Improved Rényi Arguments for Lattice-Based Threshold Encryption

Katharina Boudgoust[1] and Anamaria Costache[2]
katharina.boudgoust@lirmm.fr, anamaria.costache@ntnu.no

[1] CNRS, Univ Montpellier, LIRMM, France
[2] NTNU, Norway

Abstract. Threshold encryption schemes provide a common tool to secure a public-key encryption scheme against single point of failure attacks. Despite the success of lattices in building fully-homomorphic and presumably quantum-resistant encryption schemes, the task of thresholdizing those schemes remains challenging. The major bottleneck in the standard approach is the use of statistical noise flooding, leading to a significant efficiency loss and the need of stronger hardness assumptions. Recent works have replaced the heavy statistical noise flooding by a lighter one using the Rényi divergence. The new Rényi noise flooding both improves the efficiency and allows to use weaker hardness assumptions. However, arguing semantic security of lattice-based threshold schemes in the presence of Rényi noise flooding showed to be challenging. Chowdhury et al. (IACR ePrint'22) argued in the fully-homomorphic case that the Rényi divergence directly applies for semantic security by making use of an existing framework called *public sampleability*.
In this work, we argue that their public sampleability framework was neither sufficient nor correctly used. To address both issues, we strengthen the framework and thoroughly apply it to prove semantic security of generic lattice-based threshold encryption constructions. We distinguish between the plain public-key and the fully-homomorphic settings, as different security notions are achieved. As a byproduct, this shows that the proof detour via one-way security made by Boudgoust and Scholl (Asiacrypt'23) was superfluous, now leading to tighter proofs in the standard model.

Keywords: Lattices · Threshold Encryption · Rényi Divergence

1 Introduction

Diffie and Hellman [DH76] and Merkle [Mer78] initiated the public-key cryptography revolution, by dividing the key material into a publicly available key pk and a hidden secret key sk. It allowed two parties to safely start communicating without having physically met beforehand. Threshold cryptography [DF90] can be seen as a continuation of the revolution, where now the secret key sk is split into multiple secret key shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$. These shares are then distributed

to different parties, and a large enough subset of those parties is required in order to perform the secret-key related operation, thus avoiding a single point of failure. This divides trust among multiple parties and makes the cryptosystem more robust.

In the context of public-key encryption schemes, the secret key is needed for decryption. We call a public-key encryption scheme t-out-of-n threshold, if any set of t or more parties can jointly decrypt ciphertexts, whereas any set of fewer than t parties learn nothing about the underlying message. In this work, we focus on threshold encryption schemes whose security relies on the hardness of lattice problems. Lattices play a leading role in designing presumably post-quantum cryptographic schemes, as highlighted by the standardization of Kyber [SAB+22] as the first encryption scheme in the post-quantum project run by NIST[3]. Moreover, despite significant research efforts, lattices remain the only mathematical setting for which we know how to instantiate fully-homomorphic encryption schemes. Such schemes allow for arbitrary computations over encrypted data and, as such, represent a very powerful tool, both in theory and in practice. Unfortunately, it seems particularly challenging to thresholdize lattice-based encryption schemes such as Kyber. Unsurprisingly, thresholdizing schemes with additional fully-homomorphic properties is even more challenging. Bendlin and Damgård [BD10] were the first to propose a threshold key generation and decryption for Regev's public-key encryption scheme [Reg05]. It was later extended to the fully-homomorphic setting [AJL+12; BGG+18].

Noise Flooding. Most lattice-based encryption schemes (relying on the hardness of the LWE problem [Reg05]) perform decryption in two steps: first, a linear function is applied to the secret key and the ciphertext, which yields the (potentially encoded) message m plus some ciphertext noise e^{ct}. In a second step, this noise is removed to recover the message m. We say that the decryption is *nearly linear*. The important observation is that e^{ct} depends both on the underlying plaintext *and* the secret key. Thus, a naive thresholdizing approach, where a linear secret sharing is combined with the nearly linear decryption, fails to be secure. The goal of all existing lattice-based threshold solutions is thus to make sure that e^{ct} is sufficiently "hidden" to the parties corrupted by the adversary. The common approach lets every party locally add some fresh noise on their partial decryption share before sending it to the other parties. After decryption, the parties only learn $e^{\mathrm{ct}} + e^{\mathrm{fl}}$, where $e^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{flood}}$ is a flooding noise term hidden from the adversary. When proving security, the real partial decryption shares are replaced by simulated ones, which only contain a noise term sampled from $\mathcal{D}_{\mathsf{flood}}$ and hence do not depend on the secret key. The last (but most important) step is to properly argue that the adversary cannot distinguish between real and simulated partial decryption shares. That is, that the flooding noise is big enough to "hide" the ciphertext noise (i.e., the dependency on the secret key sk). One widespread approach is to argue that the statistical distance between the two distributions is negligibly small [BD10; BGG+18]. This requires the ratio be-

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

tween flooding noise e^{fl} and ciphertext noise e^{ct} to be superpolynomial in the security parameter. For correctness, computations need to be performed over a modulus which is larger than e^{fl}. At the same time, e^{ct} depends on the associated LWE noise. As a consequence, statistical noise flooding negatively impacts both efficiency (requiring a superpolynomial modulus) and security (relying on the LWE problem with superpolynomial modulus-to-noise ratio).

Rényi Noise Flooding. Due to those significant drawbacks, recent works have replaced the heavy statistical noise flooding by a lighter Rényi noise flooding in the threshold fully-homomorphic setting [BS23b; CSS+22]. This allows to rely on the hardness of LWE with a *polynomially large* modulus, solving both the aforementioned security and efficiency issues. The Rényi divergence has been shown useful in reducing the amount of noise flooding needed in lattice-based security proofs before [Pre17; BLR+18]. In particular, Rényi arguments directly apply to search problems, like unforgeability of signatures [ASY22] or one-way security of encryption schemes [DSS+21]. However, as discussed in [BLR+18], one cannot in general directly apply Rényi divergence arguments to prove semantic security. Informally, in the t-out-of-n threshold encryption context, semantic security states that the adversary cannot distinguish between the encryptions of two messages, while having corrupted up to $t-1$ parties and access to a partial decryption oracle. One approach, taken by [BS23b], is to make a "proof detour" by first proving one-way security (where Rényi arguments apply directly) and then lifting the threshold scheme to semantic security. Another approach is to demonstrate that the security game fulfills a property called *public sampleability* [BLR+18], which we introduce in the technical overview. This approach was first described in [CSS+22] for the specific TFHE scheme [CGG+20]. Both approaches require careful analysis. In particular, the published version [BS23b] contains an error in the one-way to semantic security transformation, which allowed for an attack as demonstrated in [PS24], and whose fix required weakening the targeted security notion, as detailed in the updated IACR ePrint version [BS23a]. The new proof now achieves the same security notion as claimed in [CSS+22]. However, the updated reduction of [BS23a] requires the underlying fully-homomorphic encryption scheme to fulfill *circuit privacy*. Informally, circuit privacy asks that an adversary cannot distinguish whether a ciphertext comes from homomorphically evaluating a circuit, or from directly encrypting the output of the circuit. Surprisingly, [CSS+22] claims semantic security *without* the need of circuit privacy. Even though possible to achieve, circuit privacy is a rather strong property to require and comes with an efficiency cost. This opens the question of whether the circuit privacy required in [BS23a] is only a proof artifact, or whether the proof of [CSS+22] is incomplete.

Contributions. This work sheds light on the discrepancy between [BS23a] and [CSS+22]. First, we argue that the public sampleability framework defined in [BLR+18] and used unmodified in [CSS+22] fails to correctly capture the security game of threshold encryption schemes, both in the public-key and

fully-homomorphic setting. Moreover, we detail a flaw in the security argument in [CSS+22], which wrongly allowed them to bypass the circuit privacy property. We then address both issues. First, we strengthen the public sampleability framework to correctly capture security games where the adversary has access to oracles. We believe this to be of independent interest. Then, we thoroughly apply the extended framework to directly prove semantic security of the generic lattice-based threshold encryption construction from [BS23a]. Otherwise put, we show that the construction proven to be one-way secure in [BS23a] already provides a semantically secure scheme. We prove the security of the public-key construction separately from the one of the fully-homomorphic construction, as the achieved security notions are not the same. In the public-key encryption setting, we strengthen the security notion used in [BS23a] by giving out to the adversary the encryption randomness corresponding to the partial decryption queries. In the fully-homomorphic case, we achieve the same selective semantic security notion as in the corrected version of [BS23a]. Moreover, our thorough proof requires circuit privacy, dissolving the aforementioned discrepancy. In other words, directly applying public sampleability leads to the same requirements and results as doing the "proof detour" from [BS23a] without any additional costs. Doing so, we avoid their rather contrived notion of one-way security in the fully-homomorphic setting, requiring a potentially delicate min-entropy check. Moreover, our proofs are tighter when limiting ourselves to the standard model. Inherited from [BS23a], our results apply to a large class of lattice-based threshold encryption schemes, both in the public-key and the fully-homomorphic setting. In particular, we cover schemes working over the integers (the unstructured setting relying on LWE) and over rings of integers (the structured setting relying on R-LWE or M-LWE). Moreover, we build t-out-of-n threshold schemes for any threshold $t \leq n$. Throughout the work, we focus on exact fully-homomorphic encryption schemes, such as BGV [BGV14], BFV [FV12; Bra12] and TFHE [CGG+20]. We assume a passive adversary who statically corrupts parties at the beginning of the game, as well as a trusted setup for key generation.

Technical Overview. We now give a high-level explanation on why the original public sampleability framework [BLR+18, Sec. 4.1] is not powerful enough to capture the semantic security game of threshold public-key encryption, and how we address this issue. Our argument generally extends to any indistinguishability game where the adversary has access to some oracle. As the following argument is independent of fully-homomorphic properties, we focus on the plain public-key setting.

For two distributions X_0, X_1, we consider the decision problem P which asks an adversary, given as input smpl, to decide whether $\mathsf{smpl} \leftarrow X_0$ or $\mathsf{smpl} \leftarrow X_1$. We assume that the distributions are of the form

$$X_b := \{\, \mathsf{smpl} := x \mid \mathbf{r} \leftarrow \Phi, x \leftarrow D_b(\mathbf{r}) \,\},$$

where Φ and $D_b(\mathbf{r})$ are distributions, the latter parameterized by $b \in \{0,1\}$ and $\mathbf{r} \in \mathrm{Supp}(\Phi) \subset S$. Public sampleability requires the existence of an efficient public sampling algorithm Sample, which for all pairs $(\mathbf{r}, b) \in S \times \{0,1\}$, on input $x \leftarrow D_b(\mathbf{r})$ as well as a bit b', outputs a fresh sample distributed as $D_{b'}(\mathbf{r})$ for the same $\mathbf{r}$ (which in general remains hidden to the sampler). Then, the hardness of deciding P can be upper bounded by the hardness of deciding a related problem P', where the only change is that Φ is replaced by a different distribution Φ' with $\mathrm{Supp}(\Phi) \subseteq \mathrm{Supp}(\Phi') \subset S$. The loss in advantage mainly depends on the Rényi divergence between Φ and Φ' [BLR+18, Thm. 4.2]. The original framework is applied to argue semantic security of the dual Regev encryption scheme [BLR+18, Sec. 4.2]. Capturing its semantic security game through a decision problem P of the above form is relatively straightforward: for m_0, m_1 the two messages chosen by the adversary, one attributes $D_0(\mathbf{r})$ to the encryption of m_0 and $D_1(\mathbf{r})$ to the encryption of m_1, where $\mathbf{r}$ is the public key of the scheme.

When trying to apply the public sampleability framework to the threshold encryption context, one encounters the following issue, ignored in [CSS+22]. The semantic security game asks the adversary to distinguish the encryptions of two messages, while *additionally* giving them the secret key shares of the corrupted parties and access to a partial decryption oracle. That is, the resulting distribution $D_b(\mathbf{r})$ needs to allow for some input $\boldsymbol{\eta}$ capturing the choices made by the adversary. Note that there is some adaptiveness inherent to the security game: the adversary first sees the public key, then chooses which parties they corrupt. Only after having received the corresponding secret key shares, they (potentially adaptively) query the partial decryption oracle. And only then, they output the two challenge messages, whose encryptions they have to distinguish. In Section 3, we address the described issue by extending the public sampleability framework to allow for *adaptive adversary input*. Our extended framework considers distributions of the form

$$X_b := \{\, \mathsf{smpl} := (\mathsf{Part}(\mathbf{r}), x) \mid \mathbf{r} \leftarrow \Phi, x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r}) \,\},$$

where Φ and $D_b(\boldsymbol{\eta}, \mathbf{r})$ are distributions. The latter is now parameterized by $b \in \{0,1\}$, $\mathbf{r} \in \mathrm{Supp}(\Phi)$ as well as a parameter $\boldsymbol{\eta}$ chosen by the adversary. Moreover, the output sample also contains $\mathsf{Part}(\mathbf{r})$, some partial information on $\mathbf{r}$. For instance, $\mathbf{r}$ could contain both sk and pk, but $\mathsf{Part}(\mathbf{r})$ only contains pk. The important property is that we allow $\boldsymbol{\eta}$ and $\mathsf{Part}(\mathbf{r})$ to depend on one another. In the threshold decryption context, $\mathsf{Part}(\mathbf{r})$ contains the information the adversary obtains up to the moment they give out the two challenge messages, namely the public key, the corrupted key shares and the output of the partial decryption queries. The parameter $\boldsymbol{\eta}$ contains the choices made by the adversary, like the set of corrupted parties, the input to the partial decryption queries and the two challenge messages. The sample x directly corresponds to the challenge ciphertext. Demonstrating public sampleability boils down to showing that for fixed parameter $\mathbf{r}$ (and hence $\mathsf{Part}(\mathbf{r})$), one can use publicly available information to correctly re-encrypt the challenge message for the corresponding bit. Importantly, Theorem 1 proves that the original public sampleability result carries over to the adaptive adversary input context. This extended framework of

public sampleability might be useful beyond the threshold context, for instance to prove semantic security of public-key encryption schemes with different additional properties.

With this stronger formalism at hand, it is now possible to formulate the semantic security game of our threshold public-key encryption scheme through a decision problem P of the above form. The crucial observation is that the partial decryption queries do not depend on the challenge bit of the game. This is because the partial decryption oracle *freshly* encrypts the queried message before partially decrypting it. To highlight the arguments from the improved public sampleability and for ease of exposition, we prove in Section 4 the easier n-out-of-n threshold case, and provide the proof of the general t-out-of-n case to Appendix B.1.

When moving in Section 5 to the fully-homomorphic setting, capturing the semantic security game through the public sampleability framework becomes more delicate: the partial decryption queries may now *depend* on the challenge bit. More concretely, the adversary is allowed to ask for partial decryptions of circuit evaluations that depend on the challenge bit as long as the circuit's output is the same for $b = 0$ and $b = 1$. A priori, even though the outputs are conditioned to be the same, the evaluations, and hence the partial decryption shares might depend on b. Thus, for given partial decryption outputs, one cannot simply re-encrypt the challenge messages (as possible in the plain public-key setting). This problem has been missed by [CSS+22], resulting in an incorrect use of the public sampleability framework.[4] One way to avoid the aforementioned problem is to assume that one can replace the partial decryptions of the homomorphically evaluated ciphertexts (which might depend on the challenge bit) by partial decryptions of ciphertexts directly encrypting the output (conditioned to be independent of the challenge bit). This condition is fulfilled under the notion of circuit privacy. Note that [BS23a] uses the property of circuit privacy to make essentially the same argument. As for the public-key encryption case, for ease of exposition, we concentrate on the full threshold case in Theorem 3. It can be generalized to arbitrary thresholds in a similar way as shown in the plain public-key encryption case.

From a conceptual point of view, it is of course desirable to use our direct proof of semantic security, instead of doing the "proof detour" via one-way security, as done in [BS23a]. This is especially true as their one-way to semantic security transformation either requires the use of the random oracle model, or a rather delicate proof using unpredictable entropy. However, it remains to argue that the direct proof does lead not to a larger loss in advantage. For simplicity, we set the Rényi order to $\alpha = \lambda$ (with λ being the security parameter) and observe that the loss in Theorem 2 is dominated by the exponent $(2\alpha - 1)/(\alpha - 1) \approx 2$, thus quadratic. In comparison, the loss for the one-way secure scheme in [BS23a, Sec. 5] is dominated by $\alpha/(\alpha - 1) \approx 1$, hence linear. To the latter, one needs to add the loss coming from the one-way to semantic security transformation. Restricting ourselves to the standard model, the transformation uses Goldreich-

[4] A recent publication [PS24] makes a similar observation, without providing a fix.

