Threshold Traitor Tracