Simple Threshold (Fully Homomorphic) Encryption From LWE With Polynomial Modulus

Katharina Boudgoust and Peter Scholl

katharina.boudgoust@cs.au.dk, peter.scholl@cs.au.dk

Aarhus University

Abstract. The learning with errors (LWE) assumption is a powerful tool for building encryption schemes with useful properties, such as plausible resistance to quantum computers, or support for homomorphic computations. Despite this, essentially the only method of achieving threshold decryption in schemes based on LWE requires a modulus that is superpolynomial in the security parameter, leading to a large overhead in ciphertext sizes and computation time.

In this work, we propose a (fully homomorphic) encryption scheme that supports a simple t-out-of-n threshold decryption protocol while allowing for a polynomial modulus. The main idea is to use the Rényi divergence (as opposed to the statistical distance as in previous works) as a measure of distribution closeness. This comes with some technical obstacles, due to the difficulty of using the Rényi divergence in decisional security notions such as standard semantic security. We overcome this by constructing a threshold scheme with a weaker notion of one-way security and then showing how to transform any one-way threshold scheme into one guaranteeing semantic security.

1 Introduction

In a public key encryption (PKE) scheme, one needs the secret key sk to decrypt an encrypted message. Giving one single party control of the whole secret key can be seen as a single point of failure. The study of PKE with threshold decryption aims to mitigate this by splitting the secret key into n key shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$, such that several key shares are needed to be able to decrypt ciphertexts. In the common t-out-of-n setting, any set of t parties or fewer learns no information about encrypted messages, while any set of $t + 1$ parties can jointly decrypt ciphertexts. To decrypt, the parties first compute their own partial decryption shares and then combine them together to recover the encrypted message. When $t = n - 1$, we call it full-threshold decryption.

Recently, NIST announced the standardization of the first cryptosystems to provide security even in the presence of quantum computers.[1] Among the finalists to be standardized, a majority base their security on the presumed hardness of (structured) lattice problems, such as Dilithium [Lyu+20] and Kyber [Sch+20]

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

based on the (module) learning with errors problem (Module-LWE) [LS15]. NIST also just begun a project on threshold cryptography,[2] which aims to produce guidelines and recommendations for implementing threshold cryptosystems.

It is thus a very important research question to study the possibility of thresholdizing lattice-based PKE schemes. This line of research has been initiated by [BD10], where they proposed a threshold key generation and decryption starting from Regev's encryption scheme [Reg05]. To split the secret key they use replicated secret sharing, which has a complexity that scales with $\binom{n}{t}$. Later, it has been shown that we can even build full-threshold decryption for fully-homomorphic encryption (FHE) schemes [Ash+12]. Their results have then been extended to t-out-of-n threshold and other access structures [Bon+18].

All works above have in common that they use a technique called *noise flooding* to guarantee that partial decryption shares don't leak any information on the underlying secret key. More precisely, each party first computes a "noiseless" partial decryption of a ciphertext using their secret key share. The noiseless partial decryptions allow recovering the message, but also reveal a small noise term e_{ct} that depends on the given ciphertext and the secret key. To prevent this leakage, every party locally adds some fresh noise on their decryption share before they jointly combine the necessary number of shares to recover the message. After decryption, the revealed noise term becomes $e_{\mathsf{ct}} + e'$, where $e' \leftarrow \mathcal{D}_{\mathsf{flood}}$ is a noise term that is hidden to the adversary. When proving security, the real partial decryption shares are replaced by simulated ones which do not depend on the secret key, and instead reveal noise terms of the form $e' \leftarrow \mathcal{D}_{\mathsf{flood}}$. By arguing that the statistical distance between both ways of deriving partial decryption shares is negligible, one can argue security. While this approach has the advantage of being rather simple, it has the drawback of requiring the ratio between the flooding noise and the size of the ciphertext noise e_{ct} to be superpolynomial in the security parameter. This in turn requires the LWE problem to be secure with a superpolynomial modulus-to-noise ratio, which weakens security and requires larger LWE parameters to compensate.

Recently, multi-party reusable non-interactive secure computation (MrNISC) was constructed from LWE with a polynomial modulus [Ben+21; Shi22]. This leads to a construction of full-threshold (multi-key) FHE with a polynomial modulus. It seems plausible that their construction can also be extended to build t-out-of-n threshold FHE with polynomial modulus; however, their techniques are very complex, due to a non-black-box "round-collapsing" technique based on garbled circuits, so unlikely to be practical. We thus started our work asking the following research question:

> Is it possible to construct a fully-homomorphic encryption scheme that supports a *simple* t-out-of n threshold decryption while allowing for a *polynomial modulus*?

[2] https://csrc.nist.gov/Projects/threshold-cryptography

Our Results. We give a positive answer to this question. From a high level perspective, we show that the simple threshold decryption technique from previous works [BD10; Bon+18] can be significantly improved by replacing the noise flooding analysis with respect to the statistical distance by one with respect to the Rényi divergence.

Doing so comes with the benefit of only requiring a polynomial ratio between ciphertext and flooding noise, hence allowing for the desired polynomial modulus. However, it does come at the expense of two new obstacles. First, the Rényi divergence fits well in search-based security notions, such as OW-CPA security[3], but does not work well with decision-based security notions, such as the standard IND-CPA security. Second, the resulting parameters now depend on the number of partial decryption queries made by an adversary within the security game. More precisely, the modulus q scales as $O(\sqrt{\ell})$, where ℓ is the number of queries, and thus a polynomial modulus can only be achieved if we restrict the number of partial decryption queries to be polynomially-bounded in advance.

To overcome the first technical challenge, we introduce the notion of OW-CPA security for threshold encryption schemes, and provide two different ways to transform a OW-CPA threshold scheme into one that guarantees IND-CPA security.[4] Whereas the first transformation only applies to standard PKE and is in the random oracle model, it comes with the advantage of guaranteeing a form of *robustness* against up to t malicious parties, with no extra cost. The second transformation is in the standard model and also applies to the fully-homomorphic setting, but does not give robustness.

Concurrent work. Concurrently to our work, [Cho+22] has used the Rényi divergence to obtain threshold FHE from LWE with a polynomial modulus-to-noise ratio, similarly to our result. By arguing that the public sampleability property applies in their setting, they directly used the Rényi divergence to prove IND-CPA security. Note that their work focuses on a specific construction of TFHE based on Torus-FHE, whereas our results are phrased generically for all encryption schemes with nearly linear decryption. Also, they focus on linear integer secret sharing schemes, whereas we additionally propose pseudorandom secret sharing and different ways of achieving robustness.

Related Work. The Rényi divergence has seen widespread use in security proofs in lattice-based cryptography, since [Bai+18]. Replacing statistical noise flooding by Rényi noise flooding has led to a significant improvement in parameters for security reductions, for instance when proving the hardness of (structured) LWE with a binary secret [Bou+20] or more recently, in the context of lattice-based threshold signatures [ASY22]. The latter work of [ASY22] is quite similar to ours,

[3] OW-CPA security for PKE says that given the public key and an encryption of a random message m, it is hard to guess m.

[4] For the case of TFHE, our definition of IND-CPA is slightly weaker than previous notions, which require partial decryptions to be *statistically simulatable*. Our game-based notion still captures the security properties needed in most applications.

since they also apply Rényi noise flooding to threshold FHE; however, they do not directly prove security of the threshold FHE scheme, and instead analyze the resulting threshold signature scheme directly (which is based on a search problem, so amenable to a Rényi divergence analysis). They additionally show the optimality of their noise flooding by providing an attack when a smaller noise flooding ratio is used. As the attack uses that their signature scheme is deterministic, it does not (directly) apply to our randomized encryption scheme.

In an independent line of work, another noise flooding technique, called gentle noise flooding, has been studied in order to avoid the superpolynomial parameter blow-up [BD20a]. It was first used in theoretical hardness results on entropic (structured) LWE [BD20a; BD20b]. Later, a similar technique was used in [Cas+22] for improving parameters in additively homomorphic encryption with circuit privacy. The setting of [Cas+22] is quite different to ours, however, since with circuit privacy, the challenge is to deal with leakage on a plaintext rather than the secret key. This is handled via gentle noise flooding by applying a randomized encoding to the plaintext, so that leaking a constant fraction of its coordinates does not reveal anything about the plaintext. A similar technique does not seem to work in the threshold setting, with leakage on the secret key.

Another approach to build threshold key generation and decryption protocols is to use general multi-party computation tools like garbled circuits. This was done in [Kra+19] for a Ring-LWE based scheme. Their solution does not need any noise flooding or increased parameters of the underlying scheme, however, it relies on generic multi-party computation techniques like garbled circuits, and the partial decryption shares are generated using an expensive, interactive protocol rather than non-interactively as in our setting.

What about $\mathsf{IND\text{-}CCA}$ *security?* We could likely upgrade our construction (for PKE) to be $\mathsf{IND\text{-}CCA}$ secure using non-interactive zero-knowledge proofs, similarly to [Dev+21]. However, note that $\mathsf{IND\text{-}CCA}$ security is not possible for homomorphic encryption, and $\mathsf{IND\text{-}CPA}$ is still useful for standard PKE; indeed, [HV22] showed that an $\mathsf{IND\text{-}CPA}$ secure KEM suffices to prove security of TLS-1.3. Furthermore, when running TLS with ephemeral keys and no key re-use, the adversary only ever sees a single ciphertext under any public key — this is an ideal use-case for using our construction in a threshold post-quantum TLS setting (e.g. for hardening security of a TLS server), since we only need to choose the parameters to be secure against a single decryption query.

1.1 Overview of Techniques

Constructing full-threshold $\mathsf{OW\text{-}CPA}$*-secure* TFHE. To simplify the presentation in the introduction, we first describe our construction from Section 5 in the full-threshold setting and then explain how to get t-out-of-n threshold. As a starting point, we take any encryption scheme whose decryption function is nearly linear, as is the case for most LWE-based encryption schemes (including FHE). That is, for a given ciphertext ct on a message m with respect to a key pair $(\mathsf{sk}, \mathsf{pk})$, it holds that $\langle \mathsf{sk}, \mathsf{ct} \rangle = m + e_{\mathsf{ct}}$, where e_{ct} is what we earlier called decryption noise and depends on the ciphertext and the secret key.[5]

To achieve threshold decryption, we use standard additive secret sharing to split the secret key into $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ in a setup phase. By linearity, we could simply set the partial decryption shares as $\tilde{d}_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle$, and use these for reconstruction. However, after summing all shares together, the parties recover e_{ct}, which leaks information on sk. As in previous threshold solutions for lattice-based schemes, to compute their decryption share d_i every party now locally adds to $\tilde{d}_i$ a noise term e_i which is sampled from the noise flooding distribution $\mathcal{D}_{\mathsf{flood}}$. When summing those partial decryption shares together, the parties learn $m + e_{\mathsf{ct}} + \sum_{i=1}^{n} e_i$.

Now, consider an adversary who learns up to $n-1$ secret key shares sk_i, and then the missing partial decryption for a batch of ciphertexts. For security, we want to guarantee that the scheme is still secure in the presence of this leakage. In previous works, when $\mathcal{D}_{\mathsf{flood}}$ is superpolynomially larger than the maximum value of e_{ct}, it is argued that the partial decryptions can be *simulated* in a way that is statistically indistinguishable. Unfortunately, when $\mathcal{D}_{\mathsf{flood}}$ is polynomially bounded, this no longer holds, since the statistical distance is large.

Using the Rényi divergence instead of statistical distance, the probability preservation property allows us to reason about the probability of a bad event happening in two different games. Roughly speaking, this says that if D_1, D_2 are distributions such that the Rényi divergence of D_1 from D_2 is at most δ, then for any event E, it holds that $\Pr[D_1(E)] \leq (\Pr[D_2(E)] \cdot \delta)^c$, for some constant c close to 1. If the event E occurs with negligible probability in game D_2, then we can get by with a polynomial-sized δ to argue the same holds in D_1. However, this is inherently hard to make use of in distinguishing games like $\mathsf{IND\text{-}CPA}$, where probabilities of winning are close to $1/2$.

Instead of $\mathsf{IND\text{-}CPA}$ security, we can aim for $\mathsf{OW\text{-}CPA}$ security, which is easier to manage with the Rényi divergence. When defining $\mathsf{OW\text{-}CPA}$ in the threshold setting (see Section 3), the main difference is that the adversary also obtains $n-1$ shares of the secret key and has access to a bounded number of partial decryption queries. In the security proof, we will modify the security experiment such that in a first step, the answers to the partial decryption queries no longer depend on the underlying secret key sk, and in a second step the secret key shares are also independent of sk. In this case, $\mathsf{OW\text{-}CPA}$ security of the threshold scheme is implied by the $\mathsf{OW\text{-}CPA}$ security of the underlying standard encryption scheme. We simulate the partial decryption terms $e_{\mathsf{ct}} + \sum_{i=1}^{n} e_i$ by sampling some independent noise $e' \leftarrow \mathcal{D}_{\mathsf{sim}}$. As long as the Rényi divergence between the two noise distributions is bounded by a constant, we can appeal to the probability preservation property, and the negligible probability of some PPT adversary guessing the message is preserved in both games. Note that previous works always chose $\mathcal{D}_{\mathsf{sim}} = \mathcal{D}_{\mathsf{flood}}$, but we later exploit in Section 6 that choosing a different $\mathcal{D}_{\mathsf{sim}}$ can lead to better parameters.

[5] Actually, it only reveals an encoding of m, which is easy to decode as long as parameters are set accordingly.

From full-threshold to t-out-of-n threshold. When moving to the t-out-of-n setting, a natural choice is to use Shamir secret sharing instead of additive sharing. However, this leads to the problem that reconstruction is no longer addition, and instead requires multiplying the partial decryptions with Lagrange interpolation coefficients. These coefficients may be large, which in turn blows up the noise, breaking correctness. We offer two different solutions to this issue.

First, as in [Bon+18], we can use a special type of linear secret sharing scheme with binary coefficients, so that reconstruction is always a simple sum. Efficient threshold schemes with this property exist, for any n, t. We also consider a second method based on *pseudorandom secret sharing* [CDI05], which allows the parties to generate sharings of bounded, pseudorandom values without interaction. This uses replicated secret sharing, which is more expensive, but on the other hand, allows the partial decryptions to be converted into Shamir sharings before reconstruction. This leads to smaller partial decryptions, slightly better parameters and gives a form of robustness via Shamir error correction.

From $\mathsf{OW\text{-}CPA}$ *to* $\mathsf{IND\text{-}CPA}$ *security, Transform 1.* Our first transformation (Section 4.1) can be seen as the generalization of an existing $\mathsf{OW\text{-}CPA}$ to $\mathsf{IND\text{-}CPA}$ transformation in the random oracle model [HHK17] to the threshold setting. The main idea is to use the $\mathsf{OW\text{-}CPA}$-secure scheme to encrypt random messages. The vector $\mathbf{x}$ composed of those random messages then serves as input to a random oracle F, whose output hides the message m we are about to encrypt. By appending the output of a second and independent random oracle G queried on the same vector $\mathbf{x}$, we make sure that no adversary can provide incorrect decryption shares without getting caught. To this end, we define in Section 3.2 two new notions of *robustness* for (passively secure) threshold public key encryption, which might be of independent interest. The length of the vector $\mathbf{x}$ provides a trade-off between the security loss of the reduction and the compactness of ciphertexts.

From $\mathsf{OW\text{-}CPA}$ *to* $\mathsf{IND\text{-}CPA}$ *security, Transform 2.* Whereas the reduction from above is simple and tight, it has the disadvantage of needing a random oracle to mask the message m. When we consider threshold decryption in the fully-homomorphic setting, we need to make sure that we can homomorphically evaluate ciphertexts. However, the use of the random oracle makes such an evaluation impossible, as there is no efficient circuit description of random oracles. We thus propose in Section 4.2 a second transformation which now is in the standard model (but does not give robustness).

The high level idea to encrypt a message m of δ bits, is to sample a random message x and to encrypt it using the $\mathsf{OW\text{-}CPA}$-secure scheme. Then, the message bits are hidden by δ hard-core bits coming from a concatenation of δ Goldreich-Levin extractors. We use the notion of unpredictable entropy to give a bound on how many pseudorandom bits can be extracted from this construction. The notion of unpredictable entropy has been introduced and studied by Hsiao et al. [HLR07] in the context of conditional computational entropy. We say that a message x has unpredictability entropy $\gamma + \varepsilon'$ for some $\varepsilon' > 0$ if for any PPT

adversary $\mathcal{A}$ the probability of finding x given $\mathsf{Enc}(\mathsf{pk}, x)$ is at most $2^{-\gamma+\varepsilon'}$. We can then use existing results that show that a concatenation of δ Goldreich-Levin extractors can be used to extract $\gamma - \varepsilon' - O(\mathsf{polylog}(\gamma))$ pseudorandom bits, where γ is the bit size of the message x. Those pseudorandom bits then allow us to encrypt a message such that the ciphertexts of two given messages are computationally indistinguishable.

Sample Parameters and Security Analysis. We conclude our work by discussing in Section 6 how to choose concrete sample parameters for our threshold PKE scheme, when instantiating it with the lattice-based scheme Kyber [Sch+20].

As an example, to obtain 1-out-of-2 threshold decryption with a single query (e.g. for ephemeral key exchange), we can use the same parameters as $\mathsf{Kyber}1024$ with a modulus increased only by a factor of 5, while supporting > 100 bits of classical hardness from our reduction. In a setting with up to 2^{32} queries, we need to use a 39-bit modulus and slightly larger module rank; this increases the ciphertext size by around 5x.

Finally, we show in Section 6.2 that using the Rényi divergence leads to almost optimal parameters by providing an attack if the adversary gets access to slightly more partial decryptions.