Levin hardcore bits [BS23a, Sec. 4.2] and has a cubic loss. Concretely, [BS23a, Sec. 4.2] suggest that more than 500 bits of security for the one-way scheme are needed to obtain 128 bits of security for the resulting semantically secure in the standard model. Taking the tighter transformation in the random oracle model [BS23a, Sec. 4.1], which is not applicable to the fully-homomorphic setting, it scales multiplicatively with the number of random oracle queries. Overall, our reduction loss is comparable to [BS23a] in the random oracle model, and asymptotically tighter in the standard model.

Comparison to Related Work. To put our results into context with existing works, we recall the various flavors for defining security in the threshold encryption context. First, we distinguish between active and passive adversaries. Whereas the passive adversary can only exploit information coming from honest executions of the protocol, the active adversary can deviate from it. Moreover, the adversary either statically or adaptively corrupts the parties in the threshold scheme. When providing the adversary access to oracles, such as oracles for encryption, evaluation or partial decryption, we distinguish between selective or adaptive access. In the former, the adversary has to send out all the queries at once, wheres in the adaptive setting, they can adapt their next queries according to the outputs of the previous queries. It has been shown that adaptive security is strictly stronger than selective security in the threshold context [PS24]. Finally, there are simulation-based and game-based security notions. The former provides a strictly stronger security than the latter [MS23, App. A.3]. In the game-based setting, one can either allow for any polynomial number of oracle queries, or put a concrete bound on it.

The recent work by Micciancio and Suhl [MS23] studies threshold public-key encryption schemes. Their achieved security notion is somewhat orthogonal to ours: whereas we provide a game-based notion (with an a priori bound on the number of queries, inherent to efficient Rényi arguments), they achieve stronger simulation-based security. However, their full security proof on standard hardness assumptions is only proven for the unstructured setting over $\mathbb{Z}$ and the full threshold case (i.e., t equals n). They can allow for a significantly lighter noise flooding by making use of Gaussian convolution properties. However, it limits their scheme to the use of Gaussian distributions, which are prone to side-channel attacks and difficult to mask [HPR+20]. Their techniques do not a priori extend to the FHE case, as even linear homomorphic properties imply that decryption would not be deterministic anymore. Additionally, the ciphertext noise might not follow a Gaussian distribution. Indeed, while it has been shown that the noise in an FHE ciphertext is *well-approximated* by a Gaussian [CCH+24; MP24], it is not distributed according to a Gaussian. In contrast, our construction allows a much more flexible choice of distributions (cf. Remark 2). For instance, recent results propose sums of the bounded uniform distribution as a masking friendly alternative, while still allowing for good Rényi divergence bounds [PKP+24]. Surprisingly, the use of Gaussians also weakens the security notion as the argument in [MS23] requires that the encryption randomness of the ciphertexts used

in the partial decryption queries stays hidden from the adversary, whereas we can provide the randomness to the adversary (cf. Remark 1).

Another recent work by Passelègue and Stehlé [PS24] studies threshold fully-homomorphic encryption schemes. By introducing an additional party, which does not collude with the adversary, they can reduce the communication to and from the different parties performing partial decryption. Similarly to [MS23], they provide simulation-based security and use properties of Gaussian distributions. However, their work is subject to the same limitations, namely they are restricted to the unstructured setting over $\mathbb{Z}$, only achieve full-threshold and require the use of Gaussian distributions. As they are still performing statistical noise flooding, their security needs to assume hardness of the non-standard LWE with superpolynomial noise-to-modulus ratio. As long as the circuit privacy mechanism used in our construction allows for a polynomially large modulus (as for instance achieved in [DS16; BPM+16]), our security can rely on the standard LWE assumption. A recent follow-up work [OT25] generalizes [PS24] to the structured setting over cyclotomic rings and to arbitrary t-out-of-n thresholds, by introducing an additional round of communication and a more involved key generation.

To the best of our knowledge, only two existing works consider active adversaries: one in the public-key setting [DLN+21] and one in the fully-threshold setting [DDK+23]. Both follow the statistical noise flooding approach (even though the latter obtains some concrete improvements by making use of the efficient bootstrapping procedure of the TFHE scheme [CGG+20]). Finally, [DLN+21] are the only work providing security against adaptive corruptions.

2 Preliminaries

The abbreviation PPT stands for probabilistic polynomial-time. A function negl is said to be negligible in λ, if $\mathsf{negl}(\lambda) = \lambda^{-\omega(1)}$. Throughout the paper, we work over cyclotomic rings, i.e., rings of the form $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for $f(X)$ a cyclotomic polynomial. The degree of $\mathcal{R}$ corresponds to the degree of $f(X)$. By considering the coefficient vectors of elements in $\mathcal{R}$, we can associate norms to the ring elements. For $k \in \mathbb{N}$, we denote by $\mathbb{Z}_k$ the integers modulo k, and by $[k]$ the set $\{1, \ldots, k\}$. The first naturally extends to the ring of integers modulo k, denoted by $\mathcal{R}_k$, where we apply the modular reduction coefficient-wise. For any distribution P, we denote by $\mathrm{Supp}(P)$ its support.

2.1 Linear Secret Sharing

We recall the definition of a linear secret sharing scheme for threshold access structures. For $n \in \mathbb{N}$, any set $S \subseteq [n]$ and vector $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$, we denote by $\mathbf{v}_{|S}$ the vector of shares restricted to $\mathbf{v}_i$ for indices $i \in S$.

Definition 1 (Linear Secret Sharing). *Let q, L, t, n be positive integers. A t-out-of-n linear secret sharing scheme LSS is defined by a randomized algorithm*

$\mathsf{Share}\colon \mathbb{Z}_q \to (\mathbb{Z}_q^L)^n$ *and a family of deterministic algorithms* $\mathsf{Rec}_S\colon (\mathbb{Z}_q^L)^{|S|} \to \mathbb{Z}_q$*, for* $S \subseteq [n]$*, which satisfy:*

Privacy: *For any set* $S \subseteq [n]$ *such that* $|S| < t$*, 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* $S \subseteq [n]$ *such that* $|S| \geq t$*, any* $x \in \mathbb{Z}_q$ *and* $\mathbf{v} \leftarrow \mathsf{Share}(x)$*, the associated reconstruction algorithm outputs* $\mathsf{Rec}_S(\mathbf{v}_{|S}) = x$*.*

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

From $\mathbb{Z}_q$ *to* $\mathcal{R}_q$*.* In this work, we need to share $\mathbf{x} \in \mathcal{R}_q^r$ for $\mathcal{R}$ of the form $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for some degree-d polynomial f, positive integer q and rank r. We do so by sharing each of the d coefficients of the r polynomials in $\mathbf{x}$ separately. Now, a party's share lies in $(\mathcal{R}_q^r)^L$ and operations are over $\mathcal{R}_q$.

Additive Secret Sharing. In the proofs of the main body, we use the full threshold setting (that is, $t = n$), where additive secret sharing can be used. For a given secret $\mathsf{sk} \in \mathcal{R}_q^r$, the Share algorithm chooses independent uniform elements $\mathsf{sk}_1, \ldots, \mathsf{sk}_{n-1} \leftarrow \mathcal{R}_q^r$, then sets $\mathsf{sk}_n := \mathsf{sk} - \sum_{j=1}^{n-1} \mathsf{sk}_i$. Thus, $L = 1$. Reconstruction Rec only succeeds for $S = [n]$ and simply adds all the shares together.

General Case. When moving to the general t-out-of-n threshold case, we make use of linear secret sharing with strong binary reconstruction, whose definition we recall in Appendix A.1.

2.2 Rényi Divergence

In the following, we recall the definition and useful properties of the Rényi divergence of some order α. Even though it is technically possible to allow the order to take a real value, we limit ourselves to α being a positive integer throughout the work.

Definition 2 (Rényi Divergence). *For any two discrete probability distributions* P *and* Q *with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$*, the Rényi divergence of order* $\alpha \in (1, \infty)$ *is defined as*

$$\mathrm{RD}_\alpha(P\|Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}}.$$

For $\alpha = \inf$*, we define* $\mathrm{RD}_\infty(P\|Q) = \max_{x \in \mathrm{Supp}(P)} \frac{P(x)}{Q(x)}$*.*

Lemma 1 (Probability Preservation [LSS14, Lem. 4.1]). *Let* P, Q *denote two probability distributions with* $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$*. Let* $E \subseteq \mathrm{Supp}(Q)$*. For* $\alpha \in (1, \infty)$*,*

$$Q(E) \geq P(E)^{\frac{\alpha}{\alpha-1}} / \mathrm{RD}_\alpha(P\|Q),$$

and for $\alpha = \infty$*,*

$$Q(E) \geq P(E) / \mathrm{RD}_\infty(P\|Q).$$

Lemma 2 (Data Processing Inequality [VEH14, Thm. 9]). *Let $\alpha > 1$. For any function f, where P^f (respectively Q^f) denotes the distribution of $f(y)$ induced by sampling $y \leftarrow P$ (respectively $y \leftarrow Q$):*

$$\mathrm{RD}_\alpha(P^f \| Q^f) \leq \mathrm{RD}_\alpha(P \| Q).$$

Lemma 3 (Multiplicativity [Ros20, Prop. 2]). *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 $\alpha \in (1, \infty)$*

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

Lemma 4 (Adapted from [BS23b, Lem. 6]). *Let $\mathcal{R}$ be a polynomial ring over $\mathbb{Z}$ and ℓ be a positive integer. Let P, Q be two probability distributions over $\mathcal{R}$ and $e_1, \ldots, e_\ell$ be (possibly dependent) random variables over $S_B := \{ x \in \mathcal{R} \colon \|x\|_\infty \leq B \}$ for some $B \in \mathbb{Z}$, for which there exist $\alpha \in (1, \infty)$ and $\epsilon_{\mathrm{RD},\alpha} \geq 1$ such that for all $y \in S_B$, it holds that $\mathrm{Supp}(P + y) \subseteq \mathrm{Supp}(Q)$, and furthermore, $\mathrm{RD}_\alpha(P + y \| Q) \leq \epsilon_{\mathrm{RD},\alpha}$. Then,*

$$\mathrm{RD}_\alpha((P + e_1, \ldots, P + e_\ell) \| Q^\ell) \leq \epsilon_{\mathrm{RD},\alpha}^\ell.$$

2.3 Public-Key Encryption

We recall the definition of a secure public-key encryption scheme with nearly linear decryption.

Definition 3. *A public-key encryption scheme* PKE *over message space $\mathcal{M}$ is defined by the following* PPT *algorithms* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$*, where:*

- $(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$*: on input the security parameter λ, the setup algorithm outputs public parameters* pp*, a secret key* sk *and a public key* pk*.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$*: on input a plaintext $m \in \mathcal{M}$ and a public key* pk*, the encryption algorithm outputs a ciphertext* ct*.*
- $m' \leftarrow \mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$*: given a ciphertext* ct *and a secret key* sk*, the decryption algorithm outputs a plaintext $m' \in \mathcal{M}$ or the symbol $\perp$.*

Definition 4 (IND-CPA for PKE). *A* PKE *scheme is said to fulfill* $\mathsf{IND\text{-}CPA}$ *security for the security parameter λ, if for all* PPT *adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*

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

where $\mathsf{Expt}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}$ *is the experiment in Fig. 1 with* $\mathsf{ctr} := 0$ *at the beginning.*

We make use of the following property of an encryption scheme, called "nearly-linear" decryption, as used in [BKS19; BDG+19; BS23b].

$\mathsf{Expt}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathsf{PKE}}(1^\lambda,)$

1 : $(\mathsf{pp},\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$
2 : $(m^{(0)}, m^{(1)}) \leftarrow \mathcal{A}_1(\mathsf{pp},\mathsf{pk})$
3 : $b \leftarrow \{0,1\}$
4 : $\mathsf{ct}^{(b)} \leftarrow \mathsf{Enc}(m^{(b)},\mathsf{pk})$
5 : $b' \leftarrow \mathcal{A}_2(\mathsf{ct}^{(b)})$
6 : **return** $b = b'$

Fig. 1. Experiment for IND-CPA security of PKE schemes.

Definition 5 (Nearly Linear Decryption). *Let q, r be positive integers and $\mathcal{R}$ be a ring. Let $\mathsf{PKE} = (\mathsf{Setup},\mathsf{Enc},\mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{R}_q^r$. Suppose that Setup outputs a secret key $\mathsf{sk} \in \mathcal{R}_q^r$. Let $\beta = \beta(\lambda) \in \mathbb{N}$, $\epsilon = \epsilon(\lambda) \in [0,1]$. We say that PKE has (β,ϵ)-linear decryption if for any λ, $(\mathsf{pp},\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$, message $m \in \mathcal{M}$, ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(m,\mathsf{pk})$ it holds*

$$\langle \mathsf{sk}, \mathsf{ct} \rangle = m + e^{ct} \bmod q,$$

for some $e^{ct} \in \mathcal{R}_q$ such that $\Pr[\|e^{ct}\|_\infty \leq \beta] \geq 1 - \epsilon$ (where the probability is taken over the randomness of Setup and Enc). We refer to e^{ct} as the ciphertext noise. *We additionally assume the existence of an algorithm Decode such that, if we have that $\|e^{ct}\| \leq \beta$, then the following holds with probability 1.*

$$\mathsf{Decode}(\langle \mathsf{sk}, \mathsf{ct} \rangle) = m.$$

2.4 Fully Homomorphic Encryption

We recall the definition of a secure fully-homomorphic encryption scheme with nearly linear decryption.

Definition 6. *A fully-homomorphic encryption scheme FHE over a message space $\mathcal{M}$ and a set of admissible circuits $\mathcal{C}$ is defined by the following PPT algorithms $(\mathsf{Setup},\mathsf{Enc},\mathsf{Eval},\mathsf{Dec})$, where:*

- $(\mathsf{pp},\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda)$*: on input the security parameter λ, the setup algorithm outputs a set of public parameters pp, a secret key sk and a public key pk.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(m,\mathsf{pk})$*: on input a plaintext $m \in \mathcal{M}$ and a public key pk, the encryption algorithm outputs a ciphertext ct.*
- $\widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{ct}_1,\ldots,\mathsf{ct}_\kappa, C, \mathsf{pk})$*: given a circuit $C \in \mathcal{C}$ such that $C : \mathcal{M}^\kappa \to \mathcal{M}$, a set of κ ciphertexts $\{\mathsf{ct}_1,\ldots,\mathsf{ct}_\kappa\}$ and a public key pk, the evaluation algorithm outputs a ciphertext $\widehat{\mathsf{ct}}$.*
- $m' \leftarrow \mathsf{Dec}(\mathsf{ct},\mathsf{sk})$*: given a ciphertext ct and a secret key sk, the decryption algorithm outputs a plaintext m'.*

For fully-homomorphic encryption schemes, IND-CPA security is defined in the same manner as for the public-key setting, see Definition 4.

Note that we implicitly assume that the evaluation key (whether it is a key-switching key or a bootstrapping key) is a part of the public key pk. Note further that the Setup algorithm implicitly calls the KeyGen algorithm. This separation is not needed for the purposes of the paper, and we therefore make the simplification of collapsing the two algorithms into one.

Note that, in this work, we formulate all definitions in terms of *exact* (FHE) schemes, such as BGV [BGV14], BFV [FV12; Bra12] and TFHE [CGG+20].

We recall the definition of nearly linear decryption in the FHE case.

Definition 7 (Nearly Linear Decryption). *Let* $\mathsf{FHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *be an FHE scheme as in Definition 6. Denote the message space by* $\mathcal{M}$*, the set of admissible circuits* $\mathcal{C}$*, and the ciphertext space by* $\mathcal{R}_q^r$ *for some ciphertext modulus* q *and rank* r*. Further, assume that* Setup *outputs secret keys in* $\mathcal{R}_q^r$*. Let* $\beta = \beta(\lambda) \in \mathbb{N}$*, let* $\epsilon = \epsilon(\lambda) \in [0,1]$*. Let* $C \in \mathcal{C}$*, such that* $C : \mathcal{M}^\kappa \to \mathcal{M}$*. Let* $m_1, \dots, m_\kappa \in \mathcal{M}$ *be messages. Let* ct *be the ciphertext that was generated as a result of an honest execution of the* $\mathsf{Setup}(1^\lambda), (\mathsf{Enc}(m_i, \mathsf{pk}))_{i\in[\kappa]}$ *and* $\mathsf{Eval}(\mathsf{Enc}(m_1, \mathsf{pk}), \dots, \mathsf{Enc}(m_\kappa, \mathsf{pk}), C, \mathsf{pk})$ *algorithms. We say that* FHE *has* (β, ϵ)-linear decryption *if it holds that*

$$\langle \mathsf{ct}, \mathsf{sk} \rangle = m + e^{ct} \bmod q,$$

for some error $e^{ct} \in \mathcal{R}_q$ *such that* $\Pr[\|e^{ct}\| \leq \beta] \geq 1 - \epsilon$*, where this probability is taken over the randomness of the* $\mathsf{Setup}, \mathsf{Enc}$ *and* Eval *algorithms. We additionally assume the existence of an algorithm* Decode *such that, if we have that* $\|e\| \leq \beta$*, then the following holds with probability 1,*

$$\mathsf{Decode}(\langle \mathsf{ct}, \mathsf{sk} \rangle) = m.$$

3 Rényi Divergence for More Decision Problems

In the original definition of public sampleability [BLR+18, Sec. 4.1], the distinguisher $\mathcal{A}$ has no impact on the distributions they have to distinguish. This is sufficient when applying public sampleability to prove the semantic security of a plain public-key encryption scheme, as done in [BLR+18, Sec. 4.2]. However, the original definition reaches its limits when wanting to apply public sampleability if the adversary additionally has access to oracles. In the following, we propose a more flexible framework for public sampleability and hence we extend the use of the Rényi divergence to a larger class of decision problems.

Definition 8 (Decision Problem with Adaptive Adversary Input). *Let* τ *and* $(m_j, \bar{m}_j)_{j\in[\tau]}$ *be positive integers with* $\bar{m}_j \leq m_j$*, defining* $m := \sum_{j=1}^{\tau} m_j$ *and* $\bar{m} := \sum_{j=1}^{\tau} \bar{m}_j$*. Let* $\Phi(\boldsymbol{\eta}) = (\Phi_1(\boldsymbol{\eta}_1), \dots, \Phi_\tau(\boldsymbol{\eta}_\tau))$ *be a distribution parametrized by* $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_\tau) \in T$ *with support* $(S_1, \dots, S_\tau) = S$ *with* $S_j \subset \mathcal{R}^{m_j}$ *for fixed ring* $\mathcal{R}$*. For* $j \in [\tau]$*, we denote by* $\mathsf{Part}_j(\mathbf{r}_j) \in \mathcal{R}^{\bar{m}_j}$ *some deterministic function*

that outputs $\bar{m}_j$ coefficients of $\mathbf{r}_j \in \mathcal{R}^{m_j}$. By concatenating vectors, they define a deterministic function $\mathsf{Part}\colon \mathcal{R}^m \to \mathcal{R}^{\bar{m}}$. *Further, let $D_0(\boldsymbol{\eta}, \mathbf{r})$ and $D_1(\boldsymbol{\eta}, \mathbf{r})$ be two distributions determined by the same parameter $\boldsymbol{\eta} \in T$ and $\mathbf{r} \in S$. We define the* decision problem P with adaptive adversary input *as follows. Let $\mathcal{A}$ be a* PPT *adversary, who, given as input* smpl*, has to decide whether* $\mathsf{smpl} \leftarrow X_0$ *or* $\mathsf{smpl} \leftarrow X_1$*, where for $b \in \{0,1\}$*

$$X_b := \{\, \mathsf{smpl} := (\bar{\mathbf{r}}, x) \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), \mathbf{r} \leftarrow \Phi(\boldsymbol{\eta}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r}) \,\},$$

with $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_\tau)$, and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\tau)$, such that, $\boldsymbol{\eta}_j \leftarrow \mathcal{A}(\mathsf{Part}_{j-1}(\mathbf{r}_{j-1}))$, and $\mathbf{r}_j \leftarrow \Phi_j(\boldsymbol{\eta}_j)$, for $j \in [\tau]$. By default, we set $\mathbf{r}_0 = \mathsf{Part}_0(\mathbf{r}_0) := \mathbf{0}$. We call τ the number of iterations*. For all iterations j, $\mathcal{A}$ at iteration j also gets as input all partial information on $\mathbf{r}$ from previous iterations, and similarly Φ_j gets as input all previous full information on $\mathbf{r}$. We omit writing it explicitly for ease of notation.*

The main difference to the original definition is that, alongside the parameter $\mathbf{r}$, we introduce an additional parameter $\boldsymbol{\eta}$, controlled by the adversary. Both $\boldsymbol{\eta}$ and $\mathbf{r}$ possibly dependent on each other.

Definition 9 (Public Sampleability). *Let P be a decision problem between distribution X_0 and X_1 with adaptive adversary input as in Definition 8. We say that P fulfills* public sampleability *if there exists a sampling algorithm* Sample *with runtime* $\mathsf{T}_{\mathsf{Sample}}$ *such that for all $(\boldsymbol{\eta}, \mathbf{r}, b) \in T \times S \times \{0,1\}$, given any sample* $\mathsf{smpl} = (\bar{\mathbf{r}}, x)$ *with $x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})$ and $\bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r})$,*

- $\mathsf{Sample}(\boldsymbol{\eta}, 0, \mathsf{smpl})$ *outputs a fresh sample* $\mathsf{smpl}' = (\bar{\mathbf{r}}, x')$ *with x' distributed as $D_0(\mathbf{r}, \boldsymbol{\eta})$,*
- $\mathsf{Sample}(\boldsymbol{\eta}, 1, \mathsf{smpl})$ *outputs a fresh sample* $\mathsf{smpl}' = (\bar{\mathbf{r}}, x')$ *with x' distributed as $D_1(\mathbf{r}, \boldsymbol{\eta})$.*

Note that $\mathbf{r}$, $\bar{\mathbf{r}}$ and $\boldsymbol{\eta}$ are fixed, and only x' needs to be a fresh sample.

We remark that in the original definition of public sampleability [BLR+18, Sec. 4.1], neither the adversary $\mathcal{A}$ nor the sampler Sample were given $\bar{\mathbf{r}}$. Separating the partial information on $\mathbf{r}$ from the sample x is merely a notational change, as one can always append $\bar{\mathbf{r}}$ to x. This was for instance done in [BLR+18, Sec. 4.2], where in fact the whole parameter $\mathbf{r}$ was given out.

We further highlight that, as Sample is defined with respect to a fixed $\boldsymbol{\eta}$ and $\mathbf{r}$, it works the same way, whether $\boldsymbol{\eta}$ is adversarially chosen and depending on $\bar{\mathbf{r}}$, or not. Hence, our definition of public sampleability matches with the original definition given in [BLR+18]. Intuitively, this is also why the original proof carries over to the adaptive adversary input setting.

Theorem 1. *For a given ring $\mathcal{R}$ and integer m, let $\Phi(\boldsymbol{\eta})$ and $\Phi'(\boldsymbol{\eta})$ be two distributions parametrized by $\boldsymbol{\eta} \in T$ with support S and S' respectively, where $S \subseteq S' \subseteq \mathcal{R}^m$. Let P, P' be two decision problems with adaptive adversary input using the same distributions $D_b(\boldsymbol{\eta}, \mathbf{r})$ for $b \in \{0,1\}$, with the only difference being*

that in P, the parameter $\mathbf{r}$ is sampled from $\Phi(\boldsymbol{\eta})$, whereas in P' it is sampled from $\Phi'(\boldsymbol{\eta})$. Moreover, we assume that P' is publicly sampleable with sampling algorithm Sample *and runtime* $\mathsf{T}_{\mathsf{Sample}}$*. Then, given an adversary $\mathcal{A}$ for problem P with advantage* Adv *and runtime* T*, we can construct an adversary $\mathcal{A}'$ for problem P' with advantage* Adv' *and* T'*, where for $\alpha \in (1, \infty)$*

$$\mathsf{Adv}' \geq \frac{\mathsf{Adv}^{1+\frac{\alpha}{\alpha-1}}}{2^{3+\frac{\alpha}{\alpha-1}} \cdot \mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta}))},$$

and

$$\mathsf{T}' \leq \frac{64}{\mathsf{Adv}^2} \log_2 \left(\frac{2^{\frac{\alpha}{\alpha-1}+5} \cdot \mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta}))}{\mathsf{Adv}^{\frac{\alpha}{\alpha-1}+1}} \right) (\mathsf{T}_{\mathsf{Sample}} + \mathsf{T}) + \mathsf{T}.$$

For $\alpha = \infty$, it holds

$$\mathsf{Adv}' \geq \frac{\mathsf{Adv}^2}{2^4 \cdot \mathrm{RD}_\infty(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta}))},$$

and

$$\mathsf{T}' \leq \frac{64}{\mathsf{Adv}^2} \log_2 \left(\frac{2^6 \cdot \mathrm{RD}_\infty(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta}))}{\mathsf{Adv}^2} \right) (\mathsf{T}_{\mathsf{Sample}} + \mathsf{T}) + \mathsf{T}.$$

The concrete advantages and runtimes slightly differ with respect to the original statement [BLR+18, Thm. 4.2]. This is due to corrected typos.

Proof. Let Adv be the advantage of $\mathcal{A}$. It is defined as $\mathsf{Adv} = |p_1 - p_0|$, where for $b \in \{0, 1\}$

$$\begin{aligned} p_b &:= \Pr[\mathcal{A}(\mathsf{smpl}) = 1 \mid \mathsf{smpl} \leftarrow X_b] \\ &= \Pr[\mathcal{A}(\bar{\mathbf{r}}, x) = 1 \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), \mathbf{r} \leftarrow \Phi(\boldsymbol{\eta}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})] \\ &= \sum_{\mathbf{r} \in \mathcal{R}^m} \Pr[\mathbf{r} \leftarrow \Phi(\boldsymbol{\eta})] \cdot \Pr[\mathcal{A}(\bar{\mathbf{r}}, x) = 1 \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})] \\ &=: \sum_{\mathbf{r} \in \mathcal{R}^m} \Pr[\mathbf{r} \leftarrow \Phi(\boldsymbol{\eta})] \cdot p_b(\mathbf{r}), \end{aligned}$$

where we use the same notation as in the proof of [BLR+18]. Without loss of generality, we assume $p_1 > p_0$.

Let $\mathcal{A}'$ be given $\mathsf{smpl} = (\bar{\mathbf{r}}, x)$ with $\bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r})$, $\mathbf{r} \leftarrow \Phi'(\boldsymbol{\eta})$ and $x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})$ for some unknown $b \in \{0, 1\}$. The goal of $\mathcal{A}'$ is to correctly guess b.

The high-level idea is the following: The adversary $\mathcal{A}'$ runs the public sample algorithm Sample many times on $(\boldsymbol{\eta}, 0, \mathsf{smpl})$ and $(\boldsymbol{\eta}, 1, \mathsf{smpl})$ in order to get fresh samples (for fixed $\mathbf{r}$ and $\boldsymbol{\eta}$). They then run $\mathcal{A}$ on those samples to obtain good approximations $\hat{p}_0$ and $\hat{p}_1$ of the acceptance probabilities $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$. If $\hat{p}_1$ is significantly larger than $\hat{p}_0$, the adversary $\mathcal{A}'$ runs $\mathcal{A}$ on the given sample smpl and outputs the bit that $\mathcal{A}$ outputs. Else, $\mathcal{A}'$ outputs a random bit. In other words, $\mathcal{A}'$ behaves as $\mathcal{A}$ if the latter performed well on the tested inputs. If

not, $\mathcal{A}'$ simply guesses. Note that the fact that $\boldsymbol{\eta}$ is adversarially chosen and may depend on $\bar{\mathbf{r}}$ is not changing the proof as the sampler Sample works for fixed $\boldsymbol{\eta}, \mathbf{r}$. That is, the fresh samples output by Sample are all independent, for fixed $\boldsymbol{\eta}$ and $\mathbf{r}$ (and thus $\bar{\mathbf{r}}$).

For an Adv' to be determined later, the adversary $\mathcal{A}'$ runs the public sample algorithm Sample $N \geq 32\mathsf{Adv}^{-2}\log_2(4/\mathsf{Adv}')$ many times on $(\boldsymbol{\eta}, 0, \mathsf{smpl})$ and $(\boldsymbol{\eta}, 1, \mathsf{smpl})$ in order to get fresh samples (for fixed $\mathbf{r}$ and $\boldsymbol{\eta}$). They then run $\mathcal{A}$ on those samples to obtain estimates $\hat{p}_0$ and $\hat{p}_1$ of the acceptance probabilities $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$.

The Hoeffding's bound implies that the estimation errors $|\hat{p}_0 - p_0|$ and $|\hat{p}_1 - p_1|$ are $< \mathsf{Adv}/8$ except with probability $< \mathsf{Adv}'/2$ over the randomness of Sample.

If $\hat{p}_1 - \hat{p}_0 > \mathsf{Adv}/4$, the adversary $\mathcal{A}'$ runs $\mathcal{A}$ on the given sample smpl and outputs the bit that $\mathcal{A}$ outputs. Else, $\mathcal{A}'$ outputs a random bit.

As for Adv, we introduce a similar notation for Adv', the advantage of $\mathcal{A}'$. It is defined as $\mathsf{Adv}' = |p'_1 - p'_0|$, where for $b \in \{0,1\}$

$$\begin{aligned} p'_b &:= \Pr[\mathcal{A}'(\mathsf{smpl}) = 1 \mid \mathsf{smpl} \leftarrow X'_b] \\ &= \Pr[\mathcal{A}'(\bar{\mathbf{r}}, x) = 1 \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), \mathbf{r} \leftarrow \Phi'(\boldsymbol{\eta}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})] \\ &= \sum_{\mathbf{r} \in \mathcal{R}^m} \Pr[\mathbf{r} \leftarrow \Phi'(\boldsymbol{\eta})] \cdot \Pr[\mathcal{A}'(\bar{\mathbf{r}}, x) = 1 \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})] \\ &=: \sum_{\mathbf{r} \in \mathcal{R}^m} \Pr[\mathbf{r} \leftarrow \Phi'(\boldsymbol{\eta})] \cdot p'_b(\mathbf{r}), \end{aligned}$$

Let S_1 denote the set of $\mathbf{r}$'s such that $p_1(\mathbf{r}) - p_0(\mathbf{r}) \geq \mathsf{Adv}/2$, let S_2 denote the set of $\mathbf{r}$'s such that $p_1(\mathbf{r}) - p_0(\mathbf{r}) \geq 0$, and S_3 denote all the remaining $\mathbf{r}$'s, that is, $S_3 = \mathcal{R}^m \setminus (S_1 \cup S_2)$. We now proceed by separating the three cases. Fro $b \in \{0,1\}$ and $j \in \{1,2,3\}$, we denote by $p'_b(\mathbf{r} \mid \mathbf{r} \in S_j)$ the corresponding probability conditioned on $\mathbf{r}$ lying in the set S_j, that is $p'_b(\mathbf{r} \mid \mathbf{r} \in S_j) = \Pr[\mathcal{A}'(\bar{\mathbf{r}}, x) = 1 \mid \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}), x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r}), \mathbf{r} \in S_j]$.

- Case $\mathbf{r} \in S_1$: Except with probability $< \mathsf{Adv}'$ over the randomness of Sample, we have $\hat{p}_1 - \hat{p}_0 > \mathsf{Adv}/4$ and thus $\mathcal{A}'$ outputs $\mathcal{A}(\mathsf{smpl})$. In the case $b = 1$, we have $p'_1(\mathbf{r} \mid \mathbf{r} \in S_1) \geq p_1(\mathbf{r}) - \mathsf{Adv}'$ and in the case $b = 0$, we have $p'_0(\mathbf{r} \mid \mathbf{r} \in S_1) \leq p_0(\mathbf{r}) + \mathsf{Adv}'$.
- Case $\mathbf{r} \in S_2$: Let $u(\mathbf{r})$ be the probability over the randomness of Sample that $\hat{p}_1 - \hat{p}_0 > \mathsf{Adv}/4$. Then $\mathcal{A}'$ will output $\mathcal{A}(\mathsf{smpl})$ with probability $u(\mathbf{r})$ and a uniform bit with probability $1 - u(\mathbf{r})$. Thus, in the case of $b = 1$, we have $p'_1(\mathbf{r} \mid \mathbf{r} \in S_2) = u(\mathbf{r})p_1(\mathbf{r}) + (1 - u(\mathbf{r}))/2$, and in the case $b = 0$, we have $p'_0(\mathbf{r} \mid \mathbf{r} \in S_2) = u(\mathbf{r})p_0(\mathbf{r}) + (1 - u(\mathbf{r}))/2$.
- Case $\mathbf{r} \in S_3$: Except with probability $< \mathsf{Adv}'$ over the randomness of Sample, we have $\hat{p}_1 - \hat{p}_0 < \mathsf{Adv}/4$ and $\mathcal{A}'$ outputs a uniformly random bit. Thus, in the case $b = 1$, we have $p'_1(\mathbf{r} \mid \mathbf{r} \in S_3) \geq 1/2 - \mathsf{Adv}'$, and for $b = 0$, we have $p'_0(\mathbf{r} \mid \mathbf{r} \in S_3) \leq 1/2 + \mathsf{Adv}'$.

Putting everything together, it holds

$$\begin{aligned}
\mathsf{Adv}' &\geq \sum_{j\in\{1,2,3\}} \sum_{\mathbf{r}\in S_j} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})]\cdot (p_1'(\mathbf{r}\mid \mathbf{r}\in S_j) - p_0'(\mathbf{r}\mid \mathbf{r}\in S_j)) \\
&\geq \sum_{\mathbf{r}\in S_1} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})]\cdot (p_1(\mathbf{r}) - p_0(\mathbf{r}) - 2\mathsf{Adv}') \\
&\quad + \sum_{\mathbf{r}\in S_2} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})]\cdot u(\mathbf{r})\cdot (p_1(\mathbf{r}) - p_0(\mathbf{r})) \\
&\quad + \sum_{\mathbf{r}\in S_3} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})]\cdot (-2\mathsf{Adv}') \\
&\geq \mathsf{Adv}/2 \cdot \sum_{S_1} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})] - 2\mathsf{Adv}',
\end{aligned}$$

where the second summand was lower bounded by 0.

By Lemma 1 for $\alpha \in (1,\infty)$, it holds

$$\sum_{S_1} \Pr[\mathbf{r}\leftarrow \Phi'(\boldsymbol{\eta})] \geq \left(\sum_{S_1} \Pr[\mathbf{r}\leftarrow \Phi(\boldsymbol{\eta})]\right)^{\alpha/\alpha-1} / \mathrm{RD}_\alpha(\Phi\|\Phi').$$

We claim that $\sum_{S_1} \Pr[\mathbf{r}\leftarrow \Phi(\boldsymbol{\eta})] \geq \mathsf{Adv}/2$.

Proof. For simplicity, we denote by $\Phi(\boldsymbol{\eta},\mathbf{r}) = \Pr[\mathbf{r}\leftarrow\Phi(\boldsymbol{\eta})]$.

$$\begin{aligned}
\mathsf{Adv} &= \sum_{\mathbf{r}\in\mathcal{R}^m} \Phi(\boldsymbol{\eta},\mathbf{r})(p_1(\mathbf{r}) - p_0(\mathbf{r})) \\
&= \sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r})(p_1(\mathbf{r}) - p_0(\mathbf{r})) + \sum_{\mathbf{r}\in\mathcal{R}^m\setminus S_1} \Phi(\boldsymbol{\eta},\mathbf{r})(p_1(\mathbf{r}) - p_0(\mathbf{r})) \\
&\leq \sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r}) + \mathsf{Adv}/2 \sum_{\mathbf{r}\in\mathcal{R}^m\setminus S_1} \Phi(\boldsymbol{\eta},\mathbf{r}) \\
&= \sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r}) + \mathsf{Adv}/2(1 - \sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r})) \\
&= (1-\mathsf{Adv}/2)\sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r}) + \mathsf{Adv}/2,
\end{aligned}$$

which implies $\sum_{\mathbf{r}\in S_1} \Phi(\boldsymbol{\eta},\mathbf{r}) = \sum_{S_1} \Pr[\mathbf{r}\leftarrow\Phi(\boldsymbol{\eta})] \geq \mathsf{Adv}/2$. ■

The proof can then be completed by setting

$$\begin{aligned}
\mathsf{Adv}' &\geq \frac{\mathsf{Adv}}{6\cdot \mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta})\|\Phi'(\boldsymbol{\eta}))} \left(\frac{\mathsf{Adv}}{2}\right)^{\frac{\alpha}{\alpha-1}} \\
&\geq \frac{\mathsf{Adv}^{1+\frac{\alpha}{\alpha-1}}}{2^{3+\frac{\alpha}{\alpha-1}}\cdot \mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta})\|\Phi'(\boldsymbol{\eta}))}.
\end{aligned}$$

The proof for $\alpha=\infty$ follows by applying Lemma 1 in a similar way. □

4 Threshold PKE

As a warm-up, we focus in this section solely on public-key encryption schemes. In [BS23b], a simple construction of threshold PKE schemes from lattice assumptions was proposed. However, it was only shown to fulfill a one-way security notion. The one-way secure scheme was in a second step transformed into one which fulfills the desired semantic (also called indistinguishability) security. In this section, we prove that the scheme constructed in [BS23b] already fulfills semantic security. Hence, the transformation in [BS23b] was redundant. We start by recalling the definition of correct and secure threshold public-key encryption, as well as the construction from [BS23b]. We then use the extended public sampleability framework from Section 3 to directly show semantic security.

4.1 Threshold PKE

Definition 10. *A threshold public-key encryption scheme* ThPKE *over message space* $\mathcal{M}$ *is defined by the following* PPT *algorithms* (Setup, Enc, PartDec, Combine)*:*

- $(\mathsf{pp}, \mathsf{sk}_1, \ldots \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$*: on input the security parameter* λ*, the number of parties* n *and a threshold value* $t \in \{2, \ldots, n\}$*, the setup algorithm outputs public parameters* pp*, the list of secret key shares* $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ *and the public key* pk*.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$*: on input a public key* pk *and a plaintext* m*, the encryption algorithm outputs a ciphertext* ct *encrypting* m *under* pk*.*
- $\mathsf{p}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathsf{sk}_i)$*: on input the secret key share* sk_i *corresponding to party* P_i *and a ciphertext* ct*, the partial decryption algorithm outputs a partial decryption* p_i *corresponding to party* P_i *for* $i \in [n]$*.*
- $m \leftarrow \mathsf{Combine}((\mathsf{p}_i)_{i \in S})$*: on input a vector of partial decryptions* p_i*, for* $i \in S$*, where* $S \subset [n]$*, and a ciphertext* ct*, the combining algorithm outputs a plaintext* $m \in \mathcal{M}$ *or the symbol* $\perp$*.*

Definition 11 (Correctness of ThPKE**).** *A Threshold public-key encryption scheme* ThPKE *over message space* $\mathcal{M}$ *is called* correct*, if there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that for all* λ, n, t, S *with* $S \subset [n]$ *of size at least* t *and for all* $m \in \mathcal{M}$ *the following holds. For* $(\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$*,* $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$ *and* $\mathsf{p}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathsf{sk}_i)$ *with* $i \in S$*, it holds*

$$\Pr[\mathsf{ThPKE.Combine}((\mathsf{p}_i)_{i \in S}) = m] = 1 - \mathsf{negl}(\lambda).$$

Moreover, we assume that the randomness for encryption is sampled from a randomness distribution $\mathcal{X}$ and write $\mathsf{ct} := \mathsf{Enc}(m, \mathsf{pk}; \rho)$ for $\rho \leftarrow \mathcal{X}$ if we want to make the randomness explicit.

Indistinguishability. We define ℓ-IND-CPA security for ThPKE. Informally, this security notion corresponds to the standard IND-CPA security of public-key encryption schemes (Definition 4), where the adversary gets additionally the secret key shares of some statically corrupted parties, as well as (ℓ times) access to a partial decryption oracle (on honestly encrypted messages).

Definition 12 (ℓ-IND-CPA **for** ThPKE)**.** *A Threshold PKE scheme* ThPKE *is said to fulfill* ℓ-IND-CPA *security for the security parameter* λ*, the threshold parameters* n, t *and the query bound* ℓ*, if for all* PPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$

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

where $\mathsf{Expt}^{\ell\text{-IND-CPA}}_{\mathcal{A},\mathsf{ThPKE}}$ *is the experiment in Fig. 2.*

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

1 : $\mathsf{ctr} := 0$
2 : $(\mathsf{pp}, \mathsf{sk}_1, ..., \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$
3 : $S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk}) \colon S \subset [n] \wedge |S| < t$
4 : $(m^{(0)}, m^{(1)}) \leftarrow \mathcal{A}_2^{\mathsf{OPartDec}}((\mathsf{sk}_i)_{i \in S})$
5 : $b \leftarrow \{0, 1\}$
6 : $\mathsf{ct}^{(b)} \leftarrow \mathsf{Enc}(m^{(b)}, \mathsf{pk})$
7 : $b' \leftarrow \mathcal{A}_3(\mathsf{ct}^{(b)})$
8 : **return** $b = b'$

$\mathsf{OPartDec}(m)$

1 : $\mathsf{ctr} = \mathsf{ctr} + 1$
2 : **if** $\mathsf{ctr} > \ell$ **then return** $\perp$
3 : $\rho \leftarrow \mathcal{X}$
4 : $\mathsf{ct} := \mathsf{Enc}(m, \mathsf{pk}; \rho)$
5 : **for** $i \in [n] \setminus S$:
6 : $\quad \mathsf{p}_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$
7 : $\quad \vec{\mathsf{p}} := (\mathsf{p}_i)_{i \in [n] \setminus S}$
8 : **return** $(\rho, \mathsf{ct}, \vec{\mathsf{p}})$

Fig. 2. Experiment for ℓ-IND-CPA security of ThPKE schemes.

Remark 1. We highlight that this security notion is somewhat orthogonal to the security achieved in [MS23]. Theirs provides computational simulation-based security, but cannot allow for the encryption randomness to be provided to the adversary. Ours is only game-based with an a-priori bound on the number of decryption queries, but can give out the encryption randomness. Providing the encryption randomness to the adversary is a more realistic scenario: the adversary might convince the honest parties to participate in partial decryptions of ciphertexts the adversary honestly prepared. This also explains how [MS23] could go below the lower bound of [BS23a, Sec.6.2].

4.2 Construction

We recall in Figure 3 the blueprint construction from [BS23b], specialized to the public-key encryption setting. The construction works over the ring $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for some degree-d irreducible polynomial f. It uses three main ingredients: a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Decode})$ with nearly linear decryption and $\mathsf{sk} \in \mathcal{R}_q^r$, a linear secret sharing scheme $\mathsf{LSS} = (\mathsf{Share}, \mathsf{Rec})$ with $\mathsf{Share} : \mathcal{R}_q^r \to (\mathcal{R}_q^{rL})^n$, and a noise distribution $\mathcal{D}_{\mathsf{flood}}$ over $\mathcal{R}_q$.

We recall the definition of IND-CPA secure public-key encryption schemes in Section 2.3. There, we also recall the formal definition of PKE having nearly

linear decryption. Informally, we say that PKE has (β, ϵ)-nearly linear decryption if for every honestly generated secret key sk and ciphertext ct, it holds (except with probability ϵ over key and ciphertext generation) that $\langle \mathsf{ct}, \mathsf{sk} \rangle = m + e^{\mathrm{ct}}$, where m is the (encoded) encrypted message and e^{ct} is called the ciphertext noise and has infinity norm at most β. To recover m, one runs Decode on $m + e^{\mathrm{ct}}$.

Correctness (under specific parameter constraints) of this construction was proven in [BS23b, Thm. 5] in the more general fully-homomorphic case.

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

1 : $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$
2 : $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{LSS.Share}(\mathsf{sk})$
3 : **return** $(\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk})$

$\mathsf{ThPKE.Enc}(m, \mathsf{pk})$

1 : $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}(m, \mathsf{pk})$
2 : **return** ct

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

1 : **parse** $\mathsf{sk}_i := (\mathsf{sk}_{i,1}, \ldots, \mathsf{sk}_{i,L})$
2 : **for** $j \in [L]$:
3 : $\quad e_j^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{flood}}$
4 : $\quad \mathsf{p}_j = \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle + e_j^{\mathrm{fl}} \bmod q$
5 : **return** $\mathsf{p}_i = (\mathsf{p}_j)_{j \in [L]}$

$\mathsf{ThPKE.Combine}((\mathsf{p}_i)_{i \in S})$

1 : $y = \mathsf{LSS.Rec}_S((\mathsf{p}_i)_{i \in S})$
2 : **return** $\mathsf{PKE.Decode}(y)$

Fig. 3. Construction of ThPKE in [BS23b] from PKE and LSS schemes using distribution $\mathcal{D}_{\mathsf{flood}}$.

Remark 2. We highlight that this construction template covers a large class of lattice-based PKE schemes, in particular the Kyber scheme [SAB+22] standardized by NIST. The concrete distributions for key generation and sampling encryption randomness in the underlying PKE scheme can be chosen in a flexible manner, as long as the resulting scheme fulfills nearly linear decryption. Moreover, any flooding distribution and corresponding simulation distribution can be freely chosen, as long as they allow for a small and concise Rényi divergence.

4.3 Indistinguishability

For the rest of the section, we prove this construction to fulfill ℓ-IND-CPA security in the full threshold case, i.e., $t = n$. We concentrate on the full-threshold case for didactic reasons and prove in Appendix B.1 that the proof generalizes to arbitrary thresholds. As long as the reconstruction of LSS has the right properties, going from full to arbitrary thresholds is relatively straight forward and mimics essentially the proof of [BS23b]. When restricting to the full threshold case, we can use the simple and widespread additive secret sharing. The key step in the

proof of Theorem 2 is Lemma 6, applying the extended public sampleability framework.

Theorem 2. *Set $t = n$. Let* LSS *denote additive secret sharing over $\mathcal{R}_q$ which fulfills privacy, linearity and reconstruction. Let* PKE *be an* IND-CPA *secure public-key encryption scheme with (β, ϵ)-nearly linear decryption. Let $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$ be two distributions over $\mathcal{R}_q$, where $\mathrm{RD}_\alpha(\mathcal{D}_{\mathsf{flood}} + B \| \mathcal{D}_{\mathsf{sim}}) \leq \epsilon_{\mathrm{RD},\alpha}$ for some $\alpha \in (1, \infty)$, some $\epsilon_{\mathrm{RD},\alpha} > 1$, and for all $B \in \mathcal{R}_q$ with $\|B\|_\infty \leq \beta$. Then, the* ThPKE *constructed as in Figure 3 is ℓ-*IND-CPA *secure. More precisely, assume there exists an adversary $\mathcal{A}$ against ℓ-*IND-CPA *of* ThPKE *with advantage* Adv*, then we can construct an adversary $\mathcal{A}'$ against* IND-CPA *of* PKE *with advantage* Adv$'$*, where*

$$\mathsf{Adv}' \geq \frac{(\mathsf{Adv} - \ell\epsilon)^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}} \cdot (\epsilon_{\mathrm{RD},\alpha})^{\ell}}.$$

For given $\mathcal{D}_{\mathsf{flood}}$, β and ϵ (coming from the construction) and aimed number of partial decryption queries ℓ, the loss in the reduction can be minimized by simultaneously choosing the Rényi divergence order α and the distribution $\mathcal{D}_{\mathsf{sim}}$ (defining the bound $\epsilon_{\mathrm{RD},\alpha}$). Note that for the special case where both $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$ are discrete Gaussians, one can reduce the dependency in the exponent of $\epsilon_{\mathrm{RD},\alpha}$ from ℓ to $\sqrt{\ell}$.

Proof. Let $\mathcal{A}$ denote a PPT adversary against ℓ-IND-CPA security of ThPKE. Note that ℓ-IND-CPA security can be seen as the standard IND-CPA security, where the adversary additionally obtains corrupted secret key shares and gets adaptive access to a partial decryption oracle on honestly encrypted messages. The overall idea of the proof is to modify the ℓ-IND-CPA game of ThPKE in a sequence of hybrid games, so that in the final game, corrupted secret key shares and the partial decryption queries are independent of the underlying secret key.

Hybrid_0 : This is the real ℓ-IND-CPA security experiment as detailed in Figure 2. As we are in the full-threshold case, hence $t = n$, the adversary corrupts $n - 1$ parties, leaving one single honest party. The view of the adversary is given by (possibly dependent)

$$\mathsf{smpl} = \left(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i \in S}, (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)_{j \in [\ell]}, \mathsf{ct}^{(b)}\right), \tag{1}$$

where $(\mathsf{p}_j)_{j \in [\ell]}$ are the partial decryption shares of the single honest party. It holds $\mathsf{Adv}^{\ell\text{-IND-CPA}}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThPKE}}(\mathcal{A})$.

Hybrid_1 : In the first hybrid experiment, we slightly modify the partial decryption oracle OPartDec. On input m, and fresh ciphertext ct, before outputting the tuple $(\rho, \mathsf{ct}, \mathsf{p})$, it checks whether $\langle \mathsf{ct}, \mathsf{sk} \rangle = m + e^{\mathrm{ct}}$ for some $e^{\mathrm{ct}} \in \mathcal{R}_q$ with $\|e^{\mathrm{ct}}\|_\infty \leq \beta$. If not, the game aborts.

Hybrid_2 : In the next hybrid experiment, we replace the partial decryption oracle by the simulated oracle $\mathsf{OSimPartDec}$, as described in Figure 4. The important change is that in the simulated oracle, the partial decryption p of the single honest party is computed only using knowledge the adversary will have at the end of the game: namely message m, ciphertext ct and corrupted key shares $(\mathsf{sk}_i)_{i\in S}$. Most importantly, it is not depending on the honest key share sk_{i^*}, with $i^* \in [n]\setminus S$. Moreover, the flooding distribution $\mathcal{D}_{\mathsf{flood}}$ is replaced by a possibly different distribution $\mathcal{D}_{\mathsf{sim}}$.

$\mathsf{OSimPartDec}(m)$

1 : $\mathsf{ctr} = \mathsf{ctr} + 1$
2 : **if** $\mathsf{ctr} > \ell$ **then return** $\perp$
3 : $\rho \leftarrow \mathcal{X}$
4 : $\mathsf{ct} := \mathsf{Enc}(m, \mathsf{pk}; \rho)$
5 : $e^{\mathrm{ct}} := \langle \mathsf{ct}, \mathsf{sk}\rangle - m$
6 : **if** $\left\|e^{\mathrm{ct}}\right\|_\infty > \beta$ **then return** $\perp$
7 : $e^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{sim}}$
8 : $\mathsf{p} = m - \sum_{i\in S}\langle \mathsf{ct}, \mathsf{sk}_i\rangle + e^{\mathrm{fl}}$
9 : **return** $(\rho, \mathsf{ct}, \mathsf{p})$

$\mathsf{SimSetup}(1^\lambda)$

1 : $(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{PKE.Setup}(1^\lambda)$
2 : $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{LSS.Share}(\mathbf{0})$
3 : **return** $(\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk})$

Fig. 4. Simulated partial decryption oracle (using distribution $\mathcal{D}_{\mathsf{sim}}$) and simulated setup for ℓ-IND-CPA security of ThPKE schemes in the full-threshold case.

Hybrid_3 : Once partial decryption queries are not depending on the honest key share anymore, we can replace the honest setup by a simulated setup $\mathsf{SimSetup}$, as presented in Figure 4. Instead of providing key shares of sk to the adversary, now dummy key shares of $\mathbf{0}$ are provided to them. As in this hybrid neither the corrupted key shares nor the partial decryption queries depend on the actual secret key sk, the experiment corresponds to the IND-CPA security game of the underlying PKE. It holds $\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$.

The rest of the proof consists in bounding how much the advantage of the adversary decreases between the hybrids.

Lemma 5. *Any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_0 *is already defining an adversary against* Hybrid_1*, where*

$$\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThPKE}}(\mathcal{A}) \geq \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThPKE}}(\mathcal{A}) - \epsilon\ell.$$

Proof. By assumption, PKE has (β, ϵ)-nearly linear decryption. Both hybrids only differ in the case of an abort, which by the union bound happens with

probability

$$\Pr[\exists j \in [\ell] \colon \left\|e_j^{\mathrm{ct}}\right\|_\infty > \beta] \leq \sum_{j\in[\ell]} \Pr[\left\|e_j^{\mathrm{ct}}\right\|_\infty > \beta] \leq \ell\epsilon.$$

■

Lemma 6. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_1*, we can construct an adversary* $\mathcal{A}'$ *for* Hybrid_2 *such that it holds*

$$\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThPKE}}(\mathcal{A}') \geq \frac{\left(\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThPKE}}(\mathcal{A})\right)^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}} \cdot (\epsilon_{\mathrm{RD},\alpha})^\ell}.$$

Proof. To prove the lemma, we make use of Theorem 1. To do so, we formulate Hybrid_1 (resp. Hybrid_2) as decision problem P (resp. P') of the form $D_b(\boldsymbol{\eta}, \mathbf{r})$ with adversarially chosen $\boldsymbol{\eta}$ and $\mathbf{r} \leftarrow \Phi$ (reps. $\mathbf{r} \leftarrow \Phi'$). After this, we argue public sampleability.

We start by formulating Hybrid_1 as decision problem P with adaptive adversary input as defined in Definition 8. We thus detail the adaptive adversary input parameter $\boldsymbol{\eta}$, the distribution Φ as well as the partial information on elements of Φ, defined by Part. Finally, we describe the distribution $D_b(\boldsymbol{\eta}, \mathbf{r})$. As $\boldsymbol{\eta}$ and Φ are defined adaptively, we describe them by detailing each iteration.

Iteration 1. On any output $\boldsymbol{\eta}_1 \leftarrow \mathcal{A}$ by the adversary, we define $\Phi_1(\boldsymbol{\eta}_1) = \Phi_1$ as follows: run $\mathbf{r}_1 := (\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, n)$ and set $\mathsf{Part}_1(\mathbf{r}_1) = (\mathsf{pp}, \mathsf{pk})$. Note that Φ_1 is independent of $\boldsymbol{\eta}_1$, so without loss of generality, we omit $\boldsymbol{\eta}_1$ in the following.

Iteration 2. On output $\boldsymbol{\eta}_2 := S \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk})$, we define $\Phi_2(\boldsymbol{\eta}_2)$ as follows: set $\mathbf{r}_2 = \mathbf{r}_1$ and $\mathsf{Part}_2(\mathbf{r}_2) = (\mathsf{sk}_i)_{i\in S}$.

Iteration 3 *to* ℓ+2. These iterations correspond to the ℓ adaptive queries $(m_j)_{j\in[\ell]}$ the adversary is allowed to make to (the slightly modified) $\mathsf{OPartDec}$. We detail it out for iteration 3, the other iterations work in an analogue manner. On output $\boldsymbol{\eta}_3 := m_1 \leftarrow \mathcal{A}((\mathsf{sk}_i)_{i\in S})$, we define $\Phi_3(\boldsymbol{\eta}_3)$ as follows: sample encryption randomness $\rho_1 \leftarrow \mathcal{X}$ and compute $\mathsf{ct}_1 := \mathsf{Enc}(m_1, \mathsf{pk}; \rho_1)$. Then, run $\mathsf{PartDec}$ on input ct_1 and the honest secret key share sk_{i^*} with $i^* \in [n] \setminus S$ to obtain p_1 of the honest party. By construction of the ThFHE scheme, it holds $\mathsf{p}_1 = \langle \mathsf{ct}_1, \mathsf{sk}\rangle - \sum_{i\in S}\langle \mathsf{ct}_1, \mathsf{sk}_i\rangle + e_1^{\mathsf{fl}}$, where $e_1^{\mathsf{fl}} \leftarrow \mathcal{D}_{\mathsf{flood}}$. Finally, output $\mathbf{r}_3 = \mathsf{Part}_3(\mathbf{r}_3) = (\rho_1, \mathsf{ct}_1, \mathsf{p}_1)$.

Iteration ℓ+3. On output $\boldsymbol{\eta}_{\ell+3} := (m^{(0)}, m^{(1)}) \leftarrow \mathcal{A}(\rho_\ell, \mathsf{ct}_\ell, \mathsf{p}_\ell)$, we define $\Phi_{\ell+3}(\boldsymbol{\eta}_{\ell+3}) = \Phi_{\ell+3}$ as the constant zero-function, hence $\mathbf{r}_{\ell+3} = \mathsf{Part}_{\ell+3}(\mathbf{r}_{\ell+3}) = \mathbf{0}$.

After $\tau := \ell + 3$ iterations, we have $\boldsymbol{\eta} = (S, (m_j)_{j\in[\ell]}, m^{(0)}, m^{(1)})$ (as the first output $\boldsymbol{\eta}_1$ is irrelevant to the game, we omit it). Further, the (partial) parameter $\mathbf{r}$ is given by $\mathbf{r} = (\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk}, (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)_{j\in[\ell]})$, and $\mathsf{Part}(\mathbf{r}) = (\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)_{j\in[\ell]})$.

Distribution $D_b(\boldsymbol{\eta}, \mathbf{r})$. Given $\boldsymbol{\eta}$ and $\mathbf{r}$, the distribution $D_b(\boldsymbol{\eta}, \mathbf{r})$ runs $\mathsf{ct}^{(b)} \leftarrow \mathsf{Enc}(m^{(b)}, \mathsf{pk})$ and outputs $x := \mathsf{ct}^{(b)}$.

Decision Problem P. We have now everything ready to define the decision problem P. Given input smpl, the PPT adversary $\mathcal{A}$ has to distinguish whether $\mathsf{smpl} \leftarrow X_0$ or $\mathsf{smpl} \leftarrow X_1$, where for $b \in \{0, 1\}$

$$X_b = \{\, \mathsf{smpl} = (\bar{\mathbf{r}}, x) \mid x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r}), \mathbf{r} \leftarrow \Phi(\boldsymbol{\eta}), \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}) \,\}\,.$$

Note that smpl equals the view of the adversary in Hybrid_1, as specified in Eq. 1.

Decision Problem P'. We now define the closely related decision problem P', where Φ is replaced by Φ'. For iteration 1, iteration 2 and iteration $\ell + 3$, Φ' equals Φ. Only at iteration 3 to $\ell + 2$, we slightly modify the distribution. For $j \in [\ell]$, it outputs $\mathsf{p}_j = m_j - \sum_{i \in S} \langle \mathsf{ct}_j, \mathsf{sk}_i \rangle + e_j^{\mathsf{fl}}$, where $e_j^{\mathsf{fl}} \leftarrow \mathcal{D}_{\mathsf{sim}}$ for some distribution $\mathcal{D}_{\mathsf{sim}}$. Note that this corresponds exactly to how $\mathsf{OSimPartDec}$ works and thus P' corresponds to the experiment defined by Hybrid_2.

Given input smpl, the PPT adversary $\mathcal{A}$ has to distinguish whether $\mathsf{smpl} \leftarrow X'_0$ or $\mathsf{smpl} \leftarrow X'_1$, where

$$X'_b = \{\, \mathsf{smpl} = (\bar{\mathbf{r}}, x) \mid x \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r}), \mathbf{r} \leftarrow \Phi'(\boldsymbol{\eta}), \bar{\mathbf{r}} := \mathsf{Part}(\mathbf{r}) \,\}\,.$$

Note that smpl equals the view of the adversary in Hybrid_2.

Argue Public Sampleability. Let $(\boldsymbol{\eta}, \mathbf{r}, b)$ be any fixed tuple and let $\mathsf{smpl} = (\bar{\mathbf{r}}, x)$, with $\bar{\mathbf{r}} = \mathsf{Part}(\mathbf{r}) = (\mathsf{pk}, (\mathsf{sk}_i)_{i \in S}, (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)_{j \in [\ell]})$ and $x = \mathsf{ct}^{(b)} \leftarrow D_b(\boldsymbol{\eta}, \mathbf{r})$. We can define the public sampler as follows:

- $\mathsf{Sample}(\boldsymbol{\eta}, 0, \mathsf{smpl})$: compute a fresh encryption of $m^{(0)}$ using pk and set $x' := \mathsf{ct}^{(0)} \leftarrow \mathsf{Enc}(m^{(0)}\mathsf{pk})$. Output $\mathsf{smpl}' = (\bar{\mathbf{r}}, x')$,
- $\mathsf{Sample}(\boldsymbol{\eta}, 1, \mathsf{smpl})$: compute a fresh encryption of $m^{(1)}$ using pk and set $x' := \mathsf{ct}^{(1)} \leftarrow \mathsf{Enc}(m^{(1)}, \mathsf{pk})$. Output $\mathsf{smpl}' = (\bar{\mathbf{r}}, x')$.

Note that for a fixed $\boldsymbol{\eta}$ and $\mathbf{r}$, the tuple $\bar{\mathbf{r}} = (\mathsf{pk}, (\mathsf{sk}_i)_{i \in S}, (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)_{j \in [\ell]})$ is uniquely defined, so there is no "fresh randomness" required for the public sampleability. In other words, the public key, corrupted secret key shares and the partial decryptions are independent of the challenge bit b, making public sampleability possible.

Loss in Reduction. Invoking Theorem 1 provides a reduction $P' \Rightarrow P$, where the loss is mainly defined by $\mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta}))$. It remains to bound the latter. We claim that

$$\mathrm{RD}_\alpha(\Phi(\boldsymbol{\eta}) \| \Phi'(\boldsymbol{\eta})) \leq \rho^\ell.$$

By slightly renaming the variables, we define $\Phi(\boldsymbol{\eta}) \to \mathbf{r} = (\mathbf{r}_0, \ldots, \mathbf{r}_\ell)$, where $\mathbf{r}_0 = (\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \in S_0$ and $\mathbf{r}_j = (\rho_j, \mathsf{ct}_j, \mathsf{p}_j)$ for $j \in [\ell]$. Similarly, we define $\Phi'(\boldsymbol{\eta}) \to \mathbf{r}' = (\mathbf{r}'_0, \ldots, \mathbf{r}'_\ell)$. By Lemma 3,

$$\mathrm{RD}_\alpha(\mathbf{r} \| \mathbf{r}') \leq \mathrm{RD}_\alpha(\mathbf{r}_0 \| \mathbf{r}'_0) \cdot \max_{y \in S_0} \mathrm{RD}_\alpha((\mathbf{r}_1, \ldots, \mathbf{r}_\ell \mid y) \| (\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell \mid y)).$$

As $\mathbf{r}_0$ and $\mathbf{r}'_0$ follow the same distribution, their Rényi divergence is 1.

Now, we observe that the only difference between $(\mathbf{r}_1, \dots, \mathbf{r}_\ell)$ and $(\mathbf{r}'_1, \dots, \mathbf{r}'_\ell)$ lies in how the honest partial decryption shares are sampled. When sampling through Φ, they can be written as

$$\mathsf{p}_j = \langle \mathsf{ct}_j, \mathsf{sk} \rangle - \sum_{i \in S} \langle \mathsf{ct}_j, \mathsf{sk}_i \rangle + e_j^{\mathrm{fl}} = m_j + e_j^{\mathrm{ct}} - \sum_{i \in S} \langle \mathsf{ct}_j, \mathsf{sk}_i \rangle + e_j^{\mathrm{fl}},$$

where $\left\| e_j^{\mathrm{ct}} \right\|_\infty \leq \beta$ and $e_j^{\mathrm{fl}} \sim \mathcal{D}_{\mathsf{flood}}$. When sampling through Φ', they can be written as

$$\mathsf{p}_j = m_j - \sum_{i \in S} \langle \mathsf{ct}_j, \mathsf{sk}_i \rangle + e_j^{\mathrm{fl}},$$

where $e_j^{\mathrm{fl}} \sim \mathcal{D}_{\mathsf{sim}}$. By applying Lemma 2, we obtain

$$\begin{aligned} \mathrm{RD}_\alpha(\mathbf{r} \| \mathbf{r}') &\leq \max_{y \in S_0} \mathrm{RD}_\alpha((e_1^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}}, \dots, e_\ell^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}} \mid y) \| ((\mathcal{D}_{\mathsf{sim}})^\ell \mid y)) \\ &\leq \mathrm{RD}_\alpha(e_1^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}}, \dots, e_\ell^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}} \| \mathcal{D}_{\mathsf{sim}}^\ell) \leq (\epsilon_{\mathrm{RD},\alpha})^\ell, \end{aligned}$$

where we used Lemma 4. ■

Lemma 7. *For any* PPT *adversary* $\mathcal{A}'$ *in* Hybrid_2 *and* Hybrid_3 *it holds*

$$\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThPKE}}(\mathcal{A}') = \mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThPKE}}(\mathcal{A}').$$

Proof. The view of $\mathcal{A}$ in Hybrid_3 is perfectly indistinguishable from their view in Hybrid_2 by the perfect privacy property of LSS. ■

□

5 Threshold FHE

In this section, we move to the more challenging fully-homomorphic threshold setting. Similarly to Section 4, we show that the simple construction from [BS23a] already fulfills a selective flavor of semantic security, without the need of the proof detour via one-way security.

5.1 Threshold FHE

We first define the notion of threshold fully-homomorphic encryption, which we denote by ThFHE.

Definition 13. *A threshold fully-homomorphic encryption scheme* ThFHE *over message space* $\mathcal{M}$*, set of admissible circuits* $\mathcal{C}$ *and set of* n *parties* $\{P_1, \dots, P_n\}$ *is defined by the following* PPT *algorithms* (Setup, Enc, Eval, PartDec, Combine)*:*

- $(\mathsf{pp}, \mathsf{sk}_1, \ldots \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$*: on input the security parameter* λ*, the number of parties* n *and a threshold value* $t \in \{2, \ldots, n\}$*, the setup algorithm outputs public parameters* pp*, the list of secret key shares* $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ *and the public key* pk*.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$*: on input a public key* pk *and a plaintext* m*, the encryption algorithm outputs a ciphertext* ct *encrypting* m *under* pk*.*
- $\hat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{ct}_1, \ldots, \mathsf{ct}_\kappa, C, \mathsf{pk})$*: on input a public key* pk*, a circuit circuit* $C \in \mathcal{C}$*, where* $C : \mathcal{M}^\kappa \to \mathcal{M}$*, and a list of ciphertexts* $\mathsf{ct}_1, \ldots, \mathsf{ct}_\kappa$*, the evaluation algorithm outputs a ciphertext* $\hat{\mathsf{ct}}$*.*
- $\mathsf{p}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathsf{sk}_i)$*: on input the secret key share* sk_i *corresponding to party* P_i *and a ciphertext* ct*, the partial decryption algorithm outputs a partial decryption* p_i *corresponding to party* P_i*.*
- $m \leftarrow \mathsf{Combine}((\mathsf{p}_i)_{i \in S})$*: on input a vector of partial decryptions* p_i*, for* $i \in S$*, where* $S \subset [n]$*, and a ciphertext* ct*, the combining algorithm outputs a plaintext* $m \in \mathcal{M}$ *or the symbol* $\perp$*.*

We note that in contrast to (for example) [BGG+18; BS23b], we move away from the notion of "depth" and instead use the more general notion of "admissible circuits". If one wanted to specialize to the leveled FHE schemes setting, one could pass a depth parameter to the Setup algorithm.

Definition 14 (Correctness of ThFHE). *Let* ThFHE *be a threshold fully homomorphic encryption scheme as in Definition 13. Let a circuit* $C \in \mathcal{C}$ *such that* $C : \mathcal{M}^\kappa \to \mathcal{M}$*. Let* $(m_1, \ldots, m_\kappa) \in \mathcal{M}^\kappa$*, let* $(\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, t, n)$*,* $\mathsf{ct}_j \leftarrow \mathsf{Enc}(m_j, \mathsf{pk})$ *for* $j \in \{1, \ldots, \kappa\}$*,* $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{ct}_1, \ldots, \mathsf{ct}_\kappa, C, \mathsf{pk})$ *and* $\mathsf{p}_i \leftarrow \mathsf{PartDec}(\mathsf{ct}, \mathsf{sk}_i)$ *for* $i \in S$*. Then, for all* $S \subseteq [n]$ *with* $|S| \geq t$*, we have that*

$$\Pr[\mathsf{Combine}((\mathsf{p}_i)_{i \in S}) = C(m_1, \ldots, m_\kappa)] = 1 - \mathsf{negl}(\lambda).$$

Indistinguishability. We define a selective variant of ℓ-$\mathsf{IND\text{-}CPA}$ security for ThFHE, as done in [CSS+22]. The selective nature refers to the fact that the adversary has to decide on their queries to the partial decryption oracle and the challenge messages *all at once*. We remark that our selective security notion is slightly stronger than the one defined in [BS23a]. Ours uses the same ciphertext vector for all circuit evaluations, whereas theirs uses fresh encryptions, even for the same messages. By bloating up κ to $\kappa\ell$, ours encompasses theirs.

Definition 15 (selective-ℓ-$\mathsf{IND\text{-}CPA}$ for ThFHE). *A* ThFHE *scheme is said to fulfill* selective-ℓ-$\mathsf{IND\text{-}CPA}$ *security for the security parameter* λ*, the threshold parameters* n, t *and query bound* ℓ*, if for all* PPT *adversaries* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$

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

where $\mathsf{Expt}^{\text{selective-}\ell\text{-IND-CPA}}_{\mathcal{A},\mathsf{ThFHE}}$ *is the experiment in Figure 5.*

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

$(\mathsf{pp}, \mathsf{sk}_1, \dots \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$
$S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| < t$
$(\vec{m}^{(0)}, \vec{m}^{(1)}, (C_j)_{j\in[\ell]}) \leftarrow \mathcal{A}_2((\mathsf{sk}_i)_{i\in S})$
$b \leftarrow \{0,1\}$
parse $\vec{m}^{(b)} := (m_1^{(b)}, \dots, m_\kappa^{(b)})$
for $k \in [\kappa]$:
  $\mathsf{ct}_k^\star \leftarrow \mathsf{Enc}(m_k^{(b)}, \mathsf{pk})$
$\vec{\mathsf{ct}}^\star := (\mathsf{ct}_k)_{k\in[\kappa]}$
for $j \in [\ell]$:
  if $C_j \notin \mathcal{C}$ **then return** $\perp$
  if $C_j(\vec{m}^{(0)}) \neq C_j(\vec{m}^{(1)})$ **then return** $\perp$
  $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Eval}(\vec{\mathsf{ct}}^\star, C_j, \mathsf{pk})$
  for $i \in [n] \setminus S$:
    $\mathsf{p}_{j,i} \leftarrow \mathsf{PartDec}(\widehat{\mathsf{ct}}_j, \mathsf{sk}_i)$
  $\vec{\mathsf{p}}_j := (\mathsf{p}_{j,i})_{i\in[n]\setminus S}$
$b' \leftarrow \mathcal{A}_3(\vec{\mathsf{ct}}^\star, (\widehat{\mathsf{ct}}_j, \vec{\mathsf{p}}_j)_{j\in[\ell]})$
return $b = b'$

Fig. 5. Experiment for selective-ℓ-IND-CPA security of ThFHE schemes.

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

$(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{FHE.Setup}(1^\lambda)$
$(\mathsf{sk}_1, \dots, \mathsf{sk}_n) \leftarrow \mathsf{LSS.Share}(\mathsf{sk})$
return $(\mathsf{pp}, \mathsf{sk}_1, \dots, \mathsf{sk}_n, \mathsf{pk})$

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

parse $\mathsf{sk}_i := (\mathsf{sk}_{i,1}, \dots, \mathsf{sk}_{i,L})$
for $j \in [L]$:
  $e_j^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{flood}}$
  $\mathsf{p}_j = \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle + e_j^{\mathrm{fl}} \bmod q$
return $\mathsf{p}_i = (\mathsf{p}_j)_{j\in[L]}$

$\mathsf{ThFHE.Combine}((\mathsf{p}_i)_{i\in S})$

$y = \mathsf{LSS.Rec}_S((\mathsf{p}_i)_{i\in S})$
return $\mathsf{FHE.Decode}(y)$

Fig. 6. Setup, partial decrypt and combine algorithms for the blueprint ThFHE construction in [BS23b]. The Enc and Eval algorithms are the same as for the underlying FHE.

5.2 Construction

We recall in Figure 6 the blueprint construction for threshold FHE from [BS23b], which uses as main ingredients a fully-homomorphic encryption scheme $\mathsf{FHE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}, \mathsf{Decode})$ with nearly linear decryption modulo q and a linear secret sharing scheme $\mathsf{LSS} = (\mathsf{Share}, \mathsf{Rec})$. We recall the formal definition of a secure fully-homomorphic encryption scheme with nearly linear decryption in Section 2.4. Informally, the definition corresponds to the one in the plain public-key setting, with the exception that the decrypted ciphertext may come from homomorphic evaluations. The construction works over the ring $\mathcal{R}_q = \mathbb{Z}_q[X]/f(X)$ for some degree-d irreducible polynomial f and some modulus q, and makes use of a noise distribution $\mathcal{D}_{\mathsf{flood}}$ over $\mathcal{R}_q$. Finally, we require the notion of Circuit Privacy (CP) in our construction, which we define below.

Definition 16 (Circuit Privacy). *Let* FHE *be as in Definition 6, with message space $\mathcal{M}$ and set of admissible circuits $\mathcal{C}$. Consider the two following distributions, denoted by D_0 and D_1, respectively:*

$$\big(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}, C, \vec{m}, \vec{\mathsf{ct}}, \mathsf{FHE.Eval}(\vec{\mathsf{ct}}, C, \mathsf{pk})\big)$$
$$\textit{and}$$
$$\big(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}, C, \vec{m}, \vec{\mathsf{ct}}, \mathsf{FHE.Enc}(C(\vec{m}), \mathsf{pk})\big),$$

where $C \in \mathcal{C}$ with $C : \mathcal{M}^\kappa \to \mathcal{M}$, $m_i \in \mathcal{M}$, $(\mathsf{pp}, \mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{FHE.Setup}(1^\lambda)$ and $\mathsf{ct}_i \leftarrow \mathsf{FHE.Enc}(m_i, \mathsf{pk})$ for all $i \in [\kappa]$, defining $\vec{m} = (m_i)_{i\in[\kappa]}$ and $\vec{\mathsf{ct}} = (\mathsf{ct}_i)_{i\in[\kappa]}$. We say that FHE *fulfills ϵ_{CP}-*circuit privacy *if for every* PPT *adversary $\mathcal{A}$, it yields that*

$$|\Pr[\mathcal{A}(x) \to 1 \mid x \leftarrow D_0] - \Pr[\mathcal{A}(x) \to 1 \mid x \leftarrow D_1]| \leq \epsilon_{\mathsf{CP}}.$$

5.3 Indistinguishability

This section is dedicated to directly proving the blueprint construction presented in Figure 6 to be selective-ℓ-IND-CPA secure. The proof makes use of the extended public sampleability framework from Section 3. For ease of exposition, we focus on the full (n-out-of-n) threshold case. It can be generalized to arbitrary thresholds $t \leq n$ in the same way as it has been done in [BS23b] and detailed in Appendix B.1 for the PKE case.

Theorem 3. *Set $t = n$. Let* LSS *denote additive secret sharing over $\mathcal{R}_q$ which fulfills privacy, linearity and reconstruction. Let* FHE *be an* IND-CPA *secure fully homomorphic encryption scheme satisfying (β, ϵ)-nearly linear decryption (Def. 7) and ϵ_{CP}-circuit privacy (Def. 16). Let $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$ be two distributions over $\mathcal{R}_q$, where $\mathrm{RD}_\alpha(\mathcal{D}_{\mathsf{flood}} + B\|\mathcal{D}_{\mathsf{sim}}) \leq \epsilon_{\mathrm{RD},\alpha}$ for some $\alpha \in (1, \infty)$ and $\epsilon_{\mathrm{RD},\alpha} > 1$ and for all $B \in \mathcal{R}_q$ with $\|B\|_\infty \leq \beta$. Then,* ThFHE *constructed as in Figure 6 is* selective-ℓ-IND-CPA *secure. More precisely, assume there exists an adversary $\mathcal{A}$ against* selective-ℓ-IND-CPA *of* ThFHE *with advantage* Adv*, then we can construct an adversary $\mathcal{A}'$ against* IND-CPA *of* FHE *with advantage* Adv$'$*, where*

$$\mathsf{Adv}' \geq \frac{(\mathsf{Adv} - \ell\epsilon - \ell\epsilon_{\mathsf{CP}})^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}} \cdot (\epsilon_{\mathrm{RD},\alpha})^\ell}.$$

Proof. Let $\mathcal{A}$ denote a PPT adversary against the selective-ℓ-IND-CPA security of ThFHE. As in Section 4, the overall strategy is to sequentially modify the selective-ℓ-IND-CPA game of ThFHE. This is done via a sequence of hybrid games, such that in the final one, the information that is exposed to the adversary is independent of the underlying secret key. We detail the hybrids below.

Hybrid_0 : This is the real selective-ℓ-IND-CPA security game as in Figure 5. The view of the adversary can be written as:

$$\mathsf{smpl} = \big(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, (C_j)_{j\in[\ell]}, \vec{m}^{(0)}, \vec{m}^{(1)}, \vec{\mathsf{ct}}^{\star}, (\widehat{\mathsf{ct}}_j, \mathsf{p}_j)_{j\in[\ell]}\big),$$

where $(\mathsf{p}_j)_{j\in[\ell]}$ are the partial decryptions of the (single) honest party. We have that $\mathsf{Adv}^{\text{selective-}\ell\text{-IND-CPA}}_{\mathsf{ThFHE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThFHE}}(\mathcal{A})$.

Hybrid_1 : In this hybrid, detailed in Figure 7, we modify the experiment as follows. Recall that in the previous hybrid, shown in Figure 5, we provide the adversary with the partial decryptions of homomorphically evaluated ciphertexts. Now, we directly encrypt the results of the circuit evaluations and then partially decrypt the corresponds ciphertexts. We emphasize the difference between Hybrid_0 and Hybrid_1 is in the way the elements $\widehat{\mathsf{ct}}_j$ are formed: in the former, we encrypt then evaluate and in the latter, we evaluate then encrypt. By assumption, the results of the circuit evaluations are independent of the challenge bit. Thus, partial decryptions can now be computed *before* the challenge bit is sampled.

Hybrid_2 : In this hybrid experiment, we slightly modify the partial decryptions as follows. On input m_j, for all $j \in [\ell]$, the reduction first checks whether $\langle \widehat{\mathsf{ct}}_j, \mathsf{sk}\rangle = m_j + e_j^{\mathrm{ct}}$, for some $e_j^{\mathrm{ct}} \in \mathcal{R}_q$ with $\left\|e_j^{\mathrm{ct}}\right\|_\infty \leq \beta$. If not, the game aborts. This hybrid is illustrated in Figure 7, where the two additional lines of Hybrid_2 are written in blue and marked with // Hybrid_2.

Hybrid_3 : In this hybrid experiment, we replace honest partial decryptions by simulated partial decryptions, as described below in Figure 8. Crucially, the latter does not use the honest secret key share.

Hybrid_4 : Now that the partial decryptions no longer depend on the honest secret key share, we can replace the honest setup by a simulated setup. Instead of providing key shares of the secret key sk to the adversary, we now instead provide dummy key shares of $\mathbf{0}$. In this hybrid, neither the corrupted key shares nor the partial decryptions depend on the secret key sk. It follows that this experiment corresponds exactly to the IND-CPA game for the underlying FHE scheme. Therefore we have that $\mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathsf{ThFHE}}(\mathcal{A}) = \mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{ThFHE}}(\mathcal{A})$.

The remaining of the proof is dedicated to quantifying the advantage of the adversary $\mathcal{A}$ between the hybrids.

$\mathsf{Hybrid}_1(1^\lambda, n, n)$ and $\mathsf{Hybrid}_2(1^\lambda, n, n)$

$(\mathsf{pp}, \mathsf{sk}_1, \ldots \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, n)$
$S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| = n - 1$
$\{ i^* \} = [n] \setminus S$
$(\vec{m}^{(0)}, \vec{m}^{(1)}, (C_j)_{j\in[\ell]}) \leftarrow \mathcal{A}_2((\mathsf{sk}_i)_{i\in S})$
for $j \in [\ell]$:
  if $C_j \notin \mathcal{C}$ **then return** $\perp$
  if $m_j := C_j(\vec{m}^{(0)}) \neq C_j(\vec{m}^{(1)})$ **then return** $\perp$
  $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Enc}(m_j, \mathsf{pk})$
  $e_j^{\mathrm{ct}} := \langle \widehat{\mathsf{ct}}_j, \mathsf{sk} \rangle - m_j$ // Hybrid_2
  if $\|e_j^{\mathrm{ct}}\|_\infty > \beta$ **then return** $\perp$ // Hybrid_2
  $\mathsf{p}_j \leftarrow \mathsf{PartDec}(\widehat{\mathsf{ct}}_j, \mathsf{sk}_{i^*})$
$b \leftarrow \{0, 1\}$
parse $\vec{m}^{(b)} := (m_1^{(b)}, \ldots, m_\kappa^{(b)})$
for $k \in [\kappa]$:
  $\mathsf{ct}_k^\star \leftarrow \mathsf{Enc}(m_k^{(b)}, \mathsf{pk})$
$\vec{\mathsf{ct}}^\star := (\mathsf{ct}_k^\star)_{k\in[\kappa]}$
$b' \leftarrow \mathcal{A}_3(\vec{\mathsf{ct}}^\star, (\widehat{\mathsf{ct}}_j, \mathsf{p}_j)_{j\in[\ell]})$
return $b = b'$

Fig. 7. Description of Hybrid_1 and Hybrid_2.

Lemma 8. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_0*, we can define an adversary for* Hybrid_1*, where*

$$\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThFHE}}(\mathcal{A}) \geq \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThFHE}}(\mathcal{A}) - \ell\epsilon_{\mathsf{CP}}.$$

Proof. Let i^* be the index of the single honest party. In Hybrid_0 (i.e. the original selective-ℓ-IND-CPA security game for bit $b \in \{0, 1\}$), the adversary's view can be described as

$$\big(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, (C_j)_{j\in[\ell]}, \vec{m}^{(0)}, \vec{m}^{(1)}, \vec{\mathsf{ct}}^\star, (\widehat{\mathsf{ct}}_j, \mathsf{p}_j)_{j\in[\ell]}\big),$$

where $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Eval}(\vec{\mathsf{ct}}^\star, C_j, \mathsf{pk})$ and $\vec{\mathsf{p}} = (\mathsf{p}_j)_{j\in[\ell]}$, with $\mathsf{p}_j \leftarrow \mathsf{PartDec}(\widehat{\mathsf{ct}}_j, \mathsf{sk}_{i^*})$.

In Hybrid_1 for bit $b \in \{0, 1\}$, the adversary's view can be described as

$$\big(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, (C_j)_{j\in[\ell]}, \vec{m}^{(0)}, \vec{m}^{(1)}, \vec{\mathsf{ct}}^\star, (\widehat{\mathsf{ct}}_j, \mathsf{p}_j)_{j\in[\ell]}\big),$$

where $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Enc}(C_j(m_j), \mathsf{pk})$, where we write $m_j = C_j(m^{(0)})$ $(= C_j(m^{(1)})$ by assumption), and $\vec{\mathsf{p}} = (\mathsf{p}_j)_{j\in[\ell]}$, with $\mathsf{p}_j \leftarrow \mathsf{PartDec}(\widehat{\mathsf{ct}}_j, \mathsf{sk}_{i^*})$.

Note that $\vec{\mathsf{ct}}^\star$ is the vector of challenge ciphertexts and depends on the bit b. Informally, the only difference between the original security game and the hybrid

$\mathsf{Hybrid}_3(1^\lambda, n, t)$

1 : $(\mathsf{pp}, \mathsf{sk}_1, \dots \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{Setup}(1^\lambda, n, t)$
2 : $S \leftarrow \mathcal{A}_1(\mathsf{pp}, \mathsf{pk})\colon S \subset [n] \wedge |S| < t$
3 : $(\vec{m}^{(0)}, \vec{m}^{(1)}, (C_j)_{j\in[\ell]}) \leftarrow \mathcal{A}_2((\mathsf{sk}_i)_{i\in S})$
4 : **for** $j \in [\ell]$:
5 : **if** $C_j \notin \mathcal{C}$ **then return** $\perp$
6 : **if** $m_j := C_j(\vec{m}^{(0)}) \neq C_j(\vec{m}^{(1)})$ **then return** $\perp$
7 : $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Enc}(m_j, \mathsf{pk})$
8 : $e_j^{\mathrm{ct}} := \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}\rangle - m_j$
9 : **if** $\left\|e_j^{\mathrm{ct}}\right\|_\infty > \beta$ **then return** $\perp$
10 : $e_j^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{sim}}$
11 : $\mathsf{p}_j \leftarrow m_j - \sum_{i\in S} \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}_i\rangle + e_j^{\mathrm{fl}} \pmod q$
12 : $b \leftarrow \{0,1\}$
13 : **parse** $\vec{m}^{(b)} := (m_1^{(b)}, \dots, m_\kappa^{(b)})$
14 : **for** $k \in [\kappa]$:
15 : $\mathsf{ct}_k^\star \leftarrow \mathsf{Enc}(m_k^{(b)}, \mathsf{pk})$
16 : $\vec{\mathsf{ct}}^\star := (\mathsf{ct}_k^\star)_{k\in[\kappa]}$
17 : $b' \leftarrow \mathcal{A}_3(\vec{\mathsf{ct}}^\star, (\widehat{\mathsf{ct}}_j, \mathsf{p}_j)_{j\in[\ell]})$
18 : **return** $b = b'$

Fig. 8. Description of Hybrid_3.

game lies in how the $\widehat{\mathsf{ct}}_j$ are computed. In the original game, they are obtained from the evaluation algorithm on the challenge ciphertexts and hence might depend on the bit b. In the hybrid game, they are directly obtained from the encryption algorithm on the final message m_j, which by assumption does not depend on the bit b.

By the construction of Figure 6, $\mathsf{p}_j = \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}_{i^*}\rangle + e_j^{\mathrm{fl}}$, for independently sampled flooding noise $e_j^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{flood}}$. Using the properties of additive secret sharing, we can express p_j as

$$\mathsf{p}_j = \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}\rangle - \sum_{i\in S} \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}_i\rangle + e_j^{\mathrm{fl}}.$$

By circuit privacy, for every $j \in [\ell]$, the tuple

$$\big(\mathsf{pp}, \mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, C_j, \vec{m}^{(0)}, \vec{m}^{(1)}, \vec{\mathsf{ct}}^\star, \widehat{\mathsf{ct}}_j, \langle \widehat{\mathsf{ct}}_j, \mathsf{sk}\rangle\big),$$

given in the original security game is computationally indistinguishable from the same tuple in the hybrid game. We conclude the proof by applying ℓ times the

same function on the tuple of both original and hybrid game. We can simply subtract $\sum_{i\in S}\langle\widehat{\mathsf{ct}}_j, \mathsf{sk}_i\rangle$ from $\langle\widehat{\mathsf{ct}}_j, \mathsf{sk}\rangle$ and add a fresh sample $e_j^{\mathsf{fl}} \leftarrow D$ to obtain p_j. By a union bound, it follows that Hybrid_1 is computationally indistinguishable from Hybrid_0 with loss $\epsilon_{\mathsf{CP}} \cdot \ell$. ■

Lemma 9. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_1*, we can define an adversary for* Hybrid_2*, where*

$$\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThFHE}}(\mathcal{A}) \geq \mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThFHE}}(\mathcal{A}) - \epsilon \cdot \ell.$$

Proof. The two hybrids only differ in case of an abort. By assumption, the underlying FHE scheme has (ϵ, β)-linear decryption, meaning that for all $j \in [\ell]$, $\Pr[\|e^{\mathrm{ct}}\|_\infty \geq \beta] \leq \epsilon$. It follows that

$$\Pr[\exists j \in [\ell] : \left\|e_j^{\mathrm{ct}}\right\|_\infty \geq \beta] \leq \sum_{j\in[\ell]} \Pr[\left\|e_j^{\mathrm{ct}}\right\| \geq \beta] \leq \epsilon \cdot \ell,$$

where we invoke the union bound. ■

Lemma 10. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_2*, we can define an adversary against* Hybrid_3*, where*

$$\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThFHE}}(\mathcal{A}) \geq \frac{(\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThFHE}}(\mathcal{A}))^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}}\epsilon^{\ell}_{\mathrm{RD},\alpha}}.$$

The proof is very similar to that of Lemma 6, and so we present it in Appendix B.2 for reasons of space.

Lemma 11. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_4 *and* Hybrid_3*, we have that*

$$\mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathsf{ThFHE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThFHE}}(\mathcal{A}).$$

Proof. Here, the proof follows exactly as in the PKE case. In more detail, it follows from the perfect privacy property of the LSS that the view of $\mathcal{A}$ in Hybrid_4 is perfectly indistinguishable from the view in Hybrid_3. ■

□

References

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE". In: *EUROCRYPT 2012*. Apr. 2012.

[ASY22] Shweta Agrawal, Damien Stehlé, and Anshu Yadav. "Round-Optimal Lattice-Based Threshold Signatures, Revisited". In: *ICALP 2022*. July 2022.

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

[BDG+19] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. "Leveraging Linear Decryption: Rate-1 Fully-Homomorphic Encryption and Time-Lock Puzzles". In: *TCC 2019, Part II*. Dec. 2019.

[BGG+18] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. "Threshold Cryptosystems from Threshold Fully Homomorphic Encryption". In: *CRYPTO 2018, Part I*. Aug. 2018.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ACM Transactions on Computation Theory (TOCT)* 3 (2014).

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

[BLR+18] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. "Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather than the Statistical Distance". In: *Journal of Cryptology* 2 (Apr. 2018).

[BPM+16] Florian Bourse, Rafaël del Pino, Michele Minelli, and Hoeteck Wee. "FHE Circuit Privacy Almost for Free". In: *CRYPTO 2016, Part II*. Aug. 2016.

[Bra12] Zvika Brakerski. "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP". In: *CRYPTO 2012*. Aug. 2012.

[BS23a] Katharina Boudgoust and Peter Scholl. *Simple Threshold (Fully Homomorphic) Encryption From LWE With Polynomial Modulus*. Cryptology ePrint Archive, Report 2023/016. 2023. URL: https://eprint.iacr.org/2023/016.

[BS23b] Katharina Boudgoust and Peter Scholl. "Simple Threshold (Fully Homomorphic) Encryption from LWE with Polynomial Modulus". In: *ASIACRYPT 2023, Part I*. Dec. 2023.

[CCH+24] Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. "On the Precision Loss in Approximate Homomorphic Encryption". In: *SAC 2023*. Aug. 2024.

[CGG+20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. "TFHE: Fast Fully Homomorphic Encryption Over the Torus". In: *Journal of Cryptology* 1 (Jan. 2020).

[CKK+17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. "Homomorphic Encryption for Arithmetic of Approximate Numbers". In: *ASIACRYPT 2017, Part I*. Dec. 2017.

[CSS+22] Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukher-

jee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. *Efficient Threshold FHE with Application to Real-Time Systems*. Cryptology ePrint Archive, Report 2022/1625. 2022. URL: https://eprint.iacr.org/2022/1625.

[DDK+23] Morten Dahl, Daniel Demmler, Sarah El Kazdadi, Arthur Meyre, Jean-Baptiste Orfila, Dragos Rotaru, Nigel P. Smart, Samuel Tap, and Michael Walter. "Noah's Ark: Efficient Threshold-FHE Using Noise Flooding". In: *WAHC@CCS*. 2023.

[DF90] Yvo Desmedt and Yair Frankel. "Threshold Cryptosystems". In: *CRYPTO'89*. Aug. 1990.

[DH76] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography". In: *IEEE Trans. Inf. Theory* 6 (1976).

[DLN+21] Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. "Non-interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings". In: *PKC 2021, Part I*. May 2021.

[DS16] Léo Ducas and Damien Stehlé. "Sanitization of FHE Ciphertexts". In: *EUROCRYPT 2016, Part I*. May 2016.

[DSS+21] Julien Devevey, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. "On the Integer Polynomial Learning with Errors Problem". In: *PKC 2021, Part I*. May 2021.

[FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. Cryptology ePrint Archive, Report 2012/144. 2012. URL: https://eprint.iacr.org/2012/144.

[HPR+20] James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. "Isochronous Gaussian Sampling: From Inception to Implementation". In: *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*. 2020.

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

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. "GGHLite: More Efficient Multilinear Maps from Ideal Lattices". In: *EUROCRYPT 2014*. May 2014.

[Mer78] Ralph C. Merkle. "Secure Communications Over Insecure Channels". In: *Commun. ACM* 4 (1978).

[MP24] Sean Murphy and Rachel Player. "A Central Limit Approach for Ring-LWE Noise Analysis". In: *CiC* 2 (2024).

[MS23] Daniele Micciancio and Adam Suhl. *Simulation-Secure Threshold PKE from LWE with Polynomial Modulus*. Cryptology ePrint Archive, Report 2023/1728. 2023. URL: https://eprint.iacr.org/2023/1728.

[OT25] Hiroki Okada and Tsuyoshi Takagi. *Low Communication Threshold FHE from Standard (Module-)LWE*. Cryptology ePrint Archive,

Paper 2025/409. 2025. URL: https://eprint.iacr.org/2025/409.

[PKP+24] Rafaël del Pino, Shuichi Katsumata, Thomas Prest, and Mélissa Rossi. "Raccoon: A Masking-Friendly Signature Proven in the Probing Model". In: *CRYPTO (1)*. 2024.

[Pre17] Thomas Prest. "Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence". In: *ASIACRYPT 2017, Part I*. Dec. 2017.

[PS24] Alain Passelègue and Damien Stehlé. *Low Communication Threshold Fully Homomorphic Encryption*. To appear in ASIACRYPT. 2024. URL: https://iacr.org/cryptodb/data/paper.php?pubkey=34664.

[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.

[SAB+22] Peter Schwabe et al. *CRYSTALS-KYBER*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. National Institute of Standards and Technology, 2022.

[VEH14] Tim Van Erven and Peter Harremos. "Rényi divergence and Kullback-Leibler divergence". In: *IEEE Transactions on Information Theory* 7 (2014).

Appendix A Additional Preliminaries

A.1 Linear Secret Sharing

We start by recalling notions of valid and invalid share sets [BGG+18].

Definition 17 ((In)Valid Share Sets). *Let $x \in \mathbb{Z}_q$, and $(\mathbf{v}_1, \dots, \mathbf{v}_n) \leftarrow \mathsf{Share}(x)$, where we write $\mathbf{v}_i = (\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,L})$ for $i \in [n]$. We call a set of pairs of indices $T \subseteq [n] \times [L]$ 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 call T a* valid set of share elements. *In particular, any valid set of share elements allows for reconstruction of the shared value x. Moreover,*

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

For a given LSS, we denote by τ_{max} the size of the smallest maximal invalid set of shares, while τ_{min} is the size of the largest minimal valid share sets.

We require a binary reconstruction. We use the same formulation as [BS23b], which is equivalent to the notion of a derived $\{0,1\}$-LSS used in [JRS17].

Definition 18 (Strong $\{0,1\}$-Reconstruction). *Let* LSS *be a linear secret sharing scheme. We say that* LSS *has* strong $\{0,1\}$-reconstruction *if for any secret x, any shares $(\mathbf{v}_1, \ldots, \mathbf{v}_n) \leftarrow \mathsf{Share}(x)$, where $\mathbf{v}_i = (\mathbf{v}_{i,1}, \ldots, \mathbf{v}_{i,L})$ for $i \in [n]$, and any valid set of shares $T \subseteq [n] \times [L]$, there exists a subset $T' \subseteq T$ such that $\sum_{(i,j)\in T'} \mathbf{v}_{i,j} = x$.*

Some common examples of t-out-of-n linear secret sharing schemes with strong $\{0,1\}$-reconstruction were provided by [BS23b, Table 1].

Appendix B Additional Proofs

B.1 Proof For Arbitrary Thresholds

We now prove a generalization of Theorem 2 to arbitrary thresholds.

Theorem 4. *Let* LSS *denote a t-out-of-n secret sharing over $\mathcal{R}_q$ which fulfills privacy, linearity and strong $\{0,1\}$-reconstruction, with associated parameters τ_{max} and L. Let* PKE *be an* IND-CPA *secure public-key encryption scheme with (β, ϵ)-nearly linear decryption. Let $\mathcal{D}_{\mathsf{flood}}$ and $\mathcal{D}_{\mathsf{sim}}$ be two distributions over $\mathcal{R}_q$, where $\mathrm{RD}_\alpha(\mathcal{D}_{\mathsf{flood}} + B \| \mathcal{D}_{\mathsf{sim}}) \leq \epsilon_{\mathrm{RD},\alpha}$ for some $\alpha \in (1, \infty)$ and $\epsilon_{\mathrm{RD},\alpha} > 1$ and for all $B \in \mathcal{R}_q$ with $\|B\|_\infty \leq \beta$. Then, the* ThPKE *constructed as in Figure 3 is ℓ-*IND-CPA *secure. More precisely, assume there exists an adversary $\mathcal{A}$ against ℓ-*IND-CPA *of* ThPKE *with advantage* Adv*, then we can construct an adversary $\mathcal{A}'$ against* IND-CPA *of* PKE *with advantage* Adv$'$*, where*

$$\mathsf{Adv}' \geq \frac{(\mathsf{Adv} - \ell\epsilon)^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}} \cdot (\epsilon_{\mathrm{RD},\alpha})^{\ell(nL-\tau_{\mathsf{max}})}}.$$

Recover Theorem 2 by observing that additive secret sharing fulfills strong $\{0,1\}$-reconstruction with $L = 1$ and $\tau_{\mathsf{max}} = n - 1$.

Proof. Let $\mathcal{A}$ denote a PPT adversary against ℓ-IND-CPA security of ThPKE. As for the full-threshold case, we modify the original ℓ-IND-CPA game in a sequence of hybrid games.

Hybrid_0 : This is the real ℓ-IND-CPA security experiment as detailed in Figure 2. The view of the adversary is given by (possibly dependent)

$$\mathsf{smpl} = \left(\mathsf{pp}, \mathsf{pk}, \{\,\mathsf{sk}_i\,\}_{i\in S}, (\rho_j, \mathsf{ct}_j, \vec{\mathsf{p}}_j)_{j\in[\ell]}, \mathsf{ct}^{(b)}\right). \tag{2}$$

It holds $\mathsf{Adv}^{\ell\text{-IND-CPA}}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThPKE}}(\mathcal{A})$.

Hybrid_1 : In the first hybrid experiment, we slightly modify the partial decryption oracle OPartDec. On input m, before outputting the tuple $(\rho, \mathsf{ct}, \vec{\mathsf{p}})$, it checks whether $\langle \mathsf{ct}, \mathsf{sk} \rangle = m + e^{\mathrm{ct}}$ for some $e^{\mathrm{ct}} \in \mathcal{R}_q$ with $\|e^{\mathrm{ct}}\|_\infty \leq \beta$. If not, the game aborts.

