M}_1\|_{2,2} \leq \gamma$.
- Send $\mathbf{Z} := \mathbf{M}_1^T \in \mathcal{R}^{t_1 \times K}$ to the Exp challenger.
- Sample $\mathbf{S}_{1,c} \leftarrow \mathcal{R}_q^{t_1 \times m_c}$.
- Maintain a list L_{Query} . Answer $\mathcal{A}$'s oracle queries as follows:
 - **TagO**(i): forward the query to the tag oracle of Exp, obtain $(i, \mathbf{t}_i)$, and answer with $(i, \mathbf{t}_i)$.

²¹ We abuse notation slightly, because $\Theta \cdot \Gamma \cdot \psi(\mathbf{l}_i)$ may not lie in the image of ψ . This is not an issue, as we only operate on $\psi(\mathbf{c})$.

- $O_b(\mathbf{c}, i)$: if $L_{\text{Query}}[i] = \emptyset$, query the leak oracle $\text{leakO}_b(i)$ of Exp to obtain $(\mathbf{b}_i, \mathbf{l}_i)$, set $\mathbf{x} := \begin{bmatrix} \mathbf{c} \\ \mathbf{t}_i \end{bmatrix}$, compute

$$\mathbf{d} := \mathbf{M}_1 \cdot (\mathbf{S}_{1,c} \cdot \mathbf{c}) + \mathbf{M}_1 \cdot \mathbf{b}_i - \mathbf{l}_i + \mathbf{M}_2 \cdot (\mathbf{S}_2 \cdot \mathbf{x}) \bmod q,$$

- and set $L_{\text{Query}}[i] := \mathbf{d}$. Answer with $\mathbf{d} \leftarrow L_{\text{Query}}[i]$.
- Eventually receive a bit b' from $\mathcal{A}$, return b' .

Below we show that $\mathcal{B}$ perfectly simulates the view of either $\text{Exp}1^0$ or $\text{Exp}1^1$ to $\mathcal{A}$, if it interacts respectively with Exp^0 or Exp^1 , thus the lemma follows.

If $\mathcal{B}$ interacts with Exp^0 : By Lemma 23, for each fresh index i the leak oracle returns

$$\mathbf{b}_i = \mathbf{S} \cdot \mathbf{t}_i + \bar{\mathbf{e}}_i \bmod q, \quad \mathbf{l}_i = \mathbf{M}_1 \cdot \bar{\mathbf{e}}_i + \mathbf{e}_i,$$

where $\mathbf{S} \leftarrow \mathcal{R}_q^{t_1 \times m}$ is the fixed secret of Exp , $\bar{\mathbf{e}}_i \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^{t_1}$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$, and we used $\mathbf{Z} = \mathbf{M}_1^T$. Substituting,

$$\begin{aligned} \mathbf{d} &= \mathbf{M}_1 \cdot \mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{M}_1 \cdot (\mathbf{S} \cdot \mathbf{t}_i + \bar{\mathbf{e}}_i) - (\mathbf{M}_1 \cdot \bar{\mathbf{e}}_i + \mathbf{e}_i) + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \\ &= \mathbf{M}_1 \cdot (\mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{S} \cdot \mathbf{t}_i) - \mathbf{e}_i + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \\ &= \mathbf{M}_1 \cdot \mathbf{S}_1 \cdot \mathbf{x} - \mathbf{e}_i + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \equiv \mathbf{M} \cdot \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \cdot \mathbf{x} + \mathbf{e}_i \bmod q, \end{aligned}$$

where $\mathbf{S}_1 := (\mathbf{S}_{1,c}, \mathbf{S}) \in \mathcal{R}_q^{t_1 \times (m_c + m)}$, so that $\mathbf{S}_1 \cdot \mathbf{x} = \mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{S} \cdot \mathbf{t}_i \bmod q$, and the last $\equiv$ holds since the discrete Gaussian over $\mathcal{R}$ is symmetric around zero. Since $\mathbf{S}_{1,c}$ and $\mathbf{S}$ are both uniform, $\mathbf{S}_1$ is uniform over $\mathcal{R}_q^{t_1 \times (m_c + m)}$, exactly as in $\text{Exp}1^0$. Thus $\mathbf{d}$ is distributed identically to the oracle answer in $\text{Exp}1^0$, and $\mathcal{B}$ perfectly simulates $\text{Exp}1^0$.

If $\mathcal{B}$ interacts with Exp^1 : In this case, $\mathbf{b}_i \leftarrow \mathcal{R}_q^{t_1}$ is uniform and independent, while $\mathbf{l}_i = \mathbf{M}_1 \bar{\mathbf{e}}_i + \mathbf{e}_i$ as before. Substituting,

$$\begin{aligned} \mathbf{d} &= \mathbf{M}_1 \cdot \mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{M}_1 \cdot \mathbf{b}_i - (\mathbf{M}_1 \cdot \bar{\mathbf{e}}_i + \mathbf{e}_i) + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \\ &= \mathbf{M}_1 \cdot (\mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{b}_i - \bar{\mathbf{e}}_i) - \mathbf{e}_i + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \\ &= \mathbf{M}_1 \cdot \mathbf{w} - \mathbf{e}_i + \mathbf{M}_2 \cdot \mathbf{S}_2 \cdot \mathbf{x} \equiv \mathbf{M} \cdot \begin{bmatrix} \mathbf{w} \\ \mathbf{S}_2 \cdot \mathbf{x} \end{bmatrix} + \mathbf{e}_i \bmod q, \end{aligned}$$

where $\mathbf{w} := \mathbf{S}_{1,c} \cdot \mathbf{c} + \mathbf{b}_i - \bar{\mathbf{e}}_i \bmod q$. Since $\mathbf{b}_i$ is uniformly random and independent across queries, so is $\mathbf{w}$, identically distributed as the freshly sampled $\mathbf{w}$ in $\text{Exp}1^1$. Thus $\mathbf{d}$ is distributed identically to the oracle answer in $\text{Exp}1^1$, and $\mathcal{B}$ perfectly simulates $\text{Exp}1^1$.

Norm bound and query count: The matrix $\mathbf{Z} = \mathbf{M}_1^T$ sent by $\mathcal{B}$ to the Exp challenger is admissible, because

$$\|\mathbf{Z}^T\|_{2,2} = \|\mathbf{M}_1\|_{2,2} \leq \gamma.$$

$\mathcal{B}$ simply forwards each TagO query, and it queries leakO exactly once per fresh index i , hence at most Q times by the assertion $|L_{\text{Query}}| < Q$ in O_b of Exp^b . $\square$

G.4 Theorem 5 and proofs

Theorem 5. Let $\mathcal{R}, n, m, Q, q, K, t, t_2, \epsilon, \bar{\chi}, \chi, \chi^*, \gamma, \beta, \mathcal{D}_A$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has ring degree φ and field expansion factor $\mathbf{c}_2$ w.r.t. the coefficient ℓ_2 -norm, $n, m, Q, K, t, t_2 \in \text{poly}(\lambda)$, $\mathcal{D}_A$ is a distribution over $\mathcal{R}_q^{n \times m}$, $0 < \epsilon \leq 1/2$ and $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$. Denote by $\mathbf{u}^T := (1, 0, \dots, 0) \in \mathcal{R}^{1 \times t}$ the first unit vector, and

$$\text{par} = (\mathcal{R}, n, m, Q, q, K, t, t_2, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_A), \quad \text{par}_{\text{LWE}} = (\mathcal{R}, n, m, q, \chi^*, \mathcal{D}_A).$$

Let $\mathbf{c}_t^\otimes := 1$ if t is a power of two and $\mathbf{c}_t^\otimes := \sqrt{t}$ otherwise. Suppose that $\gamma \cdot \sqrt{\varphi m} \cdot \beta \geq 1$ and

$$\sqrt{2QK} \cdot \varphi m \cdot \mathbf{c}_2 \cdot \mathbf{c}_t^\otimes \cdot \gamma \cdot \beta \leq \chi \cdot \sqrt{(2\eta_\epsilon(\mathcal{R}^m)^2 + (\chi^*)^2)^{-1} - \bar{\chi}^{-2}}. \quad (36)$$

For any PPT $\mathcal{A}$, define the following experiment $\text{Exp}2_{\text{par}, \mathcal{A}}^b$:

$\text{Exp2}_{\text{par},\mathcal{A}}^b(1^\lambda)$	$\text{O}_b(\mathbf{x} \in \mathcal{R}^m)$
$i := 0$	assert $i < Q$
$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}$	$i := i + 1$
$\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2) \in \mathcal{R}^{K \times (t+t_2)} \leftarrow \mathcal{A}(\mathbf{A})$	assert $\ \mathbf{x}\ _\infty \leq \beta$
assert $\ \mathbf{M}_1\ _{2,2} \leq \gamma$	$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R},\chi}^K$
$\mathbf{S}_2 \leftarrow \mathcal{R}_q^{t_2 \times n}; \bar{\mathbf{e}} \leftarrow \mathcal{D}_{\mathcal{R},\bar{\chi}}^m$	if $b = 0$ then $\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \mathbf{S}_1 \mathbf{A} \\ \mathbf{S}_2 \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e} \bmod q$
if $b = 0$ then	if $b = 1$ then $\mathbf{d} := \mathbf{M} \cdot \begin{bmatrix} \bar{\mathbf{S}}_1 \\ \mathbf{S}_2 \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e} \bmod q$
$\mathbf{S}_1 \leftarrow \mathcal{R}_q^{t \times n}$	return $\mathbf{d}$
$\mathbf{b}^T := \mathbf{u}^T \mathbf{S}_1 \mathbf{A} + \bar{\mathbf{e}}^T \bmod q$	
if $b = 1$ then	
$\bar{\mathbf{S}}_1 \leftarrow \mathcal{R}_q^{t \times m}$	
$\mathbf{b}^T := \mathbf{u}^T \bar{\mathbf{S}}_1 + \bar{\mathbf{e}}^T \bmod q$	
$b' \leftarrow \mathcal{A}^{O_b}(\mathbf{A}, \mathbf{b}, \mathbf{S}_2)$	
return b'	

and denote the advantage of $\mathcal{A}$ in distinguishing the above experiments by

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp2}} := |\Pr[\text{Exp2}_{\text{par},\mathcal{A}}^0(1^\lambda) = 1] - \Pr[\text{Exp2}_{\text{par},\mathcal{A}}^1(1^\lambda) = 1]|. \quad (37)$$

$\forall$ PPT $\mathcal{A}$, $\exists$ PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par},\mathcal{A}}^{\text{Exp2}}(\lambda) \leq t \cdot \left(\text{Adv}_{\text{parLWE},\mathcal{B}}^{\text{LWE}}(\lambda) + 4 \cdot QK \cdot \epsilon + 6 \cdot \epsilon / (1 - \epsilon) \right).$$

In particular, if $QK \in \text{poly}(\lambda)$, $\epsilon \in \text{negl}(\lambda)$ and the $\text{LWE}_{\text{parLWE}}$ assumption holds, then the experiments $\text{Exp2}_{\text{par},\mathcal{A}}^0$ and $\text{Exp2}_{\text{par},\mathcal{A}}^1$ are computationally indistinguishable.