2 Preliminaries

For any positive integer q, we denote by $\mathbb{Z}_q$ the integers modulo q and for any positive integer n, we denote by $[n]$ the set $\{1, \ldots, n\}$. Vectors are denoted in bold lowercase and matrices in bold capital letters. The identity matrix of order m is denoted by $\mathbf{I}_m$. The concatenation of two matrices $\mathbf{A}$ and $\mathbf{B}$ with the same number of rows is denoted by $[\mathbf{A}|\mathbf{B}]$. The abbreviation PPT stands for probabilistic polynomial-time. When we split a PPT adversary $\mathcal{A}$ in several sub algorithms $(\mathcal{A}_i)_i$, we implicitly assume that $\mathcal{A}_i$ outputs a state that is passed to the next $\mathcal{A}_{i+1}$. We call a function $\mathsf{negl}(\cdot)$ negligible in λ if $\mathsf{negl}(\lambda) = \lambda^{-\omega(1)}$, i.e., it decreases faster towards 0 than the inverse of any polynomial.

Throughout the paper we make use of the random oracle model (ROM), where we assume the existence of perfectly random functions, realized by oracles. For a random oracle $\mathsf{F}\colon \{0,1\}^n \to \{0,1\}^m$ it holds that $\Pr[\mathsf{F}(x) = y] = 2^{-m}$ and that $\Pr[\mathsf{F}(x) = \mathsf{F}(x')\colon x \neq x'] = \Pr[\mathsf{F}(x) = y] \cdot \Pr[\mathsf{F}(x') = y] = 2^{-2m}$. Hence, random oracles are per definition collision resistant. For $x, y \in \{0,1\}^n$ we denote by $x \oplus y$ the bit-wise XOR operator.

2.1 Probability and Entropy

For a finite set S, we denote its cardinality by $|S|$ and the uniform distribution over S by $U(S)$. The operation of sampling an element $x \in S$ according to a distribution D over S is denoted by $x \leftarrow D$, where the set S is implicit.

For standard deviation $\sigma > 0$ and mean $c \in \mathbb{R}$, we define the continuous Gaussian distribution $D_{\sigma,c}\colon \mathbb{R} \to (0,1]$ by $D_{\sigma,c}(x) = 1/(\sigma\sqrt{2\pi}) \cdot \exp(-(x -$

$c)^2/(2\sigma^2))$. We also define the rounded Gaussian distribution over $\mathbb{Z}$, by rounding the result to the nearest integer, and denote this by $\lfloor D_{\sigma,c} \rceil$.

A random variable X over $\mathbb{R}$ is called τ-subgaussian for some $\tau > 0$ if for all s it holds $\mathbb{E}[\exp(sX)] \leq \exp(\tau^2 s^2/2)$. A τ-subgaussian random varable satisfies $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] \leq \tau^2$. We associate to X the width $\sigma = \sqrt{\mathbb{E}[X^2]}$. The continuous Gaussian distribution D_σ and its rounded version $\lfloor D_\sigma \rceil$ are σ-subgaussian. Further, the uniform distribution over $[-a, a] \cap \mathbb{Z}$ is a-subgaussian.

The statistical distance between two probability distributions X and Y, denoted by $\mathsf{sdist}(X, Y)$, is defined as $\max_T |\Pr[T(X) = 1] - \Pr[T(Y) = 1]|$, where T is any test function. The computational distance, denoted by $\mathsf{cdist}(X, Y)$, takes the maximum only over test functions which can be described by circuits of size $\mathsf{poly}(\lambda)$, where λ is the security parameter. For any event E, the probability preservation property of sdist (resp. cdist) states that $X(E) \leq Y(E) + \mathsf{sdist}(X, Y)$ (resp. $X(E) \leq Y(E) + \mathsf{cdist}(X, Y)$).

Definition 2.1 (Unpredictable Entropy). *For a distribution (X, Z), we say that X has* unpredictable entropy *at least k conditioned on Z, if there exists a collection of distributions Y_Z (giving rise to a joint distribution (Y, Z)) such that $\mathsf{cdist}((X, Z), (Y, Z)) \leq \varepsilon$, and for all circuits C of size $\mathsf{poly}(\lambda)$,*

$$\Pr[C(Z) = Y] \leq 2^{-k}.$$

We write $H_\varepsilon^{\mathsf{unp}}(X|Z) \geq k$.

Definition 2.2 (Concatenated Goldreich-Levin Extractor). *Fix $n, \delta \in \mathbb{N}$. We define the* concatenated Goldreich-Levin extractor $\mathcal{E} \colon \{0,1\}^n \times (\{0,1\}^n)^\delta \to \{0,1\}^\delta \times (\{0,1\}^n)^\delta$ *as*

$$\mathcal{E}(x, s_1, \ldots, s_\delta) := (\langle x, s_1 \rangle \bmod 2, \ldots, \langle x, s_\delta \rangle \bmod 2, s_1, \ldots, s_\delta).$$

Lemma 2.3 ([HLR07, Lem. 9 & Sec. 4.2]). *Let X be a distribution with unpredictable entropy $H_\varepsilon^{\mathsf{unp}}(X|Z) \geq k$ and let $\mathcal{E}$ be the concatenated Goldreich-Levin extractor for some $n, \delta \in \mathbb{N}$. If $k = \delta + O(\log_2 1/\varepsilon)$, then $\mathcal{E}$ extracts δ pseudorandom bits, i.e.,*

$$\mathsf{cdist}\left((Z, \mathcal{E}(X, U(\{0,1\}^{n\delta}))), (Z, U(\{0,1\}^\delta \times \{0,1\}^{n\delta}))\right) \leq 5\varepsilon.$$

For $\varepsilon = 2^{-\mathsf{polylog}(n)}$, the lemma above yields $k = \delta + O(\mathsf{polylog}(n))$.

The Rényi divergence (RD) defines an alternative measure of distribution closeness. We follow [Bai+18] and use a definition of the RD which is the exponential of the classical definition. We restrict the order a to be in $(1, \infty)$.

Definition 2.4 (Rényi Divergence). *Let P and Q be two discrete probability distributions such that* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. *For $a \in (1, \infty)$ the* Rényi divergence of order *a is defined by*

$$\mathrm{RD}_a(P, Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

The definitions are extended in the natural way to continuous distributions. We recall some useful properties of the RD. The first two were proven in [EH14] and the last one was proven in [Ros20, Prop. 2].

Lemma 2.5. *Let P, Q be two discrete probability distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. *For $a \in (1, \infty)$, it yields:*

Data Processing Inequality: $\mathrm{RD}_a(g(P)\|g(Q)) \leq \mathrm{RD}_a(P\|Q)$ *for any function g, where $g(P)$ (resp. $g(Q)$) denotes the distribution of $g(y)$ induced by sampling $y \leftarrow P$ (resp. $y \leftarrow Q$).*

Probability Preservation: *Let $E \subset \mathrm{Supp}(Q)$ be an event, then for $a \in (1, \infty)$*

$$Q(E) \cdot \mathrm{RD}_a(P\|Q) \geq P(E)^{\frac{a}{a-1}}.$$

Multiplicativity: *Let P, Q be two probability distributions of a pair of random variables (Y_1, Y_2). For $i \in \{1, 2\}$, let P_i (resp. Q_i) denote the marginal distribution of Y_i under P (resp. Q), and let $P_{2|1}(\cdot|y_1)$ (resp. $Q_{2|1}(\cdot|y_1)$) denote the conditional distribution of Y_2 given that $Y_1 = y_1$. Then for $a \in (1, \infty)$*

$$\mathrm{RD}_a(P\|Q) \leq \mathrm{RD}_a(P_1\|Q_1) \cdot \max_{y_1 \in Y_1} \mathrm{RD}_a(P_{2|1}(\cdot|y_1)\|Q_{2|1}(\cdot|y_1)).$$

The Rényi divergence of two shifted Gaussians is given below. This also allows us to bound the RD of rounded Gaussians by the data processing inequality.

Lemma 2.6 ([GAL13]). *Let σ be a positive real number and $c \in \mathbb{Z}$. Then for $a \in (1, \infty)$ it yields*

$$\mathrm{RD}_a(D_{\sigma,c}\|D_\sigma) = \exp\left(\frac{ac^2}{2\sigma^2}\right).$$

We provide the proof of the following lemma in Appendix A.1.

Lemma 2.7. *Let D_1, D_2 be two probability distributions over $\mathbb{Z}$ and $e_1, \ldots, e_N$ be (possibly dependent) random variables over $\mathbb{Z} \cap [-B, B]$ for some $B \in \mathbb{Z}$, for which there exist $a \in (1, \infty)$ and $\rho \geq 1$ such that for all β with $|\beta| \leq B$, it holds that* $\mathrm{Supp}(D_1 + \beta) \subseteq \mathrm{Supp}(D_2)$, *and furthermore,* $\mathrm{RD}_a(D_1 + \beta\|D_2) \leq \rho$. *Then,*

$$\mathrm{RD}_a((D_1 + e_N, \ldots, D_1 + e_1)\|D_2^N) \leq \rho^N.$$

2.2 Linear Secret Sharing

We use linear secret sharing schemes (LSSS) for monotone access structures with a special $\{0, 1\}$-reconstruction property, as follows.

Definition 2.8 (Monotone Access Structure). *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of parties and $2^{\mathcal{P}}$ its power set. A* monotone access structure *is a collection of sets $\mathbb{A} \subset 2^{\mathcal{P}}$, such that for any $S \in \mathbb{A}$, if $T \supset S$ then $T \in \mathbb{A}$. We say that $\mathbb{A}$ is* efficient *if membership of $\mathbb{A}$ can be verified in time* $\mathsf{poly}(\lambda)$, *where $\mathbb{A}$ is viewed as a function of λ.*

In this work, we only consider efficient access structures. To ease notation, we identify a party P_i with its index i, viewing each set $S \in \mathbb{A}$ as a subset of $[n]$. For any $S \subset [n]$ and vector $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, we let $\mathbf{v}|_S$ denote the vector of shares restricted to $\mathbf{v}_i$ for indices $i \in S$.

Definition 2.9 (Linear Secret Sharing Scheme). *Let q, L, n be positive integers and $\mathbb{A}$ a monotone access structure. A linear secret sharing scheme* LSSS *for $\mathbb{A}$ is defined by a randomized algorithm* $\mathsf{Share} : \mathbb{Z}_q \to (\mathbb{Z}_q^L)^n$ *and a family of deterministic algorithms* $\mathsf{Rec}_S : (\mathbb{Z}_q^L)^{|S|} \to \mathbb{Z}_q$*, for $S \subseteq [n]$, which satisfy:*

Privacy: *For any set $S \notin \mathbb{A}$, any $x, x' \in \mathbb{Z}_q$ and $\mathbf{v} \in \mathbb{Z}_q^{L|S|}$, it holds that* $\Pr[\mathsf{Share}(x)|_S = \mathbf{v}] = \Pr[\mathsf{Share}(x')|_S = \mathbf{v}]$.

Reconstruction: *For any set $S \in \mathbb{A}$, any $x \in \mathbb{Z}_q$ and $\mathbf{v} = \mathsf{Share}(x)$, the reconstruction algorithm outputs $\mathsf{Rec}_S(\mathbf{v}|_S) = x$.*

Linearity: *For any $\alpha, \beta \in \mathbb{Z}_q$, any set S with $|S| > t$ and any share vectors $\mathbf{u}, \mathbf{v}$, it holds that $\mathsf{Rec}_S(\alpha\mathbf{u}|_S + \beta\mathbf{v}|_S) = \alpha\mathsf{Rec}_S(\mathbf{u}|_S) + \beta\mathsf{Rec}(\mathbf{v}|_S)$.*

When the set of shares is $S = [n]$, we write Rec *instead of* $\mathsf{Rec}_{[n]}$.

We need the following notion of valid and invalid share sets [Bon+18].

Definition 2.10. *Let $x \in \mathbb{Z}_q$, $(\mathbf{v}_1, \ldots, \mathbf{v}_n) = \mathsf{Share}(x)$, and write $\mathbf{v}_i = (\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,L})$. A set of pairs of indices $T \subseteq [n] \times [L]$ is an* invalid set of share elements *if the corresponding shares $(\mathbf{v}_{i,j})_{(i,j) \in T}$ reveal no information about x. Otherwise, we say that T is a* valid set of share elements. *We additionally say:*

- *$T \subseteq [n] \times [L]$ is a* maximal invalid set of share elements *if it is invalid, but for any $(i, j) \in [n] \times [L] \setminus T$, the set $T \cup \{(i, j)\}$ is a valid set of share elements.*
- *$T \subseteq [n] \times [L]$ is a* minimal valid set of share elements *if it is valid, but for any $T' \subsetneq T$, the set T' is an invalid set of share elements.*

Note that in any LSSS, a valid set as defined above always allows reconstruction of the secret x. This is because an LSSS can equivalently be defined by a matrix M, such that each share element $\mathbf{v}_{i,j}$ is computed as the inner product of some row of M and $(x, r_1, \ldots, r_{n-1})$, where r is the randomness used in Share. Reconstruction is possible for a given set of share elements iff the corresponding set of rows of M span the target vector $(1, 0, \ldots, 0)$. This definition implies that any set of rows is either invalid — and reveals nothing about x — or valid, and allows full reconstruction. For further details, see e.g. [Bei96, Chapter 4].

Our main construction requires that the reconstruction function Rec_S takes a $0/1$ combination of its inputs. In the following, we require this to hold not only for any set of shares corresponding to a valid set of parties in $\mathbb{A}$, but for any valid set of share elements. This property is equivalent to the notion of a derived $\{0, 1\}$-LSSS, used in [JRS17].[6]

[6] [Bon+18] only assumed a weaker property for their threshold FHE construction. However, this is a mistake introduced when merging the two works [JRS17] and [Bon+17] (and has been confirmed by the authors of [JRS17]).

Definition 2.11 (Strong $\{0,1\}$-Reconstruction). *We say that a* LSSS *has* strong {0,1}-reconstruction *if for any secret x and $(\mathbf{v}_1,\ldots,\mathbf{v}_n) = \mathsf{Share}(x)$, for any valid set of share elements $T \subseteq [n]\times[L]$, there exists a subset $T' \subseteq T$ such that $\sum_{(i,j)\in T'} \mathbf{v}_{i,j} = x$, where $\mathbf{v}_i = (\mathbf{v}_{i,1},\ldots,\mathbf{v}_{i,L})$.*

Sharing Values in R_q. In our constructions, we share $\mathbf{x} \in R_q^r$, where $R_q = \mathbb{Z}_q[X]/f(X)$, instead of just in $\mathbb{Z}_q$. We do this coefficient-wise, by separately sharing each coefficient of the r polynomials in $\mathbf{x}$. Each party's share then lies in $(R_q^r)^L$, and the parties can perform R_q-linear operations on these shares.

Example Linear Secret Sharing Schemes. In Table 1, we detail a few example secret sharing schemes we consider. Their description can be found in Appendix A.2. The schemes are for t-out-of-n access structures, where any $t+1$ parties can reconstruct, and they all have strong $\{0,1\}$-reconstruction. In the table, we show two quantities $\tau_{\mathsf{max}}, \tau_{\mathsf{min}}$, which are relevant for choosing parameters in our constructions of Section 5 and we will refer to later. By τ_{max} we denote the size of the smallest maximal invalid set of share elements, while τ_{min} is the size of the largest minimal valid set of share elements.

Table 1. Example t-out-of-n linear secret sharing schemes with strong $\{0,1\}$-reconstruction. Details for the last row are omitted, due to their complexity.

Scheme	Sharing method	P_i's share	L	τ_{max}	τ_{min}
Additive	$x = \sum_{i=1}^{n} x_i$	x_i	1	$n-1$	n
Replicated	$x = \sum_{A,\lvert A\rvert=t} x_A$	$\{x_A\}_{i\notin A}$	$\binom{n-1}{t}$	$(n-t)(\binom{n}{t}-1)$	$\binom{n}{t}$
Naive	$x = \sum_{i\in A} x_{A,i},\ \lvert A\rvert = t+1$	$\{x_{A,i}\}_{i\in A}$	$\binom{n-1}{t}$	$t\binom{n}{t+1}$	$t+1$
Monotone Boolean formula for threshold fn. [Val84]			$O(n^{4.3})$	$O(n^{5.3})$	$O(n^{5.3})$

2.3 Learning With Errors

In the following, we recall the definitions of the decision LWE problem [Reg05] and its module variant [LS15]. Both are formulated with a bounded uniform secret and noise. We define the set $S_\beta = \{a \in \mathbb{Z} \colon \lvert a\rvert \le \beta\}$ with $\beta \in \mathbb{N}$.

Definition 2.12 (LWE). *Let $m, r, \beta, q \in \mathbb{N}$. The* Learning With Errors *problem* $\mathrm{LWE}_{q,m,r,\beta}$ *is defined as follows. Given $\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times r})$ and $\mathbf{t} \in \mathbb{Z}_q^m$. Decide whether $\mathbf{t} \leftarrow U(\mathbb{Z}_q^m)$ or if $\mathbf{t} = [\mathbf{A}|\mathbf{I}_m]\cdot\mathbf{s}$, where $\mathbf{s} \leftarrow U(S_\beta^{m+r})$.*

In the module setting, we replace $\mathbb{Z}_q$ by the quotient $R_q = \mathbb{Z}_q[X]/f(X)$ for some irreducible $f(X)$ of degree d. Further, we define $\tilde{S}_\beta = \{a \in R \colon \|a\|_\infty \leq \beta\}$ with $\beta \in \mathbb{N}$.

Definition 2.13 (M-LWE). *Let $m, r, \beta, q \in \mathbb{N}$. The* Module Learning With Errors *problem* $\text{M-LWE}_{q,m,r,\beta}$ *is defined as follows. Given $\mathbf{A} \leftarrow U(R_q^{m\times r})$ and $\mathbf{t} \in R_q^m$. Decide whether $\mathbf{t} \leftarrow U(R_q^m)$ or if $\mathbf{t} = [\mathbf{A}|\mathbf{I}_m] \cdot \mathbf{s}$, where $\mathbf{s} \leftarrow U(\tilde{S}_\beta^{m+r})$.*

Lastly, we also define a computational variant of LWE, where no reduction modulo q is performed [Boo+18], which will be relevant in Section 6.

Definition 2.14 (I-LWE). *Let $m, r \in \mathbb{N}$ and let χ_w, χ_e be two probability distributions over $\mathbb{Z}$. The* Integer Learning With Errors *problem* $\text{I-LWE}_{m,r,\chi_w,\chi_e}$ *is defined as follows. Given $\mathbf{W} \leftarrow \chi_w^{m\times r}$ and $\mathbf{t} = \mathbf{W}\mathbf{z} + \mathbf{e}$, where $\mathbf{z} \in \mathbb{Z}^r$ and $\mathbf{e} \leftarrow \chi_e^m$. Find $\mathbf{z}$. We call $(\mathbf{W}, \mathbf{t} = \mathbf{W}\mathbf{z} + \mathbf{e})$ an instance of the* I-LWE *distribution.*

Theorem 2.15 ([Boo+18, Thm. 4.5]). *Suppose that χ_w is τ_w-subgaussian and χ_e is τ_e-subgaussian. Let $(\mathbf{W}, \mathbf{t} = \mathbf{W}\mathbf{z}+\mathbf{e})$ be an instance of the* $\text{I-LWE}_{m,r,\chi_w,\chi_e}$ *distribution for some $\mathbf{z} \in \mathbb{Z}^r$. There exist constants $C_1, C_2 > 0$ such that for all $\nu \geq 1$ the least square method recovers $\mathbf{z}$ with probability $1 - \frac{1}{2r} - 2^{-\nu}$ if*

$$m \geq 4\frac{\tau_w^4}{\sigma_w^4}(C_1 r + C_2\nu) \text{ and } m \geq 32\frac{\tau_e^2}{\sigma_w^2}\log_2(2r).$$

3 Threshold Fully Homomorphic Encryption

In this section, we recall the definitions of threshold fully homomorphic encryption schemes (TFHE), as well as their properties of compactness and decryption correctness. Furthermore, we give different notions of robustness for (passively secure) threshold public key encryption, which model an adversary who may send incorrect or missing partial decryptions. We don't define these in the fully homomorphic case, where our construction assumes a passive adversary. We then define our notions of OW-CPA and IND-CPA security for TFHE schemes.

3.1 Definitions of Threshold FHE/PKE

We recall the definition of a fully homomorphic threshold public key encryption scheme. We implicitly assume that after Setup, all algorithms are given the public parameters as input. We omit the partial verification algorithm used in previous works (e.g., [BBH06]), which was only used to model stronger notions of robustness that also capture CCA attacks.

Definition 3.1 (TFHE). *A* fully homomorphic threshold public key encryption *scheme* (TFHE) *for a message space $\mathcal{M}$ and circuits of depth κ is a tuple of* PPT *algorithms* TFHE = (Setup, Enc, Eval, PartDec, Combine) *defined as follows:*

$\mathsf{Setup}(1^\lambda, 1^\kappa, n, t) \to (\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$**:** *On input the security parameter λ, a bound on the circuit depth κ, the number of parties n and a threshold value $t \in \{1, \ldots, n-1\}$, the setup algorithm outputs the public parameters* pp*, a public key* pk *and a set of secret key shares* $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$*.*

$\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$**:** *On input the public key* pk *and a message $m \in \mathcal{M}$, the encryption algorithm outputs a ciphertext* ct*.*

$\mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k) \to \mathsf{ct}$**:** *On input the public key* pk*, a circuit $C \colon \mathcal{M}^k \to \mathcal{M}$ of depth at most κ and a set of ciphertexts* $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$*, the evaluation algorithm outputs a ciphertext* ct*.*

$\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}) \to d_i$**:** *On input a key share* sk_i *for some $i \in [n]$ and a ciphertext* ct*, the partial decryption algorithm outputs a partial decryption share d_i.*

$\mathsf{Combine}(\{d_i\}_{i \in S}, \mathsf{ct}) \to m'$**:** *On input a set of decryption shares $\{d_i\}_{i \in S}$ and a ciphertext* ct*, where $S \subset [n]$ is of size at least $t+1$, the combining algorithm outputs a message $m' \in \mathcal{M} \cup \{\perp\}$.*

The above can be seen as a generalization encompassing non-threshold and threshold PKE and FHE.

Definition 3.2 (TPKE)**.** *A* threshold public key encryption *scheme* (TPKE) *for a message space $\mathcal{M}$ is a* TFHE *scheme, where $k = 1$ and the only allowed circuit $C \colon \mathcal{M} \to \mathcal{M}$ is the identity. In this case, we drop the trivial evaluation algorithm* Eval *and the parameter κ in the scheme's specifications.*

Definition 3.3 (FHE)**.** *A* fully-homomorphic public key encryption *scheme* (FHE) *for a message space $\mathcal{M}$ is a* TFHE *scheme, where $n = 1$. In this case, we drop the parameters n and t in the scheme's specifications. To simplify notations, we merge* PartDec *and* Combine *into one single algorithm that we denote* Dec*. Hence, the algorithm* Dec *takes* sk *and* ct *as input and outputs $m' \in \{\mathcal{M} \cup \{\perp\}\}$.*

We require compactness and correctness, whose definitions we recall in App. B.

3.2 Robustness

We now introduce two definitions of robustness. We call the first one *weak chosen-ciphertext robustness* and the second *strong chosen-plaintext robustness.*

In the first case, it should be hard for an adversary, having access to all secret key shares, to provide one single ciphertext and two different set of decryption shares such that they combine to two different messages. Our definition is closely related to the notion of *consistency*, as for instance defined by [BBH06], with the difference that we do not allow the adversary to win by making the decryption output $\perp$. (This is unavoidable in our setting, since we do not have a separate PartVerify algorithm to verify validity of decryption shares.)

Definition 3.4 (Weak Chosen-Ciphertext Robustness). *We call a* TPKE *scheme* weakly chosen-ciphertext robust *if for all λ, n, t and for all* PPT *adversaries $\mathcal{A}$ it yields*

$$\mathsf{Adv}^{\text{w-cc-robust}}_{\mathsf{TPKE}}(\mathcal{A}) := \Pr[\mathsf{Expt}^{\text{w-cc-robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda, n, t) = 1] = \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\text{w-cc-robust}}_{\mathcal{A},\mathsf{TPKE}}$ *is the experiment specified in Figure 1.*

$$
\begin{array}{ll}
\multicolumn{2}{l}{\underline{\mathsf{Expt}^{\mathsf{w\text{-}cc\text{-}robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda, n, t)}} \\
1: & (\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, n, t) \\
2: & (\mathsf{ct}, S, S', \{d_i\}_{i \in S}, \{d'_i\}_{i \in S'}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in [n]}) \\
3: & m \leftarrow \mathsf{Combine}(\{d_i\}_{i \in S}, \mathsf{ct}) \\
4: & m' \leftarrow \mathsf{Combine}(\{d'_i\}_{i \in S'}, \mathsf{ct}) \\
5: & \textbf{return } m' \neq m \wedge \perp \notin \{m, m'\}
\end{array}
$$

Fig. 1. Experiment for the weak chosen-ciphertext robustness of TPKE schemes.

In the second case, the adversary is given the secret key shares of the corrupted parties together with an honestly formed ciphertext. In order to win the experiment, they have to come up with partial decryption shares such that the combine algorithm, together with honestly generated partial decryption shares, outputs a different message (including the abort message $\perp$).

We note that for $t < n/2$, it's possible to transform *any* weakly chosen-ciphertext robust TPKE scheme into one that guarantees strong chosen-plaintext robustness. To do so, one simply lets Combine try all possible subsets of size $t+1$. As $t < n/2$, there exists a set of size $t + 1$ composed of only honest partial decryption shares and hence, it successfully combines to a message.

Definition 3.5 (Strong Chosen-Plaintext Robustness). *A* TPKE *scheme provides* strong chosen-plaintext robustness *if for all* λ, n, t *and for all* PPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *it yields*

$$\mathsf{Adv}^{\mathsf{s\text{-}cp\text{-}robust}}_{\mathsf{TPKE}}(\mathcal{A}) := \Pr[\mathsf{Expt}^{\mathsf{s\text{-}cp\text{-}robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda, n, t) = 1] = \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\mathsf{s\text{-}cp\text{-}robust}}_{\mathcal{A},\mathsf{TPKE}}$ *is the experiment specified in Figure 2.*

$$
\begin{array}{ll}
\multicolumn{2}{l}{\underline{\mathsf{Expt}^{\mathsf{s\text{-}cp\text{-}robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda, n, t)}} \\
1: & (\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, n, t) \\
2: & (S, m) \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| \leq t \\
3: & \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m) \\
4: & d_j \leftarrow \mathsf{PartDec}(\mathsf{sk}_j, \mathsf{ct}),\ \forall j \in [n] \setminus S \\
5: & \{d_i\}_{i \in S} \leftarrow \mathcal{A}_2(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S}, \{d_j\}_{j \notin S}, \mathsf{ct}) \\
6: & m' \leftarrow \mathsf{Combine}(\{d_i\}_{i \in [n]}, \mathsf{ct}) \\
7: & \textbf{return } m' \neq m
\end{array}
$$

Fig. 2. Experiment for strong chosen-plaintext robustness of TPKE schemes.

3.3 One-Wayness

We now present our definition of $\mathsf{OW\text{-}CPA}$ security for TFHE schemes. It is essentially the standard experiment of $\mathsf{OW\text{-}CPA}$ for PKE schemes, with some modifications to account for the threshold and the fully-homomorphic setting. Regarding threshold, the adversary additionally has access to the secret key shares of the corrupted parties, defined by the set S which is of size at most t, where t is the threshold parameter. Further, they are allowed to make (at most ℓ) adaptive queries in order to obtain the partial decryption shares of all parties for some message-ciphertext pairs. To account for the full homomorphic setting, the message-ciphertext pairs can stem from evaluations of ciphertexts on some circuit C of bounded depth (at most κ).

Definition 3.6 (ℓ-$\mathsf{OW\text{-}CPA}$ for TFHE). *We call a* TFHE *scheme* ℓ-$\mathsf{OW\text{-}CPA}$ *secure for the security parameter λ, the circuit depth bound κ, the threshold parameters n, t and the query bound ℓ, if for all* PPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$

$$\mathsf{Adv}^{\ell\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{TFHE}}(\mathcal{A}) := \Pr[\mathsf{Expt}^{\ell\text{-}\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{TFHE}}(1^\lambda, 1^\kappa, n, t) = 1] = \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\ell\text{-}\mathsf{OW\text{-}CPA}}_{\mathcal{A},\mathsf{TFHE}}$ *is the experiment in Fig. 3 with* $\mathsf{ctr} = 0$ *at the beginning.*

Definition 3.7 (ℓ-$\mathsf{OW\text{-}CPA}$ for TPKE). *We call a* TPKE *scheme* ℓ-$\mathsf{OW\text{-}CPA}$ *secure for the security parameter λ, the threshold parameters n, t and the query bound ℓ, if it is* ℓ-$\mathsf{OW\text{-}CPA}$ *secure as a* TFHE *scheme, where $k = 1$ and the only allowed circuit $C \colon \mathcal{M} \to \mathcal{M}$ is the identity. For a* PPT *adversary $\mathcal{A}$ we denote their advantage as* $\mathsf{Adv}^{\ell\text{-}\mathsf{OW\text{-}CPA}}_{\mathsf{TPKE}}(\mathcal{A})$.

Note that the only difference between ℓ-$\mathsf{OW\text{-}CPA}$ security of TPKE and the standard notion of $\mathsf{OW\text{-}CPA}$ security of PKE is that the adversary obtains t shares of the secret key and has access to ℓ partial decryption queries. Finally, note that for any scheme to be $\mathsf{OW\text{-}CPA}$ secure, the size of the plaintext space $\mathcal{M}$ must be superpolynomial in the security parameter, to prevent guessing attacks.

3.4 Indistinguishability

In the following, we present our definition of $\mathsf{IND\text{-}CPA}$ security for TFHE. It is obtained by applying the same modifications as in the section before, now to the standard definition of $\mathsf{IND\text{-}CPA}$ security for PKE.

Definition 3.8 (ℓ-$\mathsf{IND\text{-}CPA}$ for TFHE). *We call a* TFHE *scheme* ℓ-$\mathsf{IND\text{-}CPA}$ *secure for the security parameter λ, the circuit depth bound κ, the threshold parameters n, t and the query bound ℓ, if for all* PPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$

$$\mathsf{Adv}^{\ell\text{-}\mathsf{IND\text{-}CPA}}_{\mathsf{TFHE}}(\mathcal{A}) := \left|\Pr[\mathsf{Expt}^{\ell\text{-}\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{TFHE}}(1^\lambda, 1^\kappa, n, t) = 1] - \frac{1}{2}\right| = \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\ell\text{-}\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{TFHE}}$ *is the experiment in Fig. 3 with* $\mathsf{ctr} = 0$ *at the beginning.*

$\mathsf{Expt}^{\ell\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathcal{A},\mathsf{TFHE}}(1^\lambda, 1^\kappa, n, t)$

$(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, ..., \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$
$S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| \leq t$
$m \leftarrow U(\mathcal{M})$
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
$m' \leftarrow \mathcal{A}_2^{\mathsf{OPartDec}}(\mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S}, \mathsf{ct})$
$\mathbf{return}\ m = m'$

$\mathsf{OPartDec}(C, m_1, \ldots, m_k)$

$\mathsf{ctr} = \mathsf{ctr} + 1$
$\mathbf{if}\ \mathsf{ctr} > \ell\ \mathbf{then\ return}\ \perp$
$\mathbf{if}\ (m_j)_j \notin \mathcal{M}^k\ \mathbf{then\ return}\ \perp$
$\mathbf{if}\ \mathsf{depth}(C) > \kappa\ \mathbf{then\ return}\ \perp$
$\mathsf{ct}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, m_j),\ \forall j \in [k]$
$\rho =$ randomness used for Enc
$\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$
$d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}), \quad i \in [n]$
$\mathbf{return}\ \rho, (d_i)_{i \in [n]}$

$\mathsf{Expt}^{\ell\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathcal{A},\mathsf{TFHE}}(1^\lambda, 1^\kappa, n, t)$

$(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, ..., \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$
$S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| \leq t$
$state \leftarrow \mathcal{A}_2^{\mathsf{OPartDec}}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S})$
$b \leftarrow U(\{0, 1\})$
$(m_0, m_1) \leftarrow \mathcal{A}_3(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S})$
$\mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}_4^{\mathsf{OPartDec}}(\mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S}, \mathsf{ct}_b)$
$\mathbf{return}\ b = b'$

Fig. 3. Experiments for ℓ-OW-CPA and ℓ-IND-CPA security of TFHE schemes.

Definition 3.9 (ℓ**-**$\mathsf{IND\text{-}CPA}$ **for** TPKE)**.** *We call a* TPKE *scheme* ℓ-$\mathsf{IND\text{-}CPA}$ *secure for the security parameter* λ*, the threshold parameters* n, t *and the query bound* ℓ*, if it is* ℓ-$\mathsf{IND\text{-}CPA}$ *secure as a* TFHE *scheme, where* $k = 1$ *and the only allowed circuit* $C: \mathcal{M} \to \mathcal{M}$ *is the identity. For a* PPT *adversary* $\mathcal{A}$ *we denote their advantage as* $\mathsf{Adv}^{\ell\text{-}\mathsf{IND\text{-}CPA}}_{\mathsf{TPKE}}(\mathcal{A})$*.*

In the TPKE case, this is the same definition as in Boneh et al. [Bon+18, Def. 8.27], where we specify a concrete bound ℓ on the number of queries.

4 From One-Wayness to Indistinguishability

4.1 For Weakly Robust Threshold Decryption

Hofheinz et al. [HHK17, Sec. 3.4] provide a tight reduction from $\mathsf{OW\text{-}CPA}$ security to $\mathsf{IND\text{-}CPA}$ security for standard PKE schemes in the random oracle model (ROM). In the following, we adapt the transformation to the threshold setting and show how a small modification allows to obtain a weakly chosen-ciphertext robust threshold scheme as in Definition 3.4.

The construction. The transformation is parameterized by $\delta \in \mathbb{N}$ which allows for a trade-off between the security loss of the reduction and the compactness of ciphertexts. Given $\mathsf{TPKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{PartDec}, \mathsf{Combine})$ with message space $\mathcal{M}$ being $\mathsf{OW\text{-}CPA}$ secure, we define $\mathsf{TPKE}' = (\mathsf{Setup}', \mathsf{Enc}', \mathsf{PartDec}', \mathsf{Combine}')$ with message space an abelian group $(\mathcal{M}', +)$, which fulfills $\mathsf{IND\text{-}CPA}$ security, as follows. Let $\mathsf{F}: \mathcal{M}^\delta \to \mathcal{M}'$ and $\mathsf{G}: \mathcal{M}^\delta \to \{0,1\}^{2\lambda}$ be two random oracles.

- Setup'**:** On input $(1^\lambda, n, t)$, it outputs $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$.
- Enc'**:** On input (pk, m) with $m \in \mathcal{M}'$, it samples $\mathbf{x} := (x_1, \ldots, x_\delta) \leftarrow U(\mathcal{M}^\delta)$ and sets $c_0 = m + \mathsf{F}(\mathbf{x})$ and $c_{\delta+1} = \mathsf{G}(\mathbf{x})$. Then, it computes $c_j \leftarrow \mathsf{Enc}(\mathsf{pk}, x_j)$ for $j \in [\delta]$ and outputs $\mathsf{ct} := (c_0, \ldots, c_{\delta+1})$.
- $\mathsf{PartDec}'$**:** On input $(\mathsf{sk}_i, \mathsf{ct})$ for some $i \in [n]$, it computes $d_{ij} \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, c_j)$ for all $j \in [\delta]$ and outputs $\mathbf{d}_i := (d_{ij})_{j \in [\delta]}$.
- $\mathsf{Combine}'$**:** On input $((\mathbf{d}_i)_{i \in S}, \mathsf{ct})$ with $\mathsf{ct} = (c_j)_{0 \le j \le \delta+1}$ and $\mathbf{d}_i = (d_{ij})_{j \in [\delta]}$, it computes $x'_j \leftarrow \mathsf{Combine}(\{d_{ij}\}_{i \in S}, c_j)$ for $j \in [\delta]$, sets $\mathbf{x}' = (x'_1, \ldots, x'_\delta)$ and computes $m' := c_0 - \mathsf{F}(\mathbf{x}')$. If $c_{\delta+1} = \mathsf{G}(\mathbf{x}')$ it outputs m'. Else, it outputs $\perp$.

Ciphertext expansion. The ratio between the bit size of the plaintext and the ciphertext is give by

$$\frac{|\mathsf{ct}|}{|m|} = \frac{|m| + \delta \cdot |c| + 2\lambda}{|m|},$$

where c is a ciphertext coming from TPKE. We can see that with larger δ the ciphertext expansion gets worse.

We prove the decryption correctness of the resulting scheme in Appendix C.1.

Lemma 4.1 (Weak Robustness). *The scheme* TPKE' *is weakly robust. More precisely, if there is a* PPT *adversary* $\mathcal{A}$ *such that* $\mathsf{Adv}^{\mathsf{w\text{-}cc\text{-}robust}}_{\mathsf{TPKE}'}(\mathcal{A}) \ge \varepsilon$ *for some* $\varepsilon > 0$*, then there exists a* PPT *adversary* $\mathcal{B}$ *breaking collision resistance of the random oracle* G *with probability at least* ε*.*

Proof. Fix λ, n and t. We show that if there exists a PPT adversary $\mathcal{A}$ that has advantage ε in the experiment defined in Figure 1, then there exists a PPT adversary $\mathcal{B}$ that finds a collision for the random oracle G with the same probability ε. Let $\mathcal{B}$ play the role of the challenger in the weak robustness game, running the Setup' algorithm on $(1^\lambda, n, t)$ and forwarding $(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]})$ to $\mathcal{A}$. Assume that $\mathcal{A}$ wins the weak robustness game by outputting two sets of decryption shares $\{\mathbf{d}_i\}_{i\in S}$ and $\{\mathbf{d}'_i\}_{i\in S'}$ such that $\mathsf{Combine}'(\{d_i\}_{i\in S}, \mathsf{ct}) \to m \neq m' \leftarrow \mathsf{Combine}'(\{\mathbf{d}'_i\}_{i\in S'}, \mathsf{ct})$ for the same ciphertext $\mathsf{ct} = (c_i)_{0\le i\le \delta+1}$, and neither m nor m' equals $\perp$. Let $\mathbf{x}, \mathbf{x}'$ denote the vectors recovered during the combining procedure. As $c_0 = m + \mathsf{F}(\mathbf{x}) = m' + \mathsf{F}(\mathbf{x}')$, $m \neq m'$ and F is deterministic, we can deduce that $\mathbf{x} \neq \mathbf{x}'$. This implies that $\mathsf{G}(\mathbf{x}) = c_{\delta+1} = \mathsf{G}(\mathbf{x}')$ for distinct $\mathbf{x} \neq \mathbf{x}'$ and hence $\mathcal{B}$ has found a collision in G. $\square$

Theorem 4.2 (Security). *Let $\delta, \ell \in \mathbb{N}$. If* TPKE *is* $(\ell\delta)$-OW-CPA *secure, then is* TPKE' ℓ-IND-CPA *secure in the* ROM*. More precisely, for any* ℓ-IND-CPA *adversary $\mathcal{A}$ that does at most q_F queries to the random oracle* F*, there exists an* $(\ell\delta)$-OW-CPA *adversary $\mathcal{B}$ with*

$$\mathsf{Adv}^{\ell\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{TPKE}'}(\mathcal{A}) \le q_F^{1/\delta} \cdot \mathsf{Adv}^{(\ell\delta)\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{TPKE}}(\mathcal{B}).$$

Note that the number of queries to G doesn't impact the tightness of the reduction as the output $\mathsf{G}(\mathbf{x})$ is completely independent of $\mathsf{F}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{M}^\delta$.

Proof. The proof closely follows the original proof by Hofheinz et al. [HHK17, Thm. 3.7]. The main modifications compared to the original proof are that $\mathcal{A}$ can query up to ℓ partial decryption outputs to some oracle $\mathsf{OPartDec}$ during the game and that we added a second random oracle G to obtain weak robustness.

Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4)$ be a PPT adversary against the ℓ-IND-CPA security of TPKE'. We consider two games G_0 and G_1 as described in Figure 4, where we specify the security game, the queries to the random oracles F and G and to the partial decryption oracle $\mathsf{OPartDec}$. The lists $\mathcal{L}_\mathsf{F}$ and $\mathcal{L}_\mathsf{G}$ are initialized as empty sets and the counter ctr is set to 0 at the beginning. Both games only differ in the way how queries to F are handled.

Game G_0. Note that Game G_0 is exactly the original ℓ-IND-CPA game (as in Def. 3.9) and hence $\mathsf{Adv}^{\ell\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{TPKE}'}(\mathcal{A}) = |\Pr[G_0(\mathcal{A}) = 1] - 1/2|$.

Game G_1. The only modification between game G_0 and G_1 is that we added line $3-5$ in the specification of F. More precisely, F raises a flag and aborts if it is queried by the vector $\mathbf{x}^*$ that is used for the challenge ciphertext ct^*. Hence, $|\Pr[G_0(\mathcal{A}) = 1] - \Pr[G_1(\mathcal{A}) = 1]| \le \Pr[flag]$. Now, as F aborts when queried on $\mathbf{x}^*$, the view of $\mathcal{A}$ is independent of the bit b chosen in the game. This implies that $\Pr[G_1(\mathcal{A}) = 1] = 1/2$, leading to $\mathsf{Adv}^{\ell\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{TPKE}'}(\mathcal{A}) \le \Pr[flag]$. The only thing left to do is to bound the latter probability. A direct adaptation of Lemma 3.8 in [HHK17] bounds this probability above by $q_F^{1/\delta} \cdot \mathsf{Adv}^{(\ell\delta)\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{TPKE}}(\mathcal{A})$. Here, the adversary $\mathcal{A}$ is embedded in $\mathcal{B}$'s own $(\ell\delta)$-OW-CPA security game and hence $\mathcal{B}$ takes care of simulating the random oracles F and G as well as the partial

Games G_0 and G_1

$(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, ..., \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$
$\mathbf{x}^* := (x_1^*, \ldots, x_\delta^*) \leftarrow U(\mathcal{M}^\delta)$
$S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| \leq t$
$\mathit{state} \leftarrow \mathcal{A}_2^{\mathsf{OPartDec}}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S})$
$b \leftarrow U(\{0,1\})$
$(m_0, m_1) \leftarrow \mathcal{A}_3(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S})$
$c_0^* = m_b + \mathsf{F}(\mathbf{x}^*)$
$c_j^* = \mathsf{Enc}(\mathsf{pk}, x_j^*)\ \forall j \in [\delta]$
$c_{\delta+1}^* = \mathsf{G}(\mathbf{x}^*)$
$\mathsf{ct}^* = (c_0^*, \ldots, c_{\delta+1}^*)$
$b' \leftarrow \mathcal{A}_4^{\mathsf{OPartDec}}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i \in S}, \mathsf{ct}^*)$
return $b = b'$

$\mathsf{F}(\mathbf{x})$

if $\exists r\colon (\mathbf{x}, r) \in \mathcal{L}_\mathsf{F}$
  then return r
if $\mathbf{x} = \mathbf{x}^*$ // G_1
  $\mathit{flag} = \mathsf{true}$ // G_1
  then return $\perp$ // G_1
$r \leftarrow U(\{0,1\}^\lambda)$
$\mathcal{L}_\mathsf{F} := \mathcal{L}_\mathsf{F} \cup \{(\mathbf{x}, r)\}$
return r

$\mathsf{OPartDec}'(m)$

$\mathsf{ctr} = \mathsf{ctr} + 1$
if $\mathsf{ctr} > \ell$ **then return** $\perp$
if $m \notin \mathcal{M}'$ **then return** $\perp$
$(c_j)_{0 \leq j \leq \delta+1} = \mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{pk}, m)$
$\rho =$ used randomness for Enc'
$\mathbf{d}_i \leftarrow \mathsf{PartDec}'(\mathsf{sk}_i, \mathsf{ct})\ \forall i \in [n]$
return $\rho, (\mathbf{d}_i)_{i \in [n]}$

$\mathsf{G}(\mathbf{x})$

if $\exists r\colon (\mathbf{x}, r) \in \mathcal{L}_\mathsf{G}$
  then return r
$r \leftarrow U(\{0,1\}^{2\lambda})$
$\mathcal{L}_\mathsf{G} := \mathcal{L}_\mathsf{G} \cup \{(\mathbf{x}, r)\}$
return r

Fig. 4. Games G_0 and G_1 for the proof of Theorem 4.2.

decryption queries to $\mathsf{OPartDec}'$. The latter is done by querying their own partial decryption oracle $\mathsf{OPartDec}$. Note that the increase from ℓ to $\ell\delta$ comes from the fact that $\mathcal{B}$ must do δ queries to $\mathsf{OPartDec}$ for every query to $\mathsf{OPartDec}'$ by $\mathcal{A}$. $\square$

4.2 For Fully Homomorphic Threshold Decryption

Whereas the reduction from above is simple and tight, it has the disadvantage of needing the random oracle F to mask the message m. When considering not only threshold PKE, but more generally threshold FHE, we need to make sure that we can homomorphically evaluate ciphertexts. The use of the random oracle F when computing $c_0 = m + \mathsf{F}(\mathbf{x})$ makes such an evaluation impossible, as there is no efficient circuit description of the random oracle F. We thus need another transformation which allows for homomorphic evaluation of ciphertexts. In the following, we describe a generic way of transforming a OW-CPA secure TFHE scheme into an IND-CPA secure one in the standard model, via hardcore bits.

The construction. The transformation is parameterized by $\delta, n \in \mathbb{N}$. Given $\mathsf{TFHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{PartDec}, \mathsf{Combine})$ with message space $\mathcal{M} = \{0,1\}^\gamma$ being OW-CPA secure, we define $\mathsf{TFHE}' = (\mathsf{Setup}', \mathsf{Enc}', \mathsf{Eval}', \mathsf{PartDec}', \mathsf{Combine}')$ with message space $\mathcal{M}' = \{0,1\}^\delta$, which fulfills IND-CPA security, as follows.

Setup': On input $(1^\lambda, 1^\kappa, n, t)$, it outputs $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$.

Enc': On input (pk, m) with $m = (m_j)_{j\in[\delta]} \in \mathcal{M}'$, it samples $x \leftarrow U(\mathcal{M})$ and computes $c_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$. For $j \in [\delta]$, it samples $s_j \leftarrow U(\mathcal{M})$ and computes $c_j = \langle x, s_j\rangle + m_j \bmod 2$. It outputs $\mathsf{ct} = (c_0, s_1, \ldots, s_\delta, c_1, \ldots, c_\delta)$.

Eval': On input $I := (\mathsf{pk}, C', \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$, where $\mathsf{ct}_i = (c_{i0}, s_{i1}, \ldots, s_{i\delta}, c_{i1}, \ldots, c_{i\delta})$ such that $c_{i0} \leftarrow \mathsf{Enc}(\mathsf{pk}, x_i)$ for $i \in [k]$ and $C' \colon (\mathcal{M}')^k \to \mathcal{M}'$, it first defines a circuit $C \colon (\mathcal{M})^k \to \mathcal{M}$ as follows:
- C takes as input $(x_1, \ldots, x_k)$ and has the information I hard-coded
- It computes $m_{ij} = c_{ij} + \langle x_i, s_{ij}\rangle \bmod 2$, for $j \in [\delta]$ and $i \in [k]$
- It outputs $C'(m_1, \ldots, m_k)$, where $m_i = (m_{ij})_{j\in[\delta]}$

It then outputs $\mathsf{ct}' = \mathsf{Eval}(\mathsf{pk}, C, c_{10}, \ldots, c_{k0})$.

$\mathsf{PartDec}'$: On input $(\mathsf{sk}_i, \mathsf{ct}')$, where ct' is from Eval', it outputs $d_i = \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$.

$\mathsf{Combine}'$: On input $(\{d_i\}_{i\in S}, \mathsf{ct}')$, it outputs $m = \mathsf{Combine}(\{d_i\}_{i\in S}, \mathsf{ct})$.

Ciphertext expansion. The ratio between the bit size of the plaintext and the ciphertext is give by

$$\frac{|\mathsf{ct}|}{|m|} = \frac{|c_0| + \delta(n+1)}{\delta},$$

where c_0 is the OW-CPA ciphertext encrypting γ bits coming from TFHE. We can see that with larger δ the ciphertext expansion gets better.

We prove compactness and decryption correctness in Appendix C.2.

Remark 4.3. One way to reduce the size of the ciphertext to $|c_0|+n+\delta$ (and hence to improve the ciphertext expansion) is to replace the δ random seeds $s_1, \ldots, s_\delta$ by one single seed and a random oracle F. More precisely, one could define $s_j :=$

$\mathsf{F}(r,j)$ for a random seed $r \leftarrow U(\mathcal{M})$ and $j \in [\delta]$. As a result, the transformation wouldn't be in the standard, but in the random oracle model. As the random oracle is only used to derive the seeds, not when masking the message, this transformation still applies to the threshold FHE setting.

Remark 4.4. Note that the reduction in the standard model restricted to TPKE, in contrast to the one from Section 4.1, doesn't satisfy weak robustness (Def. 3.4).

Theorem 4.5 (Security). *Fix $\ell, \gamma, \delta \in \mathbb{N}$ and $\varepsilon, \varepsilon' > 0$. Let* TFHE *be an ℓ-*OW-CPA *secure scheme with $\mathcal{M} = \{0,1\}^\gamma$, such that any* PPT *adversary $\mathcal{B}$ has advantage $\mathsf{Adv}_{\mathsf{TFHE}}^{\ell\text{-OW-CPA}}(\mathcal{B}) \leq 2^{-\gamma+\varepsilon'}$. Then,* TFHE′ *is ℓ-*IND-CPA *secure with $\mathcal{M}' = \{0,1\}^\delta$, where $\delta = \gamma - \varepsilon' - O(\log_2 1/\varepsilon)$ and for any* PPT *adversary $\mathcal{A}$ it yields*

$$\mathsf{Adv}_{\mathsf{TFHE}'}^{\ell\text{-IND-CPA}}(\mathcal{A}) \leq 5\varepsilon.$$

Parameters δ, γ and ε'. Since TFHE needs to be OW-CPA-secure, we need $\gamma \geq \lambda$ to achieve λ-bit security and avoid a trivial guessing attack on the γ-bit plaintext. The parameter ε' then measures how close TFHE is to achieving the best-possible one-way security. Smaller values of ε' lead to a smaller value of δ, which improves the ciphertext expansion in the scheme. On the other hand, a small ε' requires the TFHE parameters to be chosen according to a higher security level. The hidden constant in the big-O-notation from the theorem's statement depends on the Goldreich-Levin theorem. Note that the decryption correctness doesn't depend on δ.

Proof. The high level idea of the proof is to modify the experiment for ℓ-IND-CPA security of TFHE′ in such a way that the challenge ciphertext ct_b information-theoretically hides the selected message m_b. Hence, the adversary can only guess which message was encrypted.

Game G_0. We denote by G_0 the original ℓ-IND-CPA security game as described in Figure 3. Recall the definition of the concatenated Goldreich-Levin extractor $\mathcal{E}$ as given in Definition 2.2. It yields $\mathcal{E}\colon \{0,1\}^\gamma \times (\{0,1\}^\gamma)^\delta \to \{0,1\}^\delta \times (\{0,1\}^\gamma)^\delta$, where $\mathcal{E}(x, s_1, \ldots, s_\delta) = (\langle x, s_1\rangle, \ldots, \langle x, s_\delta\rangle, s_1, \ldots, s_\delta)$. We can thus rewrite the challenge ciphertext (in line 6 of Figure 3) as

$$\mathsf{ct}_b = \mathsf{Enc}'(\mathsf{pk}, m_b) = (\mathsf{Enc}(\mathsf{pk}, x), \mathcal{E}(x, s_1, \ldots, s_\delta) \oplus (m_b, 0, \ldots, 0)),$$

where $x, s_j \leftarrow U(\mathcal{M})$ for $j \in [\delta]$. For any PPT adversary $\mathcal{A}$ we denote their advantage by $\mathsf{Adv}_{\mathsf{TFHE}'}^{\ell\text{-IND-CPA}}(\mathcal{A}) = |\Pr[G_0(\mathcal{A}) = 1] - 1/2|$.

Game G_1. We denote by G_1 the game, where we change the ciphertext for the challenge message m_b. It is now computed as $\widetilde{\mathsf{ct}_b} := (\mathsf{Enc}(\mathsf{pk}, x), (r, s_1, \ldots, s_\delta) \oplus (m_b, 0, \ldots, 0)$, where $x, s_j \leftarrow U(\mathcal{M})$ and $r \leftarrow U(\mathcal{M}')$. Now, m_b is information-theoretically hidden and the adversary $\mathcal{A}$ can only guess the bit b and hence their advantage in this game is $|\Pr[G_1(\mathcal{A}) = 1] - 1/2| = 0$.

From G_0 *to* G_1. We finally show that the success probability of $\mathcal{A}$ in G_0 doesn't differ much from their success probability in G_1. By the probability preservation property, it yields $\Pr[G_0(\mathcal{A}) = 1] \leq \Pr[G_1(\mathcal{A}) = 1] + \mathsf{cdist}(G_0, G_1)$. We show in the following by using Lemma 2.3 that $\mathsf{cdist}(G_0, G_1)$ is bounded above by 5ε. More precisely, we set $X = U(\mathcal{M})$ and $Z = \mathsf{Enc}(\mathsf{pk}, X)$, the probability distribution that is defined by the randomized encryption algorithm for uniform random messages. Furthermore, we set $Y = X$, such that $\mathsf{cdist}((X, Z), (Y, Z)) \leq \varepsilon$ for all $\varepsilon > 0$. For any PPT algorithm $\mathcal{B}$ is yields

$$\Pr[\mathcal{B}(Z) = X] = \Pr_{x \leftarrow U(\mathcal{M})}[\mathcal{B}(\mathsf{Enc}(\mathsf{pk}, x)) = x] \leq \mathsf{Adv}^{\ell\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{TFHE}}(\mathcal{B}) \leq 2^{-\gamma+\varepsilon'}.$$

Thus, $H^{\mathsf{unp}}_{\varepsilon}(X|Z) \geq \gamma - \varepsilon'$ for any $\varepsilon > 0$. We now apply Lem. 2.3 which implies $\mathsf{cdist}(G_0, G_1) \leq \mathsf{cdist}(\mathsf{ct}_b, \widetilde{\mathsf{ct}_b}) \leq 5\varepsilon$, and thus $\mathsf{Adv}^{\ell\text{-}\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{TFHE}}(\mathcal{A}) \leq \mathsf{negl}(\lambda) + 5\varepsilon$. □

5 Threshold Fully Homomorphic Encryption From LWE With Polynomial Modulus

We now present our construction of a t-out-of-n TFHE scheme with OW-CPA security. First, we describe and analyze our main construction based on any LSSS with strong $\{0, 1\}$-reconstruction. Then, in Section 5.5, we give an alternative construction that combines pseudorandom secret sharing with Shamir sharing to improve efficiency when $\binom{n}{t}$ is small.

By applying the OW-CPA to IND-CPA transformation for TFHE from Section 4.2, we hence obtain an IND-CPA secure scheme. When we restrict ourselves to standard PKE, our construction gives us a standard TPKE scheme (cf. Def. 3.2). We can then also apply the transformation from Section 4.1, which gives as a weakly chosen-ciphertext robust IND-CPA secure scheme.

5.1 Nearly Linear Decryption of FHE

We use the following abstraction of LWE-based encryption schemes, where decryption is viewed as a linear function of the secret key that outputs a "noisy" version of the correct message. Similar notions were used in [BKS19; Bra+19].

Definition 5.1 (FHE with (β, ε)-linear decryption). *Let* $\mathsf{FHE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a fully-homomorphic encryption scheme (as in Def 3.3) with message space* $\mathcal{M} \subseteq R_p$ *and ciphertext space* R_q^r. *Suppose that* Setup *outputs a secret key* $\mathsf{sk} \in R_q^r$ *which has the form* $(1, \mathbf{s})$ *for some* $\mathbf{s} \in R_q^{r-1}$.

Let $\beta = \beta(\lambda) \in \mathbb{N}, \varepsilon = \varepsilon(\lambda) \in [0, 1]$. *We say that* FHE *has* (β, ε)-linear decryption *if for any* $\lambda, \kappa \in \mathbb{N}$, $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa)$, *depth-*$\kappa$ *circuit* $C\colon \mathcal{M}^k \to \mathcal{M}$, *messages* $m_1, \ldots, m_k \in R_p$, *ciphertexts* $\mathbf{c}_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i) \in R_q^r$ *and* $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, \mathbf{c}_1, \ldots, \mathbf{c}_k)$, *it holds that*

$$\langle \mathsf{sk}, \mathsf{ct} \rangle = \lfloor q/p \cdot C(m_1, \ldots, m_k) \rceil + e \mod q,$$

for some $e \in R_q$ *such that* $\Pr[\|e\|_\infty \leq \beta] \geq 1 - \varepsilon$ *(where the probability is taken over the randomness of* $\mathsf{Setup}, \mathsf{Enc}$ *and* Eval*).*

In standard (Module)-LWE based constructions, it's possible to securely set the parameters such that the ratio β/q can be made arbitrarily small, and as long as we have $\beta/q = 1/\mathsf{poly}(\lambda)$, then q is $\mathsf{poly}(\lambda)$.

For security, we require that FHE is IND-CPA secure.[7] This can be instantiated under the Module-LWE assumption to obtain (leveled) FHE using, for instance, the BGV scheme [BGV12] (with superpolynomial q). For $p = 2$, $d = 1$ and $R = \mathbb{Z}$, we also get (leveled) FHE under the standard LWE assumption with a polynomial modulus q [BV14].

5.2 Construction from LSSS with Strong $\{0,1\}$-Reconstruction

Our construction works over the ring $R = \mathbb{Z}[X]/f(X)$ for some degree-d irreducible polynomial f, and uses the following main ingredients:

- $\mathcal{D}_{\mathsf{flood}}$: a noise distribution over $\mathbb{Z}_q$ with magnitude bounded by β_{flood},
- $\mathcal{D}_{\mathsf{sim}}$: a noise distribution over $\mathbb{Z}_q$, where $\mathrm{RD}_a(\mathcal{D}_{\mathsf{sim}} \| \mathcal{D}_{\mathsf{flood}} + B) \leq \varepsilon_{\mathrm{RD}_a}$, for some $a \in (1, \infty), \varepsilon_{\mathrm{RD}_a} > 1$ and for all B with $|B| \leq \beta_{\mathsf{fhe}}$,
- LSS: a t-out-of-n linear secret sharing scheme $\mathsf{LSS} = (\mathsf{Share}, (\mathsf{Rec}_S)_{S \subset [n]})$ with strong $\{0,1\}$-reconstruction, associated parameters $L, \tau_{\mathsf{max}}, \tau_{\mathsf{min}}$ and shares in $\mathbb{Z}_q^L$ (cf. Def. 2.10),
- FHE: a OW-CPA secure $\mathsf{FHE} = (\mathsf{Setup}', \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ scheme with message space $\mathcal{M} \subseteq R_p$, ciphertext space R_q^r, and $(\beta_{\mathsf{fhe}}, \varepsilon)$-linear decryption for some $\beta_{\mathsf{fhe}} < q/(2p) - \tau_{\mathsf{min}}\beta_{\mathsf{flood}}$ and some negligible ε.

We now define the scheme $\mathsf{TFHE} := (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{PartDec}, \mathsf{Combine})$ by using Enc and Eval from the underlying FHE scheme and setting Setup, PartDec and Combine as specified in Figure 5. We prove its correctness in Appendix D.

For now, we assume the plaintext space $\mathcal{M} \subseteq R_p$ is superpolynomial in the security parameter, so that FHE is OW-CPA secure. In Section 5.3, we show how to extend this to use FHE with any plaintext space, which allows instantiating from LWE with polynomial modulus.

We write $\mathcal{D}_{\mathsf{flood}, R_q^r}$ (resp. $\mathcal{D}_{\mathsf{sim}, R_q^r}$) to refer to the distribution consisting of rd independent $\mathcal{D}_{\mathsf{flood}}$ (resp. $\mathcal{D}_{\mathsf{sim}}$) random variables, used to sample the coefficients of r elements of R_q.

We show security in the following.

Theorem 5.2 (Security). *For any adversary $\mathcal{A}$ against the ℓ-OW-CPA property of the TFHE scheme in Fig. 5, there exists an adversary $\mathcal{B}$ against the OW-CPA property of FHE, such that*

$$\mathsf{Adv}^{\ell\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{TFHE}}(\mathcal{A}) \leq \left(\mathsf{Adv}^{\mathsf{OW}\text{-}\mathsf{CPA}}_{\mathsf{FHE}}(\mathcal{B}) \cdot \varepsilon_{\mathrm{RD}_a}^{\ell d(nL - \tau_{\mathsf{max}})}\right)^{(a-1)/a} + \ell\varepsilon,$$

where L and τ_{max} are parameters from the LSS.

[7] In our main construction, we assume $\mathcal{M}$ is large and only rely on OW-CPA security of FHE. When extending to smaller $\mathcal{M}$ in Sec. 5.3, we instead need IND-CPA security.

$\mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$

$(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}'(1^\lambda, 1^\kappa)$
    // $\mathsf{sk} \in R_q^r, \mathsf{sk}_i \in (R_q^r)^L$
$(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{LSS.Share}(\mathsf{sk})$
return $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$

$\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$

$\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathsf{flood}, R_q}$ for $j \in [L]$
    // $\mathsf{sk}_i = (\mathsf{sk}_{i,1}, \ldots, \mathsf{sk}_{i,L}) \in (R_q^r)^L$
$\mathbf{d}_{i,j} \leftarrow \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle + \mathbf{e}_{i,j}$
return $\mathbf{d}_i \leftarrow (\mathbf{d}_{i,1}, \ldots, \mathbf{d}_{i,L})$

$\mathsf{Combine}(\{\mathbf{d}_i\}_{i \in S}, \mathsf{ct})$

$y \leftarrow \mathsf{Rec}_S((\mathbf{d}_i)_{i \in S})$
return $\lfloor (p/q) \cdot y \rceil$

Fig. 5. Setup, partial decrypt and combine algorithms for OW-CPA secure TFHE. The Enc and Eval algorithms are the same as for FHE.

Proof. The high-level idea is to modify the ℓ-OW-CPA game such that the t secret shares and the answers to the ℓ partial decryption queries provided to the adversary no longer depend on the underlying secret key sk. In this case, the game equals the standard OW-CPA game of FHE schemes.

Game G_0: This is the real threshold ℓ-OW-CPA experiment as in Figure 3. The view of $\mathcal{A}$ consists of the public parameters pp, the public key pk, challenge ciphertext ct, secret key shares $\{\mathsf{sk}_i\}_{i \in S}$ and results of up to ℓ partial decryption queries. In each query, $\mathcal{A}$ inputs a circuit C and set of messages $\mathbf{m} = (m_1, \ldots, m_k)$, and receives $(\rho, (\mathbf{d}_i)_{i \in [n]})$, where ρ is the randomness used to compute the ciphertexts $(\mathsf{ct}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, m_j))_{j=1}^k$, and $\mathbf{d}_i$ is the partial decryption of $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ under sk_i. It yields, $\mathsf{Adv}_{\mathsf{TFHE}}^{\ell\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}(\mathcal{A}) = \mathsf{Adv}_{\mathsf{TFHE}}^{G_0}(\mathcal{A})$.

Game G_1: In this game, we redefine how the partial decryptions are computed. After the adversary chooses the set $S \subset [n]$ of corrupt parties, let $S_L = \{(i,j)\}_{i \in S, j \in [L]}$ be the corresponding set of share elements. Fix $T \supseteq S_L$ to be a maximal invalid set of share elements. Then, compute the partial decryptions $\mathbf{d}_i$ for a ciphertext ct as follows:

1. For $(i,j) \in T$, let $\tilde{\mathbf{d}}_{i,j} = \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle$;
2. For $(i,j) \in ([n] \times [L]) \setminus T$, let $T_{i,j} \subseteq T \cup \{(i,j)\}$ be a minimal valid set of share elements, and compute $\tilde{\mathbf{d}}_{i,j} = \langle \mathsf{ct}, \mathsf{sk} \rangle - \sum_{(k,l) \in T_{i,j} \setminus \{(i,j)\}} \tilde{\mathbf{d}}_{k,l}$;
3. Sample $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathsf{flood}, R_q^L}$ and compute $\mathbf{d}_i = \tilde{\mathbf{d}}_i + \mathbf{e}_i$, for $i \in [n]$.

Note that the view of $\mathcal{A}$ in G_1 is identical to that in G_0, due to the strong $\{0,1\}$-reconstruction property of LSS. This is because every share belonging to the maximally invalid set T is computed the same way as in G_0, using the shares sk_i, while each share outside this set is deterministically fixed to be a sharing of the correct secret $\langle \mathsf{ct}, \mathsf{sk} \rangle$, plus noise sampled from $\mathcal{D}_{\mathsf{flood}}$, as in G_0. Hence, $\mathsf{Adv}_{\mathsf{TFHE}}^{G_0}(\mathcal{A}) = \mathsf{Adv}_{\mathsf{TFHE}}^{G_1}(\mathcal{A})$.

Game G_2: In this game, before outputting the partial decryptions for a ciphertext ct, we first check that $\langle \mathsf{ct}, \mathsf{sk} \rangle = \lfloor q/p \rceil \cdot C(m_1, \ldots, m_k) + e$ for some e

with $\|e\|_\infty \leq \beta_{\mathsf{fhe}}$. If not, the game aborts. Due to the $(\beta_{\mathsf{fhe}}, \varepsilon)$-linear decryption property of FHE, and applying a union bound over the ℓ queries, we have that $\mathsf{Adv}^{G_1}_{\mathsf{TFHE}}(\mathcal{A}) \leq \mathsf{Adv}^{G_2}_{\mathsf{TFHE}}(\mathcal{A}) + \ell\varepsilon$.

Game G_3: We replace the partial decryptions corresponding to shares outside of T with simulated ones. Firstly, in step (2) above, for $(i,j) \in ([n]\times[L])\setminus T$, we now compute $\tilde{\mathbf{d}}_{i,j}$ as $\tilde{\mathbf{d}}_{i,j} = \lfloor q/p \cdot C(m_1,\ldots,m_k)\rceil - \sum_{(k,l)\in T_{i,j}\setminus\{(i,j)\}} \tilde{\mathbf{d}}_{k,l}$.

Secondly, in step (3), instead of always sampling $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathsf{flood},R_q}$, we only sample $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathsf{flood},R_q}$ if $(i,j) \in T$, and $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathsf{sim},R_q}$ otherwise.

Game G_4. In the final game, we change how the secret key shares are sampled: pick $(\mathsf{sk}'_1,\ldots,\mathsf{sk}'_n) \leftarrow \mathsf{LSS.Share}(0)$ and give to $\mathcal{A}$ the shares $\{\mathsf{sk}'_i\}_{i\in S}$.

This is perfectly indistinguishable from G_3, by the perfect privacy property of LSS, hence $\mathsf{Adv}^{G_3}_{\mathsf{TFHE}}(\mathcal{A}) = \mathsf{Adv}^{G_4}_{\mathsf{TFHE}}(\mathcal{A})$. Note also that for any adversary $\mathcal{A}$ against game G_4, there exists an adversary $\mathcal{B}$ for the $\mathsf{OW\text{-}CPA}$ property of FHE with the same success probability as $\mathcal{A}$: $\mathsf{Adv}^{G_4}_{\mathsf{TFHE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{FHE}}(\mathcal{B})$.

The theorem then follows from the following lemma.

Lemma 5.3. *For any adversary $\mathcal{A}$ in Games G_2 and G_3, it holds that*

$$\mathsf{Adv}^{G_2}_{\mathsf{TFHE}}(\mathcal{A}) \leq (\mathsf{Adv}^{G_3}_{\mathsf{TFHE}}(\mathcal{A}) \cdot \varepsilon_{\mathrm{RD}_a}^{\ell d(nL-\tau_{\mathsf{max}})})^{(a-1)/a},$$

where τ_{max} is the size of the smallest maximal invalid share set in LSS.

Proof. We compute the Rényi divergence between the views of the adversary in each game. Each view consists of the adversary's random tape and the values

$$\left(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in S}, \{C^\eta, \mathbf{m}^\eta, \rho^\eta, (\mathbf{d}_i^\eta)_{i\in[n]}\}_{\eta\in[\ell]}\right),$$

where $C^\eta, \mathbf{m}^\eta = (m_1^\eta,\ldots,m_k^\eta)$ are circuit and messages chosen by $\mathcal{A}$ in the η-th query. Let D_2 and D_3 denote the distributions of the above values in games G_2 and G_3, respectively. Since the partial decryption queries are adaptive, note that the circuit C^η and message $\mathbf{m}^\eta$ depend on the previous queries $(\mathbf{m}^{\eta-1},\ldots,\mathbf{m}^1)$ and their corresponding responses $(\mathbf{d}_i^{\eta-1},\ldots,\mathbf{d}_i^1)_{i\in[n]}$. However, since each $\mathbf{m}^\eta$ is a deterministic function of the other values in the view (including the random tape), by the data processing inequality (Lem. 2.5), $\mathrm{RD}_a(D_2\|D_3) \leq \mathrm{RD}_a(D'_2\|D'_3)$, where D'_2, D'_3 are the distributions with the $C^\eta, \mathbf{m}^\eta$ values removed. D'_2 are D'_3 are now defined identically, except in the way the partial decryption components $\mathbf{d}^\eta_{i,j}$ are computed for indices $(i,j)\notin T$. In G_2, $\mathbf{d}^\eta_{i,j}$ is computed using (amongst other values) $\langle \mathsf{ct}^\eta, \mathsf{sk}\rangle + \mathcal{D}_{\mathsf{flood},R_q}$, whereas G_3 instead uses $\lfloor q/p\cdot C(\mathbf{m}^\eta)\rceil + \mathcal{D}_{\mathsf{sim},R_q}$. Since $\langle \mathsf{ct}^\eta, \mathsf{sk}\rangle = \lfloor q/p\cdot C(\mathbf{m}^\eta)\rceil + e_\eta$ for some e_η with $\|e_\eta\|_\infty \leq \beta_{\mathsf{fhe}}$, and the view contains $nL - |T|$ pairs $(i,j)\notin T$ where the sampling of $\mathbf{d}^\eta_{i,j}$ changes from G_2 to G_3, to compute $\mathrm{RD}_a(D'_2\|D'_3)$, it suffices to compute

$$\mathrm{RD}_a\left(((e_1+\mathcal{D}_{\mathsf{flood},R_q})^{nL-|T|},\ldots,(e_\ell+\mathcal{D}_{\mathsf{flood},R_q})^{nL-|T|})\|\mathcal{D}_{\mathsf{sim},R_q}^{\ell(nL-|T|)}\right).$$

Applying Lem. 2.7 with $N = d\ell(nL-|T|)$, $D_1 = \mathcal{D}_{\mathsf{flood}}, D_2 = \mathcal{D}_{\mathsf{sim}}$, we get

$$\mathrm{RD}_a(D'_2\|D'_3) \leq \varepsilon_{\mathrm{RD}_a}^{d\ell(nL-|T|)}.$$

Applying the probability preservation property of Rényi divergence, we bound the success probability of the adversary as required. ■

□

5.3 Supporting a Larger Plaintext Space

The above construction works for a plaintext space $\mathcal{M} \subseteq R_p$. Since we only obtain one-way security, this requires $|R_p|$ to be superpolynomial in λ to give a meaningful security guarantee. If R_p is small, we can easily modify our threshold scheme to still be secure by using several ciphertexts to encrypt larger messages with the underlying FHE scheme. Note that this change is necessary to obtain an instantiation from LWE with polynomial modulus, since there $\mathcal{M} = R_p = \mathbb{Z}_2$.

Concretely, suppose that FHE is $\mathsf{IND\text{-}CPA}$ secure and has small message space $\mathcal{M}$. Define FHE' with message space $\mathcal{M}^k$, such that $\left|\mathcal{M}^{-k}\right|$ is negligible, by encrypting each of the k message components separately under FHE. We then instantiate our threshold scheme using FHE' instead of FHE, where during the partial decrypt and combine steps, we run the algorithms for the previous construction on each component separately. If FHE is $\mathsf{IND\text{-}CPA}$ secure, then so is FHE', and the proof carries over in the same way, except that the ℓ values in the statement of Theorem 5.2 will be replaced with $k\ell$, to account for the fact that each of the ℓ decryption queries involves k decryptions of ciphertexts from FHE.

5.4 Bounding the Rényi Divergence

We now analyze parameters and instantiate the distributions $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$. For now, we simply choose them both to be rounded Gaussian distributions $\lfloor D_\sigma \rceil$ with the same standard deviation σ. In Sec. 6.1, we obtain tighter parameters by carefully optimizing the choice of distributions. If FHE has a maximum ciphertext noise bound of β_{fhe}, then using Lem. 2.6 with our choice of distributions, we get $\varepsilon_{\mathrm{RD}_a} = \mathrm{RD}_a(\mathcal{D}_{\mathsf{flood}} + \beta_{\mathsf{fhe}} \| \mathcal{D}_{\mathsf{sim}}) \leq \exp\left(\frac{a\beta_{\mathsf{fhe}}^2}{2\sigma^2}\right)$. If FHE has λ_{FHE} bits of security, then from Thm. 5.2, the resulting TFHE scheme is λ_{TFHE}-bit secure, such that

$$\lambda_{\mathsf{TFHE}} \geq (\lambda_{\mathsf{FHE}} - \ell d(nL - \tau_{\mathsf{max}}) \log_2 \varepsilon_{\mathrm{RD}_a}) \, \frac{a-1}{a} \tag{1}$$

Combining the above two equations, we obtain $\lambda_{\mathsf{TFHE}} \geq \frac{a-1}{a}\lambda_{\mathsf{FHE}} - \ell d(nL - \tau_{\mathsf{max}})(a-1)\frac{\beta_{\mathsf{fhe}}^2}{2\sigma^2} \log_2 e$. Setting for instance $a = \lambda_{\mathsf{TFHE}}$, and choosing $\sigma, q, \beta_{\mathsf{fhe}}$ such that $\sigma = O(\beta_{\mathsf{fhe}}\sqrt{\ell d(nL - \tau_{\mathsf{max}})(a-1)})$ while decryption is still correct, the loss in security is only a constant factor. Smaller values of a give different tradeoffs between the size of σ and the security loss. Note that in any case, if ℓ and nL are polynomially bounded then both σ and the modulus q can be also.

5.5 Alternative Construction Using Pseudorandom Secret Sharing

We also give a different construction based on pseudorandom secret sharing (PRSS), which improves upon the previous one in some aspects. Instead of having each party perturb their share by an independent, random noise term, we will use PRSS [GI99; CDI05]. This allows them to jointly sample replicated secret sharings of small noise terms, without interaction, after a one-time setup that distributes PRF keys. We also exploit the fact that replicated secret shares can be locally converted to any other LSS, and convert the secret shared noise terms into Shamir sharings before using them for partial decryption. These means that the partial decryptions are Shamir shares, which are much smaller, consisting of only 1 element over R_q each. Furthermore, this leads to improved parameters in the security reduction, and we can additionally take advantage of the error-correction capability of Shamir to achieve strong robustness (Def. 3.5) when $t < n/3$. This offers a way of getting robustness for TFHE instead of only TPKE with our previous transformations, with the drawback that we require $\binom{n}{t}$ to be not too large, due to using replicated secret sharing.

The details and security proof of this construction are in Appendix E.

6 Sample Parameters and Security Estimates

In this section, we discuss how to choose concrete parameters for our $\mathsf{OW\text{-}CPA}$ secure threshold construction, where we take as a starting point the lattice-based scheme Kyber [Sch+20]. Hence, we are not in the fully-homomorphic case, but in the standard PKE case and thus obtain a standard TPKE scheme. We denote the thresholdized version of Kyber by TKyber.

After deriving sample parameter sets in Section 6.1, we give in Section 6.2 an attack if the adversary has access to sufficiently many partial decryptions. We will see that the bound is close to the one obtained in Section 5, showing that using the Rényi divergence leads to almost optimal results.

We recall the high level description of Kyber in App. F. The relevant parameters for Kyber are the ring degree d, the rank r, the modulus q and the two distribution parameter η. Whereas the specifications of Kyber only consider three parameter sets, called $\mathsf{Kyber}512$, $\mathsf{Kyber}768$, $\mathsf{Kyber}1024$, we additionally consider three more parameter sets, that we subsequently call $\mathsf{Kyber}1280$, $\mathsf{Kyber}1536$ and $\mathsf{Kyber}1792$. As the name suggest, they are obtained in a similar manner as the previous parameter sets, simply by increasing the rank by $+1$. All parameter sets are summarized in Table 4 in Appendix F.

6.1 Security From the Reduction

Let λ_{PKE} (resp. λ_{TPKE}) denote the security level of the starting PKE (resp. the resulting TPKE) from Theorem 5.2. Further, we set $\Delta_\lambda := \lambda_{\mathsf{PKE}} - \lambda_{\mathsf{TPKE}}$, which describe the security loss in our reduction. Instantiating Equation 1 in the standard PKE setting yields

$$\lambda_{\mathsf{TPKE}} \geq \frac{a-1}{a} \cdot \left(\lambda_{\mathsf{PKE}} - \ell d(nL - \tau_{\mathsf{max}}) \log_2 \varepsilon_{\mathrm{RD}_a}\right), \tag{2}$$

where ℓ is the number of partial decryption queries, d the degree of the ring R, L and τ_{max} parameters of the underlying LSSS and $\varepsilon_{\mathrm{RD}_a}$ an upper bound on the Rényi divergence $\mathrm{RD}_a(\mathcal{D}_{\mathsf{sim}} \| \mathcal{D}_{\mathsf{flood}} + \beta_{\mathsf{pke}})$ of order a. Here, $\mathcal{D}_{\mathsf{sim}}$ (resp. $\mathcal{D}_{\mathsf{flood}}$) denotes the simulating (resp. flooding) noise distribution and β_{pke} is a bound on the decryption noise that depends on the concrete parameters of Kyber, in particular on the ring degree d, the module rank r and the parameters η and η, as well as the maximal failure probability ε we want to achieve. For concreteness we set λ_{PKE} as the core-SVP classical hardness, i.e., the resulting BKZ block estimated from the Lattice Estimator [APS15] size multiplied by 0.292.

Table 2 and Table 3 present some sample parameters. We explain in Appendix F in more details how we concretely derived them. The relevant difference between the two is that in the first table, we focus on larger numbers of parties n and samples ℓ while accepting a modulus of up to 39 bits. For simplicity, we assume that both $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$ follow a Gaussian distribution of width σ. In contrast, in the second table we fine-tuned the flooding and simulation distributions so that we can allow for very small q (only multiplying the original Kyber modulus by small constants up to 10).

Table 2. Sample parameters and security estimates following the reduction from Thm. 5.2 using a generic approach.

Set	$(\beta_{\mathsf{pke}}, \varepsilon)$	n	t	ℓ	$\lceil \log_2 \sigma \rceil$	$\lceil \log_2 q \rceil$	λ_{PKE}	λ_{TPKE}	Δ_λ
TKyber1024	$(390, 2^{-60})$	2	1	1	17	23	120	117	3
TKyber1024	$(934, 2^{-300})$	2	1	1	18	24	111	108	3
TKyber1024	$(390, 2^{-60})$	10	9	1	17	25	105	102	3
TKyber1280	$(435, 2^{-60})$	10	5	1	21	29	120	117	3
TKyber1536	$(476, 2^{-60})$	20	10	10	27	36	112	109	3
TKyber1792	$(513, 2^{-60})$	2	1	2^{32}	33	39	123	120	3

Table 3. Sample parameters and security estimates following the reduction from Thm. 5.2 obtained from a hand-tuned Python program.

Set	q	n	t	ℓ	$\mathcal{D}_{\mathsf{flood}}$	$\mathcal{D}_{\mathsf{sim}}$	λ_{TPKE}	Δ_λ
TKyber1024	$5 \cdot 3329$	2	1	1	947	1087	100	111
TKyber1024	$10 \cdot 3329$	2	1	2	1994	2034	104	91
TKyber1024	$9 \cdot 3329$	3	2	1	1197	1297	106	92

6.2 Statistical Attack

In the following, we describe an attack against our proposed threshold decryption scheme if the adversary obtains sufficiently many partial decryption queries. Note that the obtained lower bound on the samples for this attack is only slightly higher than the upper bound for security from Section 5. This shows that using the Rényi divergence leads to quasi optimal parameters.

As in the previous section, we focus on Kyber and denote by TKyber the thresholdized scheme as in Section 5. For simplicity, we consider the full-threshold setting for n parties using additive secret sharing. We use as flooding noise distribution a rounded Gaussian $\lfloor \mathcal{D}_{\mathsf{flood},R_q} \rceil$ of width σ_{flood}.

Lemma 6.1. *Let q, d, r, η be the* Kyber *parameters (as introduced in App. F). Further, let ℓ denote the number of partial decryption queries to* TKyber *an adversary $\mathcal{A}$ has access to. Further, let $\nu \in \mathbb{N}$. If*

$$\ell d = \Omega((2r+1)d + \nu) \quad \text{and} \quad \ell d = \Omega\left(\frac{\sigma_{\mathsf{flood}}^2}{\eta^2} \log_2(2d(2r+1))\right),$$

then $\mathcal{A}$ can recover the secret key of TKyber *with probability* $1-1/2d(2r+1)-2^{-\nu}$.

Proof. As we use additive secret sharing, every party receives exactly one secret key share sk_i, where $\mathsf{sk} = \sum_{i=1}^n \mathsf{sk}_i$.

Following the description of Kyber from App. F and the threshold function from Figure 5, a partial decryption of TKyber is of the form $d = (d_i)_{i \in [n]}$, with

$$d_i = v \cdot 1_i - \mathbf{u}^T \mathbf{s}_i + e_i,$$

where 1_i is a share of 1 (e.g. $1_i = 1$ if $i = 1$ and 0 otherwise) and $e_i \leftarrow \mathcal{D}_{\mathsf{flood},R_q}$.

Without loss of generality, we say that Party 1 is honest and all other parties are controlled by the adversary $\mathcal{A}$. After receiving all n decryption shares, the adversary can sum them up to obtain

$$\sum_{i=1}^n d_i = \mathbf{r}^T \mathbf{e} - \mathbf{e}_1^T \mathbf{s} + e_2 + \lfloor q/2 \rceil m + \sum_i e_i,$$

where $(\mathbf{r}, \mathbf{e}_1, e_2)$ is the encryption randomness used for this query.

We can re-write $\sum_i d_i = \langle \mathbf{w}, \mathbf{z} \rangle + \lfloor q/2 \rceil m + \sum_i e_i$, where $\mathbf{w} = (\mathbf{r}, \mathbf{e}_1, e_2)^T \leftarrow \mathsf{CBD}_\eta^{(2r+1)d}$ and $\mathbf{z} = (\mathbf{e}, -\mathbf{s}, 1)^T$.

After subtracting $\lfloor q/2 \rceil m$, the adversary obtains $d' = \langle \mathbf{w}, \mathbf{z} \rangle + \sum_{i=1}^n e_i$. Moreover, the adversary knows the flooding noise of the corrupted parties and can further subtract it from d', leading to $d'' = \langle \mathbf{w}, \mathbf{z} \rangle + e_1$.

Interestingly, we observe that all elements appearing in the equation of d'' are of small norm, thus no reduction modulo q is necessary. After applying the coefficient embedding, we can interpret d'' as d samples of I-LWE as defined in Section 2.3. Due to the concrete shape of $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ in Kyber, the resulting public matrix $\mathbf{W}$ of the I-LWE instance is now the concatenation of nega-cyclic matrices over $\mathbb{Z}_q$. Overall, after ℓ partial decryption queries, the adversary has seen an instance of the I-LWE distribution of parameters $R := (2r+1)d$ and $M := \ell d$ with underlying secret $\mathbf{z} \in \mathbb{Z}^R$. Recall that in TKyber, the distribution of $\mathbf{w}$ is given by a centered binomial distribution of parameter η, defining a η-subgaussian distribution with $\sigma_w = \sqrt{\mathbb{E}[\chi_e^2]} \leq \sqrt{\eta^2} = \eta$. The noise follows a rounded Gaussian distribution, is thus σ_{flood}-subgaussian. Thus, Theorem 2.15 leads to an attacker with success probability $1 - 1/2R - 2^{-\nu}$ if $M = \Omega((2r+1)d + \nu)$ and $M = \Omega\left(\frac{\sigma^2_{\mathsf{flood}}}{\eta^2}\log_2(2d(2r+1))\right)$. Here we use that the least square method performs for $\mathbf{W}$ (with the nega-cyclic structure) as good as for matrices where every entry is independent of all the others. That is the case, as the nega-cyclic structure preserves the required properties to prove Theorem 2.15. $\square$

In comparison, in Section 5.4 we require $M = \ell d = O\left(\frac{\sigma^2_{\mathsf{flood}}}{\beta^2_{\mathsf{fhe}}}\right)$. Recall that β_{fhe} is the bound on the ciphertext noise, which depends on the decryption failure probability one wants to tolerate. Some concrete parameters for TKyber are given in Table 2. In all cases, $\beta_{\mathsf{fhe}} \geq \eta / \log_2(2d(2r+1))$ and hence our upper bound from Section 5 is below the lower bound from the attack.

Note that [ASY22] showed that the Rényi divergence in their threshold signature leads to optimal bounds by providing an attack for larger bounds. As they use a deterministic signature scheme, their analysis boils down to a straight forward averaging attack. In our case, we argue with the results on Integer LWE, using the least square method, as our encryption scheme is randomized.

Acknowledgments

We would like to thank Ivan Damgård and Aayush Jain for helpful discussions on secret sharing.

This work has been supported by the Independent Research Fund Denmark (DFF) under project number 0165-00107B (C3PO), and the Aarhus University Research Foundation.

References

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. *On The Concrete Hardness Of Learning With Errors*. Cryptology ePrint Archive, Report 2015/046. https://eprint.iacr.org/2015/046. 2015.

[Ash+12] Gilad Asharov et al. “Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE”. In: *EUROCRYPT 2012*. Apr. 2012.

[ASY22] Shweta Agrawal, Damien Stehle, and Anshu Yadav. *Round-Optimal Lattice-Based Threshold Signatures, Revisited*. Cryptology ePrint Archive, Report 2022/634. https://eprint.iacr.org/2022/634. 2022.

[Bai+18] Shi Bai et al. “Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather than the Statistical Distance”. In: *Journal of Cryptology* 2 (Apr. 2018).

[BBH06] Dan Boneh, Xavier Boyen, and Shai Halevi. “Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles”. In: *CT-RSA 2006*. Feb. 2006.

[BD10] Rikke Bendlin and Ivan Damgård. "Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems". In: *TCC 2010*. Feb. 2010.

[BD20a] Zvika Brakerski and Nico Döttling. "Hardness of LWE on General Entropic Distributions". In: *EUROCRYPT 2020, Part II*. May 2020.

[BD20b] Zvika Brakerski and Nico Döttling. "Lossiness and Entropic Hardness for Ring-LWE". In: *TCC 2020, Part I*. Nov. 2020.

[Bei96] Amos Beimel. "Secure schemes for secret sharing and key distribution". PhD thesis. Technion, Haifa, Israel, 1996. URL: https://www.cs.bgu.ac.il/~beimel/Papers/thesis.pdf.

[Ben+21] Fabrice Benhamouda et al. "Multiparty Reusable Non-interactive Secure Computation from LWE". In: *EUROCRYPT 2021, Part II*. Oct. 2021.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ITCS 2012*. Jan. 2012.

[BKS19] Elette Boyle, Lisa Kohl, and Peter Scholl. "Homomorphic Secret Sharing from Lattices Without FHE". In: *EUROCRYPT 2019, Part II*. May 2019.

[BL90] Josh Cohen Benaloh and Jerry Leichter. "Generalized Secret Sharing and Monotone Functions". In: *CRYPTO'88*. Aug. 1990.

[Bon+17] Dan Boneh et al. *A Lattice-Based Universal Thresholdizer for Cryptographic Systems*. Cryptology ePrint Archive, Report 2017/251. https://eprint.iacr.org/2017/251. 2017.

[Bon+18] Dan Boneh et al. "Threshold Cryptosystems from Threshold Fully Homomorphic Encryption". In: *CRYPTO 2018, Part I*. Aug. 2018.

[Boo+18] Jonathan Bootle et al. "LWE Without Modular Reduction and Improved Side-Channel Attacks Against BLISS". In: *ASIACRYPT 2018, Part I*. Dec. 2018.

[Bou+20] Katharina Boudgoust et al. "Towards Classical Hardness of Module-LWE: The Linear Rank Case". In: *ASIACRYPT 2020, Part II*. Dec. 2020.

[Bra+19] Zvika Brakerski et al. "Leveraging Linear Decryption: Rate-1 Fully-Homomorphic Encryption and Time-Lock Puzzles". In: *TCC 2019, Part II*. Dec. 2019.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. "Lattice-based FHE as secure as PKE". In: *ITCS 2014*. Jan. 2014.

[Cas+22] Leo de Castro et al. "Asymptotically Quasi-Optimal Cryptography". In: *Advances in Cryptology – EUROCRYPT 2022*. 2022. ISBN: 978-3-031-06944-4.

[CDI05] Ronald Cramer, Ivan Damgård, and Yuval Ishai. "Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation". In: *TCC 2005*. Feb. 2005.

[Cho+22] Siddhartha Chowdhury et al. "Efficient FHE with Threshold Decryption and Application to Real-Time Systems". In: *Cryptology ePrint Archive* (2022).

[Dac+20] Dana Dachman-Soled et al. "LWE with Side Information: Attacks and Concrete Security Estimation". In: *CRYPTO 2020, Part II*. Aug. 2020.

[Dev+21] Julien Devevey et al. "Non-interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings". In: *PKC 2021, Part I*. May 2021.

[EH14] Tim van Erven and Peter Harremoës. "Rényi Divergence and Kullback-Leibler Divergence". In: *IEEE Trans. Inf. Theory* 7 (2014).

[GAL13] Manuel Gil, Fady Alajaji, and Tamas Linder. "Rényi divergence measures for commonly used univariate continuous distributions". In: *Information Sciences* (2013).

[GI99] Niv Gilboa and Yuval Ishai. "Compressing Cryptographic Resources". In: *CRYPTO'99*. Aug. 1999.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. "A Modular Analysis of the Fujisaki-Okamoto Transformation". In: *TCC 2017, Part I*. Nov. 2017.

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. "Conditional Computational Entropy, or Toward Separating Pseudoentropy from Compressibility". In: *EUROCRYPT 2007*. May 2007.

[HMP06] Shlomo Hoory, Avner Magen, and Toniann Pitassi. "Monotone circuits for the majority function". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. 2006.

[HV22] Loïs Huguenin-Dumittan and Serge Vaudenay. "On IND-qCCA Security in the ROM and Its Applications - CPA Security Is Sufficient for TLS 1.3". In: *EUROCRYPT 2022, Part III*. 2022.

[ISN89] Mitsuru Ito, Akira Saito, and Takao Nishizeki. "Secret sharing scheme realizing general access structure". In: *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 9 (1989).

[JRS17] Aayush Jain, Peter M. R. Rasmussen, and Amit Sahai. *Threshold Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2017/257. https://eprint.iacr.org/2017/257. 2017.

[Kra+19] Michael Kraitsberg et al. "Adding Distributed Decryption and Key Generation to a Ring-LWE Based CCA Encryption Scheme". In: *ACISP 19*. July 2019.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 3 (2015). URL: https://doi.org/10.1007/s10623-014-9938-4.

[Lyu+20] Vadim Lyubashevsky et al. *CRYSTALS-DILITHIUM*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. National Institute of Standards and Technology, 2020.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *37th ACM STOC*. May 2005.

[Ros20] Mélissa Rossi. "Extended Security of Lattice-Based Cryptography. (Sécurité étendue de la cryptographie fondée sur les réseaux euclidiens)". PhD thesis. Paris Sciences et Lettres University, France, 2020. URL: https://www.di.ens.fr/~mrossi/docs/thesis.pdf.

[Sch+20] Peter Schwabe et al. *CRYSTALS-KYBER*. Tech. rep. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. National Institute of Standards and Technology, 2020.

[Shi22] Sina Shiehian. "mrNISC from LWE with Polynomial Modulus". In: *Security and Cryptography for Networks*. 2022. ISBN: 978-3-031-14791-3.

[Val84] Leslie G. Valiant. "Short monotone formulae for the majority function". In: *Journal of Algorithms* 3 (1984).

Appendix A Missing Preliminaries

A.1 Proof of Lemma 2.7

Proof. We apply N times the multiplicativity property of the Rényi divergence as follows. Let $P = (D_1 + e_N, \ldots, D_1 + e_1)$ and $Q = D_2^N$. Our goal is to bound $\mathrm{RD}_a(P\|Q)$. We start with setting their marginal distributions as $P_1 = (D_1 + e_{N-1}, \ldots, D_1 + e_1)$, $Q_1 = D_2^{N-1}$, $P_2 = D_1 + e_N$ and $Q_2 = D_2$. For $j \in [N]$, let E_j denote the random variable given by the distribution $D_1 + e_j$. By Lemma 2.5, it yields

$$\begin{aligned}
\mathrm{RD}_a(P\|Q) &\leq \mathrm{RD}_a(P_1\|Q_1) \cdot \max_{y_1 \in Y_1} \mathrm{RD}_a(D_1 + e_N | Y_1 = y_1 \| D_2 | Y_1 = y_1) \\
&\leq \mathrm{RD}_a(P_1\|Q_1) \cdot \max_{y_1 \in Y_1} \mathrm{RD}_a(D_1 + \beta | Y_1 = y_1 \| D_2 | Y_1 = y_1) \\
&\leq \mathrm{RD}_a(P_1\|Q_1) \cdot \mathrm{RD}_a(D_1 + \beta \| D_2) \\
&\leq \rho \cdot \mathrm{RD}_a(P_1\|Q_1),
\end{aligned}$$

where β is such that $|\beta| \leq B$ and $Y_1 = (E_{N-1}, \ldots, E_1)$. From line 2 to line 3 we used the fact that neither $D_1 + \beta$ nor D_2 depend on Y_1 anymore. Finally, we obtain $\mathrm{RD}_a(P\|Q) \leq \rho^N$ by induction. □

A.2 Example Linear Secret Sharing Schemes

Here, we give more details on the schemes in Table 1.

Additive Secret Sharing. In the $(n-1)$-out-of-n case, we use simple additive secret sharing, where x is split into random shares $x_1, \ldots, x_n \in \mathbb{Z}_q$ such that $x = \sum_{i=1}^n x_i$. Every party receives exactly one share, hence $L = 1$, $\tau_{\mathsf{max}} = n - 1$ and $\tau_{\mathsf{min}} = n$.

Replicated Secret Sharing [ISN89]. To share x using replicated secret sharing (also called CNF sharing), first sample a set of additive shares $\{s_A\}_A$, over all size-t subsets $A \subset [n]$, such that $\sum_A s_A = x$. Then, party P_i's share consists of every s_A where $i \notin A$. The share size is $L = \binom{n-1}{t}$.

A maximal invalid set contains all the copies of s_A for $A \neq A'$, for some A'. Since $n - t$ parties get A', this gives $\tau_{\mathsf{max}} = nL - (n-t) = (n-t)(\binom{n}{t} - 1)$. On the other hand, a minimal valid set of share elements contains every share s_A, so $\tau_{\mathsf{min}} = \binom{n}{t}$.

Naive Threshold Secret Sharing. In the simplest form of threshold secret sharing, which can be seen as the dual of replicated secret sharing, the dealer distributes a fresh sharing of x to each set S of size $t+1$. There are $\binom{n}{t+1}$ such sets, but only $\binom{n-1}{t}$ of these contain party P_i, so $L = \binom{n-1}{t}$. It's easy to see that $\tau_{\mathsf{max}} = t\binom{n}{t+1}$ and $\tau_{\mathsf{min}} = t + 1$.

Threshold LSS From Monotone Boolean Formulae. An asymptotically more efficient approach is the construction of Benaloh and Leichter [BL90], which builds a linear secret scheme for $\mathbb{A}$ using any monotone Boolean formula for verifying membership of $\mathbb{A}$. A monotone Boolean formula is a circuit with AND/OR gates of fan-in 2 and fan-out 1, where the input wires may have multiple fan-out. The share size of party P_i equals the fan-out of the i-th input wire in the circuit.

Valiant [Val84] described a randomized construction of a monotone Boolean formula for threshold functions with size $O(n^{5.3})$. This leads to an average share size of $O(n^{4.3})$. Hoory et al. [HMP06] gave an improved circuit of size $O(n^{1+\sqrt{2}})$, however, their circuit is not a formula, so cannot be used to build threshold LSS.

Appendix B Missing Definitions of Section 3

We define the properties of compactness and decryption correctness in the following. Note that compactness is only relevant in the fully homomorphic setting.

Definition B.1 (Compactness). *We say that a* TFHE *scheme satisfies* compactness *if there exists a polynomial* poly *such that for all* λ, κ, n, t, C *with* $C\colon \mathcal{M}^k \to \mathcal{M}$ *a circuit of depth at most* κ *and for all* $(m_j)_{j\in[k]} \in \mathcal{M}^k$ *the following holds. For* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$, $\mathsf{ct}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, m_j)$ *for* $j \in [k]$ *and* $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$*, it yields*

$$|\mathsf{ct}| \le \mathsf{poly}(\lambda, \kappa, n),$$

where $|\mathsf{ct}|$ *denotes the bit size of* ct.

Definition B.2 (Decryption Correctness). *We say that a* TFHE *scheme satisfies* decryption correctness *if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all* $\lambda, \kappa, n, t, S, C$ *with* $S \subset [n]$ *of size at least* $t+1$ *and* $C\colon \mathcal{M}^k \to \mathcal{M}$ *of depth at most* κ*, and for all* $(m_j)_{j\in[k]} \in \mathcal{M}^k$ *the following holds. For* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}(1^\lambda, 1^\kappa, n, t)$, $\mathsf{ct}_j \leftarrow \mathsf{Enc}(\mathsf{pk}, m_j)$ *for* $j \in [k]$, $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ *and decryption shares* $d_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$ *for* $i \in S$*, it holds*

$$\Pr[\mathsf{Combine}(\{d_i\}_{i\in S}, \mathsf{ct}) = C(m_1, \ldots, m_k)] = 1 - \mathsf{negl}(\lambda).$$

Appendix C Missing Proofs of Section 4

C.1 Missing Proofs of Section 4.1

Lemma C.1 (Decryption Correctness). *The scheme* TPKE' *of Section 4.1 satisfies decryption correctness, if* TPKE *satisfies decryption correctness and* $\delta = \mathsf{poly}(\lambda)$.

Proof. Fix λ, n, t, S with $S \subset [n]$ of size at least $t+1$ and let $m \in \mathcal{M}'$. Compute $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}'(1^\lambda, n, t)$ and $\mathsf{ct} \leftarrow \mathsf{Enc}'(\mathsf{pk}, m)$. For $i \in S$ we denote by $(d_{ij})_{j\in[\delta]} = \mathbf{d}_i \leftarrow \mathsf{PartDec}'(\mathsf{sk}_i, \mathsf{ct})$ the decryption shares. The inequality $\mathsf{Combine}'(\{\mathbf{d}_i\}_{i\in S}, \mathsf{ct}) \neq m$ holds if for at least one $j \in [\delta]$ the inequality $\mathsf{Combine}(\{d_{ij}\}_{i\in S}, c_j) \neq x_j$ is true. By the union bound we have

$$\begin{aligned}
\Pr\left[\mathsf{Combine}'(\{\mathbf{d}_i\}_{i\in S}, \mathsf{ct}) = m\right] &= 1 - \Pr\left[\mathsf{Combine}'(\{\mathbf{d}_i\}_{i\in S}, \mathsf{ct}) \neq m\right] \\
&= 1 - \Pr\left[\bigcup_{j\in[\delta]} \mathsf{Combine}(\{d_{ij}\}_{i\in S}, c_j) \neq x_j\right] \\
&\leq 1 - \delta \cdot \mathsf{negl}(\lambda) = 1 - \mathsf{negl}(\lambda),
\end{aligned}$$

when $\delta = \mathsf{poly}(\lambda)$. □

C.2 Missing Proofs of Section 4.2

Lemma C.2 (Compactness). *The scheme* TFHE' *of Section 4.2 satisfies compactness if* TFHE *satisfies compactness and* $\delta = \mathsf{poly}(\lambda, \kappa, n)$*.*

Proof. It yields $|\mathsf{ct}| = |c_0| + (n+1)\delta$. From the compactness of TFHE follows that $|c_0| \leq \mathsf{poly}(\lambda, \kappa, n)$ and hence the claim follows. □

Lemma C.3 (Decryption Correctness). *The scheme* TFHE' *of Section 4.2 satisfies decryption correctness if* TFHE *satisfies decryption correctness.*

Proof. Fix $\lambda, \kappa, n, t, S, C'$ with $S \subset [n]$ of size at least $t+1$ and $C' \colon (\mathcal{M}')^k \to \mathcal{M}'$ of depth at most κ. Further, let $(m_j)_{j\in[k]} \in (\mathcal{M}')^k$. Compute $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Setup}'(1^\lambda, 1^\kappa, n, t)$, $\mathsf{ct}_j \leftarrow \mathsf{Enc}'(\mathsf{pk}, m_j)$ for $j \in [k]$ and $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{pk}, C', \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$. Then,

$$\begin{aligned}
&\Pr\left[\mathsf{Combine}'(\{d_i\}_{i\in S}, \mathsf{ct}) = C'(m_1, \ldots, m_k)\right] \\
=&\Pr\left[\mathsf{Combine}(\{d_i\}_{i\in S}, c_0) = C(x_1, \ldots, x_k)\right] \\
=&1 - \mathsf{negl}(\lambda),
\end{aligned}$$

where C is defined as in Eval'. □

Appendix D Missing Proofs of Section 5

Theorem D.1. *The construction in Fig. 5 satisfies decryption correctness.*

Proof. Let $S \subset [n]$ be of size $> t$, and ct be a ciphertext output from Eval on input a set of honestly generated ciphertexts and a circuit C of depth $\leq \kappa$. Let $\mathbf{d}_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$ for $i \in S$, where $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) = \mathsf{Share}(\mathsf{sk})$.

By the strong $\{0,1\}$-reconstruction property of LSS and the validity of S, there exists a minimal valid set of share elements $T \subseteq S \times [L]$ such that

$$\mathsf{Rec}_S((\mathsf{sk}_i)_{i\in S}) = \sum_{(i,j)\in T} \mathsf{sk}_{i,j} = \mathsf{sk}.$$

It follows that

$$\begin{aligned}
\mathsf{Combine}(\{\mathbf{d}_i\}_{i\in S},\mathsf{ct}) &= \left\lfloor (p/q)\cdot(\langle \mathsf{ct},\mathsf{sk}\rangle + \sum_{(i,j)\in T}\mathbf{e}_{i,j}) \right\rceil \\
&= \left\lfloor (p/q)\cdot\left(\lfloor (q/p)m\rceil + e_{\mathsf{ct}} + \sum_{(i,j)\in T}\mathbf{e}_{i,j}\right) \right\rceil \\
&= \left\lfloor (p/q)\cdot\left((q/p)m + e_{\mathsf{rnd}} + e_{\mathsf{ct}} + \sum_{(i,j)\in T}\mathbf{e}_{i,j}\right) \right\rceil \\
&= m + \left\lfloor (p/q)(e_{\mathsf{rnd}} + e_{\mathsf{ct}} + \sum_{(i,j)\in T}\mathbf{e}_{i,j}) \right\rceil,
\end{aligned}$$

where e_{ct} is the ciphertext error and e_{rnd} is a rounding polynomial with coefficients $\leq 1/2$. Letting $e = e_{\mathsf{rnd}} + \dots$ be the sum of the 3 error terms, by the $(\beta_{\mathsf{fhe}}, \varepsilon)$-linear decryption property of FHE, except with probability ε, we have $\|e\|_\infty \leq 1/2 + \beta_{\mathsf{fhe}} + |T| \cdot \beta_{\mathsf{flood}}$. Since $\beta_{\mathsf{fhe}} \leq q/(2p) - \tau_{\mathsf{min}}\beta_{\mathsf{flood}} - 1$ and T is a minimal valid set (so $|T| \leq \tau_{\mathsf{min}}$), we have $\|e\|_\infty < q/(2p)$, so the resulting error term rounds to zero, giving the correct message m. □

Appendix E Details on PRSS-based Construction

E.1 Pseudorandom Secret Sharing

Pseudorandom secret sharing (PRSS) [GI99; CDI05] allows parties to non-interactively obtain secret-sharings of pseudorandom values, after a one-time setup phase which distributes PRF keys among the parties. We use a variant of PRSS over the integers, where the parties do not get shares of uniform values, but instead values bounded from a small range (similarly to [BD10]).

Using a PRF $F : \{0,1\}^\lambda \times \{0,1\}^* \to [-B, B] \cap \mathbb{Z}$, the t-out-of-n threshold case works as follows:

- As setup, for each size-t subset $A \subset [n]$, sample $k_A \leftarrow \{0,1\}^\lambda$. Give k_A to each party P_i, for $i \in [n]$ where $i \notin A$.
- To sample a pseudorandom share on input a nonce v, party P_i computes the shares $s_A = F(k_A, v)$, for each size-t A where $i \notin A$.