Hybrid_2 : In this hybrid game, we continue modifying the partial decryption oracle. Given the adversary's choice of corrupted parties $S \subset [n]$, let $S_L = (i,j)_{(i,j)\in S\times[L]}$ denote the corresponding share sets. As $|S| < t$, we know that S_L is an invalid set of share elements. Fix $T \supseteq S_L$ to be a maximal invalid set of share elements (as of Definition 17). For $i \in [n]$, every secret key share sk_i can be parsed as $\mathsf{sk}_i = (\mathsf{sk}_{i,1}, \ldots, \mathsf{sk}_{i,L})$. Then, we compute the partial decryption of the honest parties as follows:

1. For $(i,j) \in T \setminus S_L$, set $\widetilde{\mathsf{p}}_{i,j} = \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle$; then sample $e^{\mathsf{fl}}_{i,j} \leftarrow \mathcal{D}_{\mathsf{flood}}$ and define $\mathsf{p}_{i,j} = \widetilde{\mathsf{p}}_{i,j} + e^{\mathsf{fl}}_{i,j}$;
2. For $(i,j) \in ([n] \times [L]) \setminus T$, let $T_{i,j} \subset T \cup \{(i,j)\}$ be a minimal valid set of share elements, and compute $\widetilde{\mathsf{p}}_{i,j} = \langle \mathsf{ct}, \mathsf{sk} \rangle - \sum_{(k,l)\in T_{i,j}\setminus\{(i,j)\}} \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle$; then sample $e^{\mathsf{fl}}_{i,j} \leftarrow \mathcal{D}_{\mathsf{flood}}$ and define $\mathsf{p}_{i,j} = \widetilde{\mathsf{p}}_{i,j} + e^{\mathsf{fl}}_{i,j}$;
3. Define $\vec{\mathsf{p}} = (\mathsf{p}_{i,j})_{i\in[n]\setminus S, j\in[L]}$.

Hybrid_3 : In the next hybrid experiment, we again modify how partial decryption shares are computed. Item 2. of the above is replaced by the following:

2. For $(i,j) \in ([n] \times [L]) \setminus T$, let $T_{i,j} \subset T \cup \{(i,j)\}$ be a minimal valid set of share elements, and compute $\widetilde{\mathsf{p}}_{i,j} = m - \sum_{(k,l)\in T_{i,j}\setminus\{(i,j)\}} \langle \mathsf{ct}, \mathsf{sk}_{i,j} \rangle$; then sample $e^{\mathsf{fl}}_{i,j} \leftarrow \mathcal{D}_{\mathsf{sim}}$ and define $\mathsf{p}_{i,j} = \widetilde{\mathsf{p}}_{i,j} + e^{\mathsf{fl}}_{i,j}$;

The important change is that the partial decryption shares $\mathsf{p}_{i,j}$ of the honest parties is computed only using knowledge the adversary will have at the end of the game: namely message m, ciphertext ct and corrupted key shares $(\mathsf{sk}_{i,j})_{(i,j)\in T}$, where T is an invalid share set. Most importantly, it is not depending on the remaining honest key shares $(\mathsf{sk}_{i,j})_{(i,j)\in([n]\times[L])\setminus T}$. Moreover, the flooding distribution $\mathcal{D}_{\mathsf{flood}}$ is replaced a possibly different distribution $\mathcal{D}_{\mathsf{sim}}$.

Hybrid_4 : Once partial decryption queries only depend on secret key shares of an invalid set, and hence do not leak any information on sk, we can replace the honest setup by a simulated setup. Instead of providing key shares of sk to the adversary, now dummy key shares of $\mathbf{0}$ are provided to them, that is $(\mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{LSS.Share}(\mathbf{0})$. As in this hybrid neither the corrupted key shares nor the partial decryption queries depend on the actual secret key sk, the experiment corresponds to the IND-CPA security game of the underlying PKE. It holds $\mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A})$.

The step from Hybrid_0 to Hybrid_1 is exactly the same as in the full-threshold case, hence by Lemma 5,

$$\mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThPKE}}(\mathcal{A}) \geq \mathsf{Adv}^{\mathsf{Hybrid}_0}_{\mathsf{ThPKE}}(\mathcal{A}) - \epsilon\ell.$$

Lemma 12. *Any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_1 *is already defining an adversary for* Hybrid_2*, where*

$$\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_1}_{\mathsf{ThPKE}}(\mathcal{A}).$$

Proof. We observe that the view of $\mathcal{A}$ in Hybrid_1 is identical to their view in Hybrid_2, assuming the strong $\{0,1\}$-reconstruction of LSS. In more words, in Hybrid_2, every partial decryption share corresponding to $T \setminus S_L$ is computed exactly as in Hybrid_1, whereas the honest shares outside this set are deterministically computed to yield a sharing of $\langle \mathsf{ct}, \mathsf{sk} \rangle$ plus noise sampled from $\mathcal{D}_{\mathsf{flood}}$. ■

Lemma 13. *For any* PPT *adversary* $\mathcal{A}$ *for* Hybrid_2*, we can construct an adversary* $\mathcal{A}'$ *for* Hybrid_3 *such that it holds*

$$\mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThPKE}}(\mathcal{A}') \geq \frac{\left(\mathsf{Adv}^{\mathsf{Hybrid}_2}_{\mathsf{ThPKE}}(\mathcal{A})\right)^{\frac{2\alpha-1}{\alpha-1}}}{2^{\frac{4\alpha-3}{\alpha-1}} \cdot \epsilon_{\mathrm{RD},\alpha}^{\ell(nL-\tau_{\mathsf{max}})}}.$$

Proof. The proof essentially is the same as in the full-threshold case, Lemma 6. The two hybrids only differ on how the $nL - |T|$ partial decryption shares of item 2. are computed. We apply the RD argument on each of the shares. As $|T| \geq \tau_{\mathsf{max}}$, we can upper bound the Rényi divergence by replacing $nL - |T|$ by $nL - \tau_{\mathsf{max}}$. ■

Finally, the step from Hybrid_3 to Hybrid_4 is exactly the same as in the full-threshold case, hence by Lemma 7,

$$\mathsf{Adv}^{\mathsf{Hybrid}_4}_{\mathsf{ThPKE}}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{Hybrid}_3}_{\mathsf{ThPKE}}(\mathcal{A}).$$

□

B.2 Proof of Lemma 10

Proof. We appeal to Theorem 1 to prove this Lemma. More specifically, we formulate Hybrid_2 (respectively Hybrid_3) as a decision problem P (respectively P') of the form $D_b(\boldsymbol{\eta}, \mathbf{r})$, where $\boldsymbol{\eta}$ is adversarially chosen, and both $\boldsymbol{\eta}$ and Φ are adaptively chosen, so we detail them on each iteration. we have that $\mathbf{r} \leftarrow \Phi$ (respectively $\mathbf{r} \leftarrow \Phi'$). λ, n and t are given as global parameters.

We begin by detailing out the decision problem P, with adaptive adversary input as in Definition 8.

Iteration 1: On any input $\boldsymbol{\eta}_1 \leftarrow \mathcal{A}$ from the adversary, we define $\Phi_1(\boldsymbol{\eta}_1)$ as follows. Run $(\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathsf{ThFHE.Setup}(1^\lambda, n, t)$, which defines $\mathsf{sk} = \mathsf{Rec}(\{\, \mathsf{sk}_i \,\}_{i \in [n]})$. Set $\mathbf{r}_1 := (\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk})$ and $\mathsf{Part}(\mathbf{r}_1) := (\mathsf{pp}, \mathsf{pk})$. Note that Φ_1 is independent of $\boldsymbol{\eta}_1$, so we omit it in what follows. Note further that pp is included in both $\mathbf{r}_1$ and $\mathsf{Part}(\mathbf{r}_1)$ and we omit it in the rest of the section for ease of notation. Finally, at each iteration i, we assume that both $\boldsymbol{\eta}_i$ and Φ_i include information from $\boldsymbol{\eta}_{i-1}$, Φ_{i-1}, and omit writing it for ease of notation.

Iteration 2: On output $\boldsymbol{\eta}_2 := S \leftarrow \mathcal{A}(\mathsf{Part}(\mathbf{r}_1) = \mathsf{pk})$, we define $\Phi_2(\boldsymbol{\eta}_2)$ as follows. Set $\mathbf{r}_2 := \mathbf{r}_1$, and set $\mathsf{Part}(\mathbf{r}_2) := (\mathsf{sk}_i)_{i \in S}$.

Iteration 3: On output $\boldsymbol{\eta}_3 := (\vec{m}^{(0)}, \vec{m}^{(1)}, (C_j)_{j\in[\ell]})$, we define $\Phi_3(\boldsymbol{\eta}_3)$ as follows. Let $m_j := C_j(\vec{m}^{(0)}) = C_j(\vec{m}^{(1)})$, for $j \in [\ell]$. Let $\hat{\mathsf{ct}}_j \leftarrow \mathsf{ThFHE.Enc}(m_j, \mathsf{pk})$, for $j \in [\ell]$. Let $\mathsf{p}_j \leftarrow \mathsf{ThFHE.PartDec}(\hat{\mathsf{ct}}_j, \mathsf{sk}_1)$, for $j \in [\ell]$. Write $\vec{\hat{\mathsf{ct}}} := (\hat{\mathsf{ct}}_j)_{j\in[\ell]}$ and $\vec{\mathsf{p}} := (\mathsf{p}_j)_{j\in[\ell]}$. Set $\mathbf{r}_3 = (\vec{\hat{\mathsf{ct}}}, \vec{\mathsf{p}})$, and $\mathsf{Part}(\mathbf{r}_3) := \mathbf{r}_3$.

At the end of the three iterations, we have $\boldsymbol{\eta} := (\boldsymbol{\eta}_2, \boldsymbol{\eta}_3) = (S, \vec{m}^{(0)}, \vec{m}^{(1)}, (C_j)_{j\in[\ell]})$ (recall that $\boldsymbol{\eta}_1$ is irrelevant to the game, so we omit it). We also have $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) = (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \vec{\hat{\mathsf{ct}}}, \vec{\mathsf{p}})$ as well as $\mathsf{Part}(\mathbf{r}) = \big(\mathsf{Part}(\mathbf{r}_1), \mathsf{Part}(\mathbf{r}_2), \mathsf{Part}(\mathbf{r}_3)\big) = \big(\mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, \vec{\hat{\mathsf{ct}}}, \vec{\mathsf{p}}\big)$.

Decision problem P: We can now define the decision problem P. Given an input smpl, a PPT adversary $\mathcal{A}$ has to distinguish whether $\mathsf{smpl} \leftarrow X_0$ or $\mathsf{smpl} \leftarrow X_1$, where:

$$X_b := \{\mathsf{smpl} = (\mathsf{Part}(\mathbf{r}), x) \mid x \leftarrow D_b(\mathbf{r}), \mathbf{r} \leftarrow \Phi\}.$$

Note this corresponds exactly to Hybrid_2.

The distribution Φ' We define a closely related decision problem P', defined by the distribution Φ'. On Iterations 1 and 2, the distribution Φ' equals the distribution Φ. On Iteration 3, we now simulate the partial decryptions $\mathsf{p}_j \leftarrow \mathsf{OSimPartDec}(m_j)$, as described in Figure 8. In other words, in Φ we compute the partial decryptions honestly using the secret key share of the honest party sk_1, whereas in Φ', we compute the partial decryptions using the shares of the corrupted parties $(\mathsf{sk}_i)_{i\in S}$. Everything else proceeds exactly as in Φ.

Distribution $D_b(\boldsymbol{\eta}, \mathbf{r})$: Given $\boldsymbol{\eta}$ and $\mathbf{r}$, the distribution $D_b(\boldsymbol{\eta}, \mathbf{r})$ runs $\mathsf{ct}^\star \leftarrow \mathsf{ThFHE.Enc}(\vec{m}^{(b)}, \mathsf{pk})$ and outputs $x := \mathsf{ct}^\star$.

Decision problem P' : We now define the decision problem P'. Given an input smpl, a PPT adversary $\mathcal{A}$ has to distinguish whether $\mathsf{smpl} \leftarrow X'_0$ or $\mathsf{smpl} \leftarrow X'_1$, where:

$$X_b := \{\mathsf{smpl} = (\mathsf{Part}(\mathbf{r}), x) \mid x \leftarrow D_b(\mathbf{r}), \mathbf{r} \leftarrow \Phi'\}.$$

Arguing public sampleability Let $(\boldsymbol{\eta}, \mathbf{r}, b) \in T \times S \times \{0, 1\}$, let $\mathsf{smpl} = (\mathsf{Part}(\mathbf{r}), x)$, where $\mathsf{Part}(\mathbf{r}) = \big(\mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, \vec{\hat{\mathsf{ct}}}, \vec{\mathsf{p}}\big)$ and $\mathsf{ct}^\star \leftarrow D_b(\boldsymbol{\eta})$. We define a public sampler as follows.

- $\mathsf{Sample}(\boldsymbol{\eta}, 0, \mathsf{smpl})$: compute a fresh encryption of $\vec{m}^{(0)}$ using pk, set $x^{\mathsf{fresh}} := \mathsf{ct}^{(0)} \leftarrow \mathsf{ThFHE.Enc}(\vec{m}^{(0)}, \mathsf{pk})$.
- $\mathsf{Sample}(\boldsymbol{\eta}, 1, \mathsf{smpl})$: compute a fresh encryption of $\vec{m}^{(1)}$ using pk, set $x^{\mathsf{fresh}} := \mathsf{ct}^{(1)} \leftarrow \mathsf{ThFHE.Enc}(\vec{m}^{(1)}, \mathsf{pk})$.

Note that for fixed $\boldsymbol{\eta}$ and $\mathbf{r}$, $\mathsf{Part}(\mathbf{r}) = \big(\mathsf{pk}, (\mathsf{sk}_i)_{i\in S}, \vec{\hat{\mathsf{ct}}}, \vec{\mathsf{p}}\big)$ is uniquely defined and, as such, there is no "fresh randomness" required for public sampleability. In

particular, Part is independent of the challenge bit b, meaning that public sampleability is possible. We can now invoke Theorem 1, which gives us a reduction $P' \implies P$. It only remains to quantify the loss in the reduction.

For ease of exposition, we rename the variables as follows. Write $\mathbf{r} \leftarrow \Phi(\boldsymbol{\eta})$, where $\mathbf{r} := (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_\ell)$ with support S_0. Then, we let $\mathbf{r}_0 := (\mathsf{pp}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{pk})$. Then, for each $j \in [\ell]$, we write $\mathbf{r}_j := (\hat{\mathsf{ct}}_j, \mathsf{p}_j)$. We do the same for $\mathbf{r}' \leftarrow \Phi(\boldsymbol{\eta}')$. Note that $\mathbf{r}_0 = \mathbf{r}'_0$. We invoke Lemma 3 and write

$$\mathrm{RD}_\alpha(\mathbf{r}\|\mathbf{r}') \leq \mathrm{RD}_\alpha(\mathbf{r}_0\|\mathbf{r}'_0) \cdot \max_{y \in S_0} \mathrm{RD}_\alpha((\mathbf{r}_1, \ldots, \mathbf{r}_\ell \mid y)\|(\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell \mid y)).$$

Since we have that $\mathbf{r}_0 = \mathbf{r}'_0$, it remains to bound the second term of the above product.

We notice that the difference between $(\mathbf{r}_1, \ldots, \mathbf{r}_\ell)$ and $(\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell)$ lies in how the partial decryptions are performed. Recall that in the former, since we have that $\mathsf{p}_j = \langle \hat{\mathsf{ct}}_j, \mathsf{sk}_1 \rangle$, $\mathsf{sk} = \sum_{i \in [n]} \mathsf{sk}_i = \mathsf{sk}_1 + \sum_{i \in S} \mathsf{sk}_i$ and $\langle \hat{\mathsf{ct}}_j, \mathsf{sk} \rangle = e^{\mathrm{ct}} + m_j$, we can write the partial decryptions as

$$\mathsf{p}_j = e^{\mathrm{ct}} + m_j - \langle \hat{\mathsf{ct}}_j, \sum_{i \in S} \mathsf{sk}_i \rangle + e_j^{\mathrm{fl}}.$$

In the above, we additionally have that $\|e^{\mathrm{ct}}\|_\infty \leq \beta$ and that $e^{\mathrm{ct}} \leftarrow \mathcal{D}_{\mathsf{flood}}$.

In $(\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell)$, we instead have that

$$\mathsf{p}_j = m_j - \langle \hat{\mathsf{ct}}_j, \sum_{i \in S} \mathsf{sk}_i \rangle + e_j^{\mathrm{fl}},$$

where $e_j^{\mathrm{fl}} \leftarrow \mathcal{D}_{\mathsf{sim}}$. We can now apply Lemma 2 and Lemma 4 and finalise the proof.

$$\begin{aligned}
\mathrm{RD}_\alpha(\mathbf{r}\|\mathbf{r}') &\leq \max_{y \in S_0} \mathrm{RD}_\alpha((\mathbf{r}_1, \ldots, \mathbf{r}_\ell \mid y)\|(\mathbf{r}'_1, \ldots, \mathbf{r}'_\ell \mid y)) \\
&\leq \mathrm{RD}_\alpha(e_1^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}}, \ldots, e_\ell^{\mathrm{ct}} + \mathcal{D}_{\mathsf{flood}}\|\mathcal{D}_{\mathsf{sim}}^\ell) \\
&\leq \epsilon_{\mathrm{RD},\alpha}^\ell.
\end{aligned}$$

■