Proof. Follows from putting Corollary 2 and Lemmas 25 to 27 together. In particular:

- By Corollary 2, invoked with the query bound QK and with its leakage parameter α instantiated as $\mathbf{c}_t^\otimes \cdot \gamma \cdot \sqrt{\varphi m} \cdot \beta$, so that its leakage set is $\mathcal{L} = \{\mathbf{z} \in \mathcal{R}^m : \|\mathbf{z}\|_2 \leq \mathbf{c}_t^\otimes \cdot \gamma \cdot \sqrt{\varphi m} \cdot \beta\}$, the $\text{LWE}_{\text{parLWE}}$ assumption implies AdpLLWE with $\text{par}^* = (\mathcal{R}, n, m, QK, q, \bar{\chi}, \chi, \mathcal{L}, \mathcal{D}_{\mathbf{A}})$, at an additive loss of $4 \cdot QK \cdot \epsilon + 6 \cdot \epsilon / (1 - \epsilon)$. Its requirements hold: $\alpha \geq 1$ follows from $\gamma \cdot \sqrt{\varphi m} \cdot \beta \geq 1$ as $\mathbf{c}_t^\otimes \geq 1$, $\chi^* \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$ is assumed, and Eq. (36) is Eq. (31) with α instantiated and with Q replaced by QK , which suffices by Remark 9.
- By Lemma 25, AdpLLWE with par^* implies AdpLLWE with $\text{par}^\dagger = (\mathcal{R}, nt, mt, QK, q, \bar{\chi}, \chi, \mathcal{L}^\dagger, \mathbf{I}_t \otimes \mathcal{D}_{\mathbf{A}})$, where $\mathcal{L}^\dagger = \{\mathbf{z} \in \mathcal{R}^{mt} : \|\mathbf{z}\|_2 \leq \mathbf{c}_t^\otimes \cdot \gamma \cdot \sqrt{\varphi m} \cdot \beta\}$ is the leakage set of Lemma 26, at a loss of a factor t .
- By Lemma 26, AdpLLWE with $\text{par}^\dagger$ implies the indistinguishability of the experiments Exp3^0 and Exp3^1 defined therein, and by Lemma 27 the latter implies the claimed indistinguishability. Both steps are tight.

Putting the advantage bounds in the above lemmas together yields the stated bound. $\square$

Lemma 25 below is a standard hybrid argument, reducing adaptive leaky LWE for matrix $\mathbf{A}$ to that for matrix $\mathbf{I} \otimes \mathbf{A}$. It is a direct adaption of [CLW25, Proposition 1, Eprint].

Lemma 25. Let $\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{D}_{\mathbf{A}}$ be parametrised by λ , where $\mathcal{D}_{\mathbf{A}}$ is a distribution over $\mathcal{R}_q^{n \times m}$. Let $t \in \text{poly}(\lambda)$, $\mathcal{L} = \{\mathbf{z} \in \mathcal{R}^m : \|\mathbf{z}\|_2 \leq \alpha\}$ and $\mathcal{L}^\dagger = \{\mathbf{z} \in \mathcal{R}^{mt} : \|\mathbf{z}\|_2 \leq \alpha\}$ for some $\alpha > 0$. Let

$$\text{par}^* = (\mathcal{R}, n, m, Q, q, \bar{\chi}, \chi, \mathcal{L}, \mathcal{D}_{\mathbf{A}}), \quad \text{par}^\dagger = (\mathcal{R}, nt, mt, Q, q, \bar{\chi}, \chi, \mathcal{L}^\dagger, \mathbf{I}_t \otimes \mathcal{D}_{\mathbf{A}}).$$

For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}^\dagger, \mathcal{A}}^{\text{AdpLLWE}}(\lambda) \leq t \cdot \text{Adv}_{\text{par}^*, \mathcal{B}}^{\text{AdpLLWE}}(\lambda).$$

In particular, if the $\text{AdpLLWE}_{\text{par}^*}$ assumption holds, then the $\text{AdpLLWE}_{\text{par}^\dagger}$ assumption holds.

Proof. For each $i \in \{0, 1, \dots, t\}$, define the following i -th hybrid experiment:

$\text{Exp}_i(1^\lambda)$	$\text{O}(\mathbf{z})$
$\mathbf{A} \leftarrow \mathcal{D}_{\mathbf{A}}; j := 0$	assert $j < Q$
for $k \in [t]$ do $\bar{\mathbf{e}}_k \leftarrow \$ D_{\mathcal{R}, \bar{\chi}}^m$	$j := j + 1$
for $k \in [1, \dots, i]$ do $\mathbf{b}_k \leftarrow \$ \mathcal{R}_q^m$	assert $\mathbf{z} \in \mathcal{L}^\dagger$
for $k \in [i+1, \dots, t]$ do $\mathbf{r}_k \leftarrow \$ \mathcal{R}_q^n; \mathbf{b}_k^\top := \mathbf{r}_k^\top \mathbf{A} + \bar{\mathbf{e}}_k^\top \bmod q$	parse $\begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_t \end{pmatrix} \leftarrow \mathbf{z} \quad // \mathbf{z}_k \in \mathcal{R}^m \forall k \in [t]$
$\mathbf{b}^\top := (\mathbf{b}_1^\top, \dots, \mathbf{b}_t^\top); \bar{\mathbf{e}}^\top := (\bar{\mathbf{e}}_1^\top, \dots, \bar{\mathbf{e}}_t^\top)$	$e \leftarrow \$ D_{\mathcal{R}, \chi}; \ell := \sum_{k \in [t]} \bar{\mathbf{e}}_k^\top \cdot \mathbf{z}_k + e$
return $\mathcal{A}^{\text{O}}(\mathbf{A}, \mathbf{b})$	return ℓ

It is direct to verify that the 0-th hybrid is identical to the $\text{AdpLLWE}_{\text{par}^\dagger}$ experiment 0, as

$$\mathbf{b}^\top = (\mathbf{r}_1^\top \mathbf{A} + \bar{\mathbf{e}}_1^\top, \dots, \mathbf{r}_t^\top \mathbf{A} + \bar{\mathbf{e}}_t^\top) = (\mathbf{r}_1^\top, \dots, \mathbf{r}_t^\top)(\mathbf{I}_t \otimes \mathbf{A}) + \bar{\mathbf{e}}^\top \bmod q,$$

and each oracle answer is

$$\ell = \sum_{k \in [t]} \bar{\mathbf{e}}_k^\top \cdot \mathbf{z}_k + e = \bar{\mathbf{e}}^\top \cdot \mathbf{z} + e.$$

Also, the t -th hybrid is identical to the $\text{AdpLLWE}_{\text{par}^\dagger}$ experiment 1. In the rest we show the $(i-1)$ -th and i -th hybrids are computationally indistinguishable under the $\text{AdpLLWE}_{\text{par}^*}$ assumption, for each $i \in [t]$.

Fix any $i \in [t]$. Given any PPT $\mathcal{A}$ against distinguishing the $(i-1)$ -th and i -th hybrids, we construct a PPT $\mathcal{B}$ against $\text{AdpLLWE}_{\text{par}^*}$. Let $\mathcal{B}$ proceed as follows:

- Receive $(\mathbf{A}, \mathbf{b}_i)$ from the $\text{AdpLLWE}_{\text{par}^*}$ challenger.
- For each $k \in [1, \dots, t] \setminus \{i\}$, sample $\bar{\mathbf{e}}_k \leftarrow \$ D_{\mathcal{R}, \bar{\chi}}^m$.
- For each $k \in [1, \dots, i-1]$, sample random $\mathbf{b}_k \leftarrow \$ \mathcal{R}_q^m$.
- For each $k \in [i+1, \dots, t]$, sample random $\mathbf{r}_k \leftarrow \$ \mathcal{R}_q^n$, set $\mathbf{b}_k^\top := \mathbf{r}_k^\top \mathbf{A} + \bar{\mathbf{e}}_k^\top \bmod q$.
- Set $\mathbf{b}^\top := (\mathbf{b}_1^\top, \dots, \mathbf{b}_t^\top)$. Send $(\mathbf{A}, \mathbf{b})$ to $\mathcal{A}$.
- Internally maintain a counter $j := 0$. For each oracle query $\text{O}(\mathbf{z})$ made by $\mathcal{A}$, do the following:
 - Assert $j < Q$ and set $j := j + 1$.
 - Assert $\mathbf{z} \in \mathcal{L}^\dagger$. Parse $\begin{pmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_t \end{pmatrix} \leftarrow \mathbf{z}$.
 - For each $k \in [1, \dots, t] \setminus \{i\}$, let $\ell_k := \bar{\mathbf{e}}_k^\top \cdot \mathbf{z}_k$.
 - Send $\mathbf{z}_i$ to the oracle LeakO available from the $\text{AdpLLWE}_{\text{par}^*}$ challenger, receive from which ℓ_i .
 - Answer with $\ell := \sum_{k \in [t]} \ell_k$.
- Eventually receive a bit b' from $\mathcal{A}$. Return b' .

It is direct to see that $\mathcal{B}$ perfectly simulates the view of the $(i-1)$ -th and the i -th hybrid to $\mathcal{A}$, if it interacts with the $\text{AdpLLWE}_{\text{par}^*}$ experiment 0 and 1, respectively. In particular, in either case, $\ell_i = \bar{\mathbf{e}}_i^\top \cdot \mathbf{z}_i + e$ for error $e \leftarrow \$ D_{\mathcal{R}, \chi}$, hence the simulated oracle answer is

$$\ell = \sum_{k \in [t]} \ell_k = \sum_{k \in [t]} \bar{\mathbf{e}}_k^\top \cdot \mathbf{z}_k + e.$$

The vector $\mathbf{z}_i$ sent from $\mathcal{B}$ to the $\text{AdpLLWE}_{\text{par}^*}$ challenger is admissible, i.e. $\mathbf{z}_i \in \mathcal{L}$, since $\mathbf{z} \in \mathcal{L}^\dagger$ implies $\|\mathbf{z}\|_2 \leq \alpha$, and we have $\|\mathbf{z}_i\|_2 \leq \|\mathbf{z}\|_2 \leq \alpha$. $\square$

Lemma 26 below restates AdpLLWE w.r.t. $\mathbf{I} \otimes \mathbf{A}$ in the matrix form used in the sequel, in which the secret is a matrix and the leakage on the error matrix is a noisy bilinear form.

Lemma 26. *Let $\mathcal{R}, n, m, Q, q, K, t, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_{\mathbf{A}}$ be parametrised by λ , where $\mathcal{R} = \mathbb{Z}[\zeta]$ has conductor $\mathfrak{f}$ and ring degree φ , $n, m, Q, K, t \in \text{poly}(\lambda)$ and $\mathcal{D}_{\mathbf{A}}$ is a distribution over $\mathcal{R}_q^{n \times m}$. Denote*

$$\alpha := \mathbf{c}_t^\otimes \cdot \gamma \cdot \sqrt{\varphi m} \cdot \beta,$$

$$\text{par}_E = (\mathcal{R}, n, m, Q, q, K, t, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_A), \quad \text{par}^\dagger = (\mathcal{R}, nt, mt, QK, q, \bar{\chi}, \chi, \mathcal{L}^\dagger, \mathbf{I}_t \otimes \mathcal{D}_A),$$

where $\mathfrak{c}_t^\otimes := 1$ if $\mathfrak{f}$ is a power of two and $\mathfrak{c}_t^\otimes := \sqrt{t}$ otherwise, and $\mathcal{L}^\dagger = \{\mathbf{z} \in \mathcal{R}^{mt} : \|\mathbf{z}\|_2 \leq \alpha\}$. For any PPT $\mathcal{A}$, define the following experiment $\text{Exp3}_{\text{par}_E, \mathcal{A}}^b$:

$\text{Exp3}_{\text{par}_E, \mathcal{A}}^b(1^\lambda)$	$\text{LeakO}(\mathbf{Y} \in \mathcal{R}^{K \times t}, \mathbf{x} \in \mathcal{R}^m)$
$i := 0$	assert $i < Q$
$\mathbf{A} \leftarrow \mathcal{D}_A; \bar{\mathbf{E}} \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^{t \times m}$	$i := i + 1$
if $b = 0$ then	assert $\ \mathbf{Y}\ _{2,2} \leq \gamma \wedge \ \mathbf{x}\ _\infty \leq \beta$
$\mathbf{S} \leftarrow \mathcal{R}_q^{t \times n}; \mathbf{B} := \mathbf{S}\mathbf{A} + \bar{\mathbf{E}} \bmod q$	$\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$
if $b = 1$ then	$\mathbf{l} := \mathbf{Y}\bar{\mathbf{E}}\mathbf{x} + \mathbf{e} \bmod q$
$\mathbf{B} \leftarrow \mathcal{R}_q^{t \times m}$	return $\mathbf{l}$
return $\mathcal{A}^{\text{LeakO}}(\mathbf{A}, \mathbf{B})$	

and denote the advantage of $\mathcal{A}$ in distinguishing the above experiments by

$$\text{Adv}_{\text{par}_E, \mathcal{A}}^{\text{Exp3}} := \left| \Pr \left[\text{Exp3}_{\text{par}_E, \mathcal{A}}^0(1^\lambda) = 1 \right] - \Pr \left[\text{Exp3}_{\text{par}_E, \mathcal{A}}^1(1^\lambda) = 1 \right] \right|.$$

For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}_E, \mathcal{A}}^{\text{Exp3}}(\lambda) = \text{Adv}_{\text{par}^\dagger, \mathcal{B}}^{\text{AdpLLWE}}(\lambda).$$

In particular, if the $\text{AdpLLWE}_{\text{par}^\dagger}$ assumption holds, then the experiments $\text{Exp3}_{\text{par}_E, \mathcal{A}}^0$ and $\text{Exp3}_{\text{par}_E, \mathcal{A}}^1$ are computationally indistinguishable.

Proof. Given a PPT $\mathcal{A}$ distinguishing Exp3^0 and Exp3^1 , we construct a PPT $\mathcal{B}$ against $\text{AdpLLWE}_{\text{par}^\dagger}$, as follows:

- Receive $(\mathbf{I}_t \otimes \mathbf{A}, \bar{\mathbf{b}}) \in \mathcal{R}_q^{nt \times mt} \times \mathcal{R}_q^{mt}$ from the AdpLLWE challenger. Parse $\bar{\mathbf{b}}^\top = (\bar{\mathbf{b}}_1^\top, \dots, \bar{\mathbf{b}}_t^\top)$ with $\bar{\mathbf{b}}_j \in \mathcal{R}_q^m$ for $j \in [t]$, set $\mathbf{B} \in \mathcal{R}_q^{t \times m}$ to be the matrix whose j -th row is $\bar{\mathbf{b}}_j^\top$, and pass $(\mathbf{A}, \mathbf{B})$ to $\mathcal{A}$.
- For each query $(\mathbf{Y}, \mathbf{x})$ made by $\mathcal{A}$:
 - Assert $i < Q$ and set $i := i + 1$. Assert $\|\mathbf{Y}\|_{2,2} \leq \gamma$ and $\|\mathbf{x}\|_\infty \leq \beta$.
 - Let $\mathbf{y}_k^\top$ denote the k -th row of $\mathbf{Y}$ and $\mathbf{z}_k := \mathbf{y}_k \otimes \mathbf{x} \in \mathcal{R}^{mt}$, i.e. $\mathbf{z}_k$ consists of t blocks of length m , the j -th being $y_{k,j} \cdot \mathbf{x}$.
 - For each $k \in [K]$, query the oracle $\text{LeakO}(\mathbf{z}_k)$ available from the AdpLLWE challenger, and receive an answer ℓ_k . Answer $\mathcal{A}$ with $\mathbf{l} := (\ell_1, \dots, \ell_K)^\top$.
- Eventually receive b' from $\mathcal{A}$, return b' .

Write $\bar{\mathbf{e}}^\top = (\bar{\mathbf{e}}_1^\top, \dots, \bar{\mathbf{e}}_t^\top)$, where each $\bar{\mathbf{e}}_j \in \mathcal{R}^m$, for the error of LWE samples in the AdpLLWE experiment, and let $\bar{\mathbf{E}} \in \mathcal{R}^{t \times m}$ be the matrix whose j -th row is $\bar{\mathbf{e}}_j^\top$.

In the AdpLLWE experiment 0, we have $\bar{\mathbf{b}}^\top = \mathbf{r}^\top(\mathbf{I}_t \otimes \mathbf{A}) + \bar{\mathbf{e}}^\top \bmod q$ for a uniform $\mathbf{r} \leftarrow \mathcal{R}_q^{nt}$. Writing $\mathbf{r}^\top = (\mathbf{r}_1^\top, \dots, \mathbf{r}_t^\top)$ where each $\mathbf{r}_j \in \mathcal{R}_q^n$, then we have

$$\bar{\mathbf{b}}_j^\top = \mathbf{r}_j^\top \mathbf{A} + \bar{\mathbf{e}}_j^\top \bmod q \quad \forall j \in [t], \quad \text{equivalently} \quad \mathbf{B} = \mathbf{S}\mathbf{A} + \bar{\mathbf{E}} \bmod q$$

where $\mathbf{S} \in \mathcal{R}_q^{t \times n}$ is uniform, whose j -th row is $\mathbf{r}_j^\top$. In the AdpLLWE experiment 1, the matrix $\mathbf{B}$ is uniform. In either case, $\bar{\mathbf{E}} \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^{t \times m}$, as in Exp3 .

For an oracle query $(\mathbf{Y}, \mathbf{x})$, the answer is $\mathbf{l} = (\ell_1, \dots, \ell_K)^\top$ where, for each $k \in [K]$,

$$\ell_k = \bar{\mathbf{e}}^\top \cdot \mathbf{z}_k + e_k = \sum_{j \in [t]} \bar{\mathbf{e}}_j^\top \cdot (y_{k,j} \cdot \mathbf{x}) + e_k = \sum_{j \in [t]} y_{k,j} \cdot (\bar{\mathbf{e}}_j^\top \mathbf{x}) + e_k = \mathbf{y}_k^\top \bar{\mathbf{E}} \mathbf{x} + e_k,$$

with $e_k \leftarrow \mathcal{D}_{\mathcal{R}, \chi}$. Vertically concatenating, we have $\mathbf{l} = \mathbf{Y}\bar{\mathbf{E}}\mathbf{x} + \mathbf{e}$ as in Exp3 .

Hence, $\mathcal{B}$ perfectly simulates Exp3^0 , resp. Exp3^1 , when interacting with the AdpLLWE experiment 0, resp. experiment 1. Note that the K answers $\ell_1, \dots, \ell_K$ have independent errors even if $\mathbf{Y}$ has two equal rows, since LeakO in Fig. 4 samples a fresh error on every query.

Query count: $\mathcal{B}$ makes K queries to LeakO per query of $\mathcal{A}$, hence at most QK in total by the assertion $i < Q$ in LeakO of Exp3^b .

Norm bound: It remains to check that the queries of $\mathcal{B}$ satisfies the norm bound constraint. For each $k \in [K]$, the entries of $\mathbf{z}_k$ are $y_{k,j} \cdot x_i$ for $(j, i) \in [t] \times [m]$. Regrouping them by the columns of $\mathbf{Y}$, and writing $\mathbf{u}_j \in \mathcal{R}^t$ for the j -th unit vector,

$$\sum_{k \in [K]} \|\mathbf{z}_k\|_2^2 = \sum_{i \in [m]} \sum_{j \in [t]} \|\mathbf{Y} \cdot (x_i \cdot \mathbf{u}_j)\|_2^2 \leq \sum_{i \in [m]} \sum_{j \in [t]} \|\mathbf{Y}\|_{2,2}^2 \cdot \|x_i\|_2^2 = t \cdot \|\mathbf{Y}\|_{2,2}^2 \cdot \|\mathbf{x}\|_2^2,$$

hence, for every $k \in [K]$,

$$\begin{aligned} \|\mathbf{z}_k\|_2 &\leq \sqrt{t} \cdot \|\mathbf{Y}\|_{2,2} \cdot \|\mathbf{x}\|_2 \\ &\leq \sqrt{t} \cdot \gamma \cdot \sqrt{\varphi m} \cdot \|\mathbf{x}\|_\infty \\ &\leq \sqrt{t} \cdot \gamma \cdot \sqrt{\varphi m} \cdot \beta. \end{aligned}$$

If $\mathfrak{f}$ is a power of two, the factor $\sqrt{t}$ in the above bound can be dropped, as follows. The idea is to measure $\mathbf{z}_k$ in the canonical embedding, where the product of two field elements has multiplicative norm, and then to relate the resulting bound back to $\|\mathbf{Y}\|_{2,2}$. We use the following three facts.

- (i) Extend each canonical embedding σ_ι of $\mathcal{K}$ (Section 3) entrywise to vectors and matrices over $\mathcal{K}$ (so that e.g. $\sigma_\iota(\mathbf{Y}) \in \mathbb{C}^{K \times t}$). Since σ_ι is a ring homomorphism, $\sigma_\iota(\mathbf{M}\mathbf{v}) = \sigma_\iota(\mathbf{M}) \cdot \sigma_\iota(\mathbf{v})$ and $\sigma_\iota(\mathbf{a} \otimes \mathbf{w}) = \sigma_\iota(\mathbf{a}) \otimes \sigma_\iota(\mathbf{w})$ for any matrix $\mathbf{M}$ and vectors $\mathbf{v}, \mathbf{a}, \mathbf{w}$ over $\mathcal{K}$ of compatible dimensions.
- (ii) As $\mathfrak{f}$ is a power of two, σ is a scaled ℓ_2 -isometry, $\|\sigma(w)\|_2 = \sqrt{\varphi} \cdot \|w\|_2$ for any $w \in \mathcal{K}$ (Section 3), hence $\|\mathbf{v}\|_2^2 = \frac{1}{\varphi} \sum_\iota \|\sigma_\iota(\mathbf{v})\|_2^2$ for any vector $\mathbf{v}$ over $\mathcal{K}$.
- (iii) By Lemma 2, $\max_\iota \|\sigma_\iota(\mathbf{M})\|_{2,2} = \|\mathbf{M}\|_{2,2}$ for any matrix $\mathbf{M}$ over $\mathcal{K}$.

Given these facts, for each $k \in [K]$,

$$\begin{aligned} \|\mathbf{z}_k\|_2^2 &= \frac{1}{\varphi} \sum_\iota \|\sigma_\iota(\mathbf{z}_k)\|_2^2 \quad // \text{ by (ii)} \\ &= \frac{1}{\varphi} \sum_\iota \|\sigma_\iota(\mathbf{y}_k)\|_2^2 \cdot \|\sigma_\iota(\mathbf{x})\|_2^2 \quad // \text{ by (i) and } \|\mathbf{a} \otimes \mathbf{w}\|_2 = \|\mathbf{a}\|_2 \cdot \|\mathbf{w}\|_2 \\ &\leq \max_\iota \|\sigma_\iota(\mathbf{Y})\|_{2,2}^2 \cdot \frac{1}{\varphi} \sum_\iota \|\sigma_\iota(\mathbf{x})\|_2^2 \\ &= \max_\iota \|\sigma_\iota(\mathbf{Y})\|_{2,2}^2 \cdot \|\mathbf{x}\|_2^2 \quad // \text{ by (ii)} \\ &\leq \|\mathbf{Y}\|_{2,2}^2 \cdot \|\mathbf{x}\|_2^2 \quad // \text{ by (iii)} \end{aligned}$$

where the first inequality holds because $\sigma_\iota(\mathbf{y}_k)^T$ is the k -th row of $\sigma_\iota(\mathbf{Y})$ and the ℓ_2 -norm of a row of a complex matrix is at most its ℓ_2 -induced norm. Taking square roots and using $\|\mathbf{Y}\|_{2,2} \leq \gamma$ and $\|\mathbf{x}\|_2 \leq \sqrt{\varphi m} \cdot \beta$ as before yields

$$\|\mathbf{z}_k\|_2 \leq \|\mathbf{Y}\|_{2,2} \cdot \|\mathbf{x}\|_2 \leq \gamma \cdot \sqrt{\varphi m} \cdot \beta,$$

i.e. the bound of the general case but without the factor $\sqrt{t}$. □

Lemma 27 below gives a reduction from Exp3 to the distinguishing problem defined in Theorem 5.

Lemma 27. *Let $\mathcal{R}, n, m, Q, q, K, t, t_2, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_\mathbf{A}$ be parametrised by λ , where $\mathcal{D}_\mathbf{A}$ is a distribution over $\mathcal{R}_q^{n \times m}$. Denote*

$$\text{par} = (\mathcal{R}, n, m, Q, q, K, t, t_2, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_\mathbf{A}), \quad \text{par}_\mathbf{E} = (\mathcal{R}, n, m, Q, q, K, t, \bar{\chi}, \chi, \gamma, \beta, \mathcal{D}_\mathbf{A}).$$

Consider the experiment $\text{Exp2}_{\text{par}, \mathcal{A}}^b$ defined in Theorem 5 and the advantage $\text{Adv}_{\text{par}, \mathcal{A}}^{\text{Exp2}}$ defined in Eq. (37), and let Exp3 denote the experiment of Lemma 26 with parameters $\text{par}_\mathbf{E}$. For any PPT $\mathcal{A}$, there exists a PPT $\mathcal{B}$ such that

$$\text{Adv}_{\text{par}, \mathcal{A}}^{\text{Exp2}}(\lambda) = \text{Adv}_{\text{par}_\mathbf{E}, \mathcal{B}}^{\text{Exp3}}(\lambda).$$

Proof. Given a PPT $\mathcal{A}$ distinguishing Exp2^0 and Exp2^1 , we construct a PPT $\mathcal{B}$ distinguishing Exp3^0 and Exp3^1 , as follows:

- Receive $(\mathbf{A}, \mathbf{B}) \in \mathcal{R}_q^{n \times m} \times \mathcal{R}_q^{t \times m}$ from the Exp3 challenger.
- Pass $\mathbf{A}$ to $\mathcal{A}$, and receive from which $\mathbf{M} = (\mathbf{M}_1, \mathbf{M}_2) \in \mathcal{R}^{K \times (t+t_2)}$. Assert $\|\mathbf{M}_1\|_{2,2} \leq \gamma$.
- Sample $\mathbf{S}_2 \leftarrow \mathcal{R}_q^{t_2 \times n}$, set $\mathbf{b}^\top := \mathbf{u}^\top \mathbf{B} \bmod q$ where $\mathbf{u}^\top := (1, 0, \dots, 0) \in \mathcal{R}^{1 \times t}$ is the first unit vector, and pass $(\mathbf{A}, \mathbf{b}, \mathbf{S}_2)$ to $\mathcal{A}$.
- Maintain a counter $i := 0$. Answer each oracle query $\text{O}_b(\mathbf{x})$ of $\mathcal{A}$ as follows: assert $i < Q$, set $i := i + 1$, and assert $\|\mathbf{x}\|_\infty \leq \beta$. Then query $\text{LeakO}(\mathbf{M}_1, \mathbf{x})$ of Exp3 to obtain $\mathbf{l}$, compute

$$\mathbf{d} := \mathbf{M}_1 \cdot \mathbf{B}\mathbf{x} - \mathbf{l} + \mathbf{M}_2 \cdot \mathbf{S}_2 \mathbf{A} \mathbf{x} \bmod q,$$

and answer with $\mathbf{d}$.

- Eventually receive a bit b' from $\mathcal{A}$, return b' .

Below we show that $\mathcal{B}$ perfectly simulates the view of either Exp2^0 or Exp2^1 to $\mathcal{A}$, if it interacts respectively with Exp3^0 or Exp3^1 , thus the lemma follows.

In both cases, write $\mathbf{W} := \mathbf{B} - \bar{\mathbf{E}} \bmod q$ for the error matrix $\bar{\mathbf{E}} \leftarrow \mathcal{D}_{\mathcal{R}, \bar{\chi}}^{t \times m}$ of Exp3 , and $\bar{\mathbf{e}}^\top := \mathbf{u}^\top \bar{\mathbf{E}}$ for its first row, which follows $\mathcal{D}_{\mathcal{R}, \bar{\chi}}^m$. By definition of the LeakO answer in Exp3 , we have $\mathbf{l} = \mathbf{M}_1 \cdot \bar{\mathbf{E}}\mathbf{x} + \mathbf{e}$ for a fresh $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}, \chi}^K$, hence for each query $\mathbf{x}$ the simulated oracle answer is

$$\begin{aligned} \mathbf{d} &= \mathbf{M}_1 \cdot \mathbf{B}\mathbf{x} - (\mathbf{M}_1 \cdot \bar{\mathbf{E}}\mathbf{x} + \mathbf{e}) + \mathbf{M}_2 \cdot \mathbf{S}_2 \mathbf{A} \mathbf{x} \\ &= \mathbf{M}_1 \cdot \mathbf{W}\mathbf{x} - \mathbf{e} + \mathbf{M}_2 \cdot \mathbf{S}_2 \mathbf{A} \mathbf{x} \equiv \mathbf{M} \cdot \begin{bmatrix} \mathbf{W} \\ \mathbf{S}_2 \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e} \bmod q, \end{aligned}$$

where the last $\equiv$ holds since the discrete Gaussian over $\mathcal{R}$ is symmetric around zero. Moreover, the vector simulated to $\mathcal{A}$ is $\mathbf{b}^\top = \mathbf{u}^\top \mathbf{B} = \mathbf{u}^\top \mathbf{W} + \bar{\mathbf{e}}^\top \bmod q$. It remains to identify $\mathbf{W}$ in the two cases.

If $\mathcal{B}$ interacts with Exp3^0 : Here $\mathbf{B} = \mathbf{S}\mathbf{A} + \bar{\mathbf{E}}$ for a uniform $\mathbf{S} \leftarrow \mathcal{R}_q^{t \times n}$, so $\mathbf{W} = \mathbf{S}\mathbf{A}$. Thus $\mathbf{b}^\top = \mathbf{u}^\top \mathbf{S}\mathbf{A} + \bar{\mathbf{e}}^\top$ and $\mathbf{d} \equiv \mathbf{M} \cdot \begin{bmatrix} \mathbf{S}\mathbf{A} \\ \mathbf{S}_2 \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e}$, which is the view of Exp2^0 with $\mathbf{S}_1 := \mathbf{S}$.

If $\mathcal{B}$ interacts with Exp3^1 : Here $\mathbf{B} \leftarrow \mathcal{R}_q^{t \times m}$ is uniform and independent of $\bar{\mathbf{E}}$, so $\mathbf{W} = \mathbf{B} - \bar{\mathbf{E}}$ is uniform over $\mathcal{R}_q^{t \times m}$ and independent of $\bar{\mathbf{E}}$. Thus $\mathbf{b}^\top = \mathbf{u}^\top \mathbf{W} + \bar{\mathbf{e}}^\top$ and $\mathbf{d} \equiv \mathbf{M} \cdot \begin{bmatrix} \mathbf{W} \\ \mathbf{S}_2 \mathbf{A} \end{bmatrix} \mathbf{x} + \mathbf{e}$, which is the view of Exp2^1 with $\bar{\mathbf{S}}_1 := \mathbf{W}$.

Norm bound and query count: The tuple $(\mathbf{M}_1, \mathbf{x})$ sent by $\mathcal{B}$ to the Exp3 challenger is admissible, since $\|\mathbf{M}_1\|_{2,2} \leq \gamma$ and $\|\mathbf{x}\|_\infty \leq \beta$ are exactly the assertions of Exp2^b . Finally, $\mathcal{B}$ queries LeakO exactly once per query of $\mathcal{A}$, hence at most Q times by the assertion $i < Q$ in O_b of Exp2^b . $\square$