The resulting set of shares $\{s_A\}_{|A|=t}$ form a replicated secret sharing of $s = \sum_A s_A$, and we have $|s| \leq B \cdot \binom{n}{t}$. Furthermore, for any collusion of t parties, there is always one share $s_A \in [-B, B]$ that remains unknown.

Converting to Another LSS. A useful property of replicated secret sharing is that replicated shares can be locally converted into any linear secret sharing scheme for the same access structure via a simple linear transformation [CDI05]. We write the procedure of converting a share $\mathbf{s}_i$ into a share $\mathbf{s}'_i$ for a LSSS as: $\mathbf{s}'_i = \mathsf{Convert}_{\mathsf{rep}\to\mathsf{LSS}}(\mathbf{s}_i)$.

E.2 Construction

The construction is shown in Fig. 6. It uses a PRF $F : \{0,1\}^\lambda \times R_q^r \to \mathbb{Z} \cap [-\beta_{\mathsf{flood}}, \beta_{\mathsf{flood}}]$, where we require that the outputs of F are indistinguishable from samples from $\mathcal{D}_{\mathsf{flood}}$.[8]

TFHE.Setup is modified to sample a set of $\binom{n}{t}$ keys k_A and distribute these to the parties in a replicated secret sharing manner. Meanwhile, the secret key of the PKE scheme is shared using standard Shamir sharing. Then, during partial decryption, the parties use the PRF to obtain replicated secret shares of a noise vector. Finally, the parties convert these to Shamir sharings of the same value, exploiting the generality of replicated secret sharing. The Combine algorithm is identical to the previous construction, but using Shamir reconstruction.

$\mathsf{TFHE.Setup}(1^\lambda, n, t)$	$\mathsf{TFHE.PartDec}((\mathsf{sk}_i, \mathbf{k}_i), \mathsf{ct})$
1 : $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.KGen}(1^\lambda)$	1 : // $\mathbf{k}_i = (\mathbf{k}_A)_{i \notin A}$, for all $\lvert A\rvert = t$
2 : $k_A \leftarrow \{0,1\}^\lambda$, for $A \subset [n], \lvert A\rvert = t$	2 : $e_A \leftarrow F(\mathbf{k}_A, \mathsf{ct})$, for $i \notin A$
3 : $\mathbf{k}_i \leftarrow (k_A)_{i \notin A}$	3 : $e_i \leftarrow \mathsf{Convert}_{\mathsf{rep}\to\mathsf{Shamir}}((e_A)_{i \notin A})$
4 : $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{Shamir.Share}(\mathsf{sk})$	4 : // $e_i \in R_q$
5 : **return** $(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_1, \mathbf{k}_1), \ldots, (\mathsf{sk}_n, \mathbf{k}_n))$	5 : **return** $d_i \leftarrow \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e_i$

Fig. 6. Setup and partial decrypt algorithms for the variant of the OW-CPA threshold PKE/FHE scheme using pseudorandom secret sharing.

Correctness. The proof of correctness follows similarly to the proof of Theorem D.1. Since the PRF outputs are bounded by β_{flood}, the noise term sampled with pseudorandom secret-sharing is bounded by $\binom{n}{t} \cdot \beta_{\mathsf{flood}}$. After converting this to Shamir shares, the parties obtain a sharing of the same noise term, so decryption succeeds under the same conditions as in Theorem D.1, with $\tau_{\mathsf{min}} = \binom{n}{t}$.

Security. We show security in the following theorem. Note that we improve the security loss compared with Theorem 5.2, since there is no longer an $nL - \tau_{\mathsf{max}}$ term in the exponent of $\varepsilon_{\mathrm{RD}_a}$.

Theorem E.1. *For any adversary $\mathcal{A}$ against the ℓ-OW-CPA property of the TFHE scheme in Fig. 6, there exists an adversary $\mathcal{B}$ against the OW-CPA property of PKE, such that*

$$\mathsf{Adv}_{\mathsf{TFHE}}^{\ell\text{-}\mathsf{OW}\text{-}\mathsf{CPA}}(\mathcal{A}) \leq \left(\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW}\text{-}\mathsf{CPA}}(\mathcal{B}) \cdot \varepsilon_{\mathrm{RD}_a}^{\ell d}\right)^{(a-1)/a} + \ell\varepsilon$$

[8] We can use any PRF, and use the resulting pseudorandom bits to sample from $\mathcal{D}_{\mathsf{flood}}$.

Proof. The proof follows a similar structure to that of Theorem 5.2, so we only highlight the main differences.

Recall that Game G_0 is the construction. In Game G_1, we changed the way the partial decryptions were computed, for all shares outside of a maximally invalid set of share elements. Since we now only need to simulate partial decryptions of Shamir shares, we instead define a maximally invalid set of *parties*, $T \supset S$, where S is the set of corrupted parties and T has size t. We then simulate the partial decryptions as follows:

1. For $i \in T$, honestly compute $e_A \leftarrow F(\mathbf{k}_A, \mathsf{ct})$, for each size-$t$ set $A \subset [n]$ with $i \notin A$, and let $d_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + \mathsf{Convert}_{\mathsf{rep}\rightarrow\mathsf{Shamir}}((e_{i,A})_A)$
2. Sample $e_T \leftarrow \mathcal{D}_{\mathsf{flood},R_q}$
3. Compute $e = \sum_{A,|A|=t} e_A$
4. For $i \notin T$, let $T' = T \cup \{i\}$ and compute
$$d_i = \lambda_{T',i}^{-1} \cdot \left(\langle \mathsf{ct}, \mathsf{sk} \rangle + e - \sum_{j \in T} \lambda_{T',j} d_j \right)$$
where $\lambda_{T',j}$ are the reconstruction coefficients for Shamir secret sharing, defined by the Lagrange basis for polynomial interpolation at points in T'.

Note that the d_i shares for $i \notin T$ are computed such that the partial decryptions form a valid Shamir sharing of $\langle \mathsf{ct}, \mathsf{sk} \rangle + e$. This is exactly as in the real protocol, except that here the one share e_T that is not part of any shares in the maximally invalid set T is sampled from $\mathcal{D}_{\mathsf{flood}}$ (step 2) instead of with the PRF. Since the PRF key k_T is not given to the adversary, this hybrid is indistinguishable from the real game G_0, by the security of the PRF.

Game G_2 then makes the same change as in Theorem 5.2, removing the possibility of decryption failure. This is indistinguishable from the previous game, except with probability $\ell\varepsilon$.

In Game G_3, in the noise term e, the share e_T sampled in step 2 is sampled with simulated noise using $\mathcal{D}_{\mathsf{sim},R_q}$. At the same time, we remove the ciphertext noise term in $\langle \mathsf{ct}, \mathsf{sk} \rangle$, so instead of the last step above, we will now compute

$$d_i = \lambda_{T',i}^{-1} \cdot \left(\lfloor (q/p) \cdot m \rceil + e - \sum_{j \in T} \lambda_{T',j} d_j \right)$$

Notice that the difference between games G_2 and G_3 is that G_2 uses the real ciphertext noise and $e_T \leftarrow \mathcal{D}_{\mathsf{flood},R_q}$ to simulate the missing partial decryptions, while G_3 instead uses zero ciphertext noise and $e_T \leftarrow \mathcal{D}_{\mathsf{sim},R_q}$. Let $e_{\mathsf{ct}} = \langle \mathsf{ct}, \mathsf{sk} \rangle - \lfloor (q/p) \cdot m \rceil$ be the ciphertext noise. Using Lemma 2.7, we have

$$\mathrm{RD}_a(\mathcal{D}_{\mathsf{flood},R_q} + e_{\mathsf{ct}} \| \mathcal{D}_{\mathsf{sim},R_q}) \leq \varepsilon_{\mathrm{RD}_a}^{d}$$

Similarly to the proof of Theorem 5.2, for ℓ decryption queries we obtain

$$\mathsf{Adv}_{\mathsf{TFHE}}^{G_2}(\mathcal{A}) \leq \left(\mathsf{Adv}_{\mathsf{PKE}}^{G_3}(\mathcal{A}) \cdot \varepsilon_{\mathrm{RD}_a}^{\ell d} \right)^{(a-1)/a}$$

and the result follows. □

Achieving Strong Robustness. An advantage of this construction is that if $t < n/3$, we can exploit the error-correction properties of Shamir sharing to guarantee that Combine outputs the correct message, even in the presence of t maliciously chosen partial decryptions. This is because a properly generated PartDec output is a valid Shamir share, so the parties can always use Reed-Solomon error correction to reconstruct the secret and decrypt, given at least $n/3$ valid shares. This allows the construction to satisfy the strong chosen-plaintext robustness property (Def. 3.5). While this is also possible to achieve using the OW-CPA to IND-CPA transformation from Section 4 (and even with $t < n/2$), by using Shamir we avoid the $\binom{n}{t}$ cost of finding the correct subset of partial decryptions, significantly improving the efficiency of the Combine algorithm. Furthermore, the Shamir approach is compatible with FHE and not just PKE.

Appendix F More Details on Parameters of Section 6

We recall the high level design of Kyber with messages of the form $m \in R_2 \cong \{0,1\}^d$, where d denotes the degree of the ring R. The scheme uses the centered binomial distribution with parameter $\eta \in \mathbb{N}$, denoted by CBD_η. We say that a ring element is sampled from CBD_η if all its d coefficients are independently sampled from CBD_η. This generalizes to vectors in R^r, where r is the underlying module rank. Let $\mathsf{Kyber} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ be as follows:

$\mathsf{Setup}(1^\lambda)$**:** Sample short vectors $\mathbf{s}, \mathbf{e} \in R_q^r$ from CBD_η and a uniform matrix $\mathbf{A} \in R_q^{r\times r}$. Set $\mathsf{sk} = (\mathbf{s}, \mathbf{e})$ and $\mathsf{pk} = (\mathbf{A}, \mathbf{t})$, where $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}$.

$\mathsf{Enc}(\mathsf{pk}, m)$**:** Sample a short vector $\mathbf{r} \in R_q^r$ from CBD_η and $\mathbf{e}_1 \in R_q^r$ and $e_2 \in R_q$ from CBD_η. Set $\mathbf{u} = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1$ and $v = \mathbf{r}^T\mathbf{t} + e_2 + \lfloor q/2\rceil \cdot m$. Output $\mathsf{ct} = (\mathbf{u}, v)$.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$**:** Compute $c' = v - \mathbf{u}^T\mathbf{s} = \mathbf{r}^T\mathbf{e} - \mathbf{e}_1^T\mathbf{s} + e_2 + \lfloor q/2\rceil \cdot m$. Output $\lfloor c' \cdot 2/q\rceil$.

For simplicity, we omit the additional rounding usually applied to ciphertexts to further reduce their size.

Table 4. Parameter sets for Kyber.

Set	d	r	q	η
Kyber768	256	3	3329	2
Kyber1024	256	4	3329	2
Kyber1280	256	5	3329	2
Kyber1536	256	6	3329	2
Kyber1792	256	7	3329	2

Generic Parameters for Large Numbers of Parties. We first describe a simplified way of deriving parameters, where we assume that $\mathcal{D}_{\mathsf{sim}}$ and $\mathcal{D}_{\mathsf{flood}}$ both are uncut rounded Gaussian distributions of the same width σ.

Using Lemma 2.6 with our choice of distributions, Equation 2 simplifies to

$$\lambda_{\mathsf{TPKE}} \geq \frac{a-1}{a} \cdot \left(\lambda_{\mathsf{PKE}} - \ell d(nL - \tau_{\mathsf{max}}) \frac{a\beta_{\mathsf{pke}}^2}{2\sigma^2} \log_2 e \right). \tag{3}$$

When setting $\sigma = \beta_{\mathsf{pke}} \sqrt{\ell d(nL - \tau_{\mathsf{max}})(a-1) \log_2 e}$, the above simplifies to

$$\lambda_{\mathsf{TPKE}} \geq \frac{a-1}{a} \cdot \lambda_{\mathsf{PKE}} - 1, \tag{4}$$

which promises a rather small security loss at the expense of a larger modulus. Note that we have to set $q > 4(\beta_{\mathsf{pke}} + \tau_{\mathsf{min}}\beta_{\mathsf{flood}})$ in order to guarantee correctness (Thm. D.1). Let's for concreteness set $\beta_{\mathsf{flood}} = 10\sigma$ and $a = 100$. Recall that Kyber is a PKE with $(\beta_{\mathsf{pke}}, \varepsilon)$-linear decryption, where β_{pke} depends on the maximal failure probability ε we tolerate. If we take as a concrete example Kyber1024, it offers $(390, 2^{-60})$ as well as $(934, 2^{-300})$-linear decryption.

When considering full threshold, we use additive secret sharing and when considering non-full threshold, we assume naive secret sharing, defining the parameters $L, \tau_{\mathsf{max}}, \tau_{\mathsf{min}}$ as in Table 1. After having set σ and q, one can use the Lattice Estimator [APS15] to derive λ_{PKE}. For simplicity we set λ_{PKE} as the core-SVP classical hardness, i.e., the resulting BKZ block size multiplied by 0.292. The resulting λ_{TPKE} and Δ_λ then come from Equation 4. We give some sample parameters for TKyber1024 in Table 2. Note that we mean by TKyber1024 that we take all the original Kyber1024 parameters, but modify the modulus q.

Hand-Tuned Parameters for Small Number of Parties. We now describe how we can obtain tighter concrete parameters (in particular a small modulus q) by allowing for different flooding and simulating Gaussian distributions and optimizing their concrete width. Throughout this section, we set $\mathcal{D}_{\mathsf{sim}}$ (resp. $\mathcal{D}_{\mathsf{flood}}$) as the rounded Gaussian distribution of width σ_{sim} (resp. σ_{flood}), where we additionally apply a tail cut after $2 \cdot \sigma_{\mathsf{sim}}$ (resp. $2 \cdot \sigma_{\mathsf{flood}}$).

By extending the Python program for computing security estimates of Kyber[9], we design a Python program that proceeds in the following three steps:

Step 1: Finding $\mathcal{D}_{\mathsf{flood}}$. The high level idea is to find the largest σ_{flood} we can use in our TPKE such that we still guarantee correctness (Theorem D.1). This is how we optimally make use of our modulus q. For simplicity, we set $p = 2$ and hence correctness is fulfilled as long as the infinity norm of the final noise is at most $q/4$. This procedure depends on the Kyber parameters (that define the noise from the decryption algorithm) as well as the maximal decryption failure probability we want to aim for. We fix this probability to be 2^{-60}. At the end, the procedure outputs σ_{flood} and the bound B.

[9] `https://github.com/pq-crystals/security-estimates`

Step 2: Finding $\mathcal{D}_{\mathsf{sim}}$. Once we have computed $\mathcal{D}_{\mathsf{flood}}$, we can find $\mathcal{D}_{\mathsf{sim}}$ such that the Rényi divergence $\mathrm{RD}_2(\mathcal{D}_{\mathsf{flood}}+B\|\mathcal{D}_{\mathsf{sim}})$ is smallest. We start by setting $\mathcal{D}_{\mathsf{sim}} = \mathcal{D}_{\mathsf{flood}}$ and compute the Rényi divergence of order 2. We now (slightly) increase $\mathcal{D}_{\mathsf{sim}}$ step by step and expect the Rényi divergence to decrease up to some optimal sweet spot. Once we observe that the Rényi divergence increases again, we stop increasing $\mathcal{D}_{\mathsf{sim}}$ and take this as the optimal choice. Note that for fixed $\mathcal{D}_{\mathsf{flood}}$, B and $\mathcal{D}_{\mathsf{sim}}$, it yields $\mathrm{RD}_2(\mathcal{D}_{\mathsf{flood}}+B\|\mathcal{D}_{\mathsf{sim}}) \leq \mathrm{RD}_a(\mathcal{D}_{\mathsf{flood}}+B\|\mathcal{D}_{\mathsf{sim}})$ for all $a > 1$. Hence, it is reasonable to compute the sweet spot for the order 2.

Step 3: Finding $\varepsilon_{\mathrm{RD}_a}$. As we now have $\mathcal{D}_{\mathsf{flood}}$, $\mathcal{D}_{\mathsf{sim}}$ and B, we can find the optimal order of the Rényi divergence. Note that, even though $\varepsilon_{\mathrm{RD}_a}$ doesn't decrease for increasing a, the factor $(a-1)/a$ in Equation 2 suggests that the optimal a might not necessarily be $a = 2$. For concreteness, we search the minimum among the orders $a \in [2, \ldots, 11]$. We then output the optimal choice of a together with the resulting Rényi divergence $\varepsilon_{\mathrm{RD}_a}$. Finally, we have everything together to compute the upper bound on λ_{TPKE}.

Table 3 summarizes our findings. We use as base security λ_{PKE} the core-SVP classical hardness of the underlying LWE instance, which can be easily computed using any LWE estimator. For convenience, we used the leaky LWE estimator [Dac+20]. We give some estimates for the final security λ_{TPKE} for different choices of small numbers of parties n, threshold t and number of queries ℓ. For all computations, we apply a (rather aggressive) Gaussian tail cut after 2 times the Gaussian width and assume a failure probability bound of 2^{-60}.

Here, we consider variants of the Kyber1024 parameter set, where we multiply the modulus q by some scaling factor. This scaling factor is intended to give an idea of the order of magnitude of the modulus we need. We remark that multiples of 3329 might not necessarily be the optimal choice when taking implementation characteristics into account.

Comparing The Rényi Divergences. We would like to highlight that the two strategies assume different flooding and simulating noise distributions $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$. Whereas in the first we assume the *same* and (quasi) *uncut* rounded Gaussian distributions, we computed the parameters in the second case with a *different* and *tail cut* rounded Gaussian distributions. When fixing a maximal decryption failure probability, one can choose the modulus q much smaller in the latter case. However, the sharper we cut off the rounded Gaussian distribution, the more the Rényi divergences from Lemma 2.6 and one computed by our Python program diverge from each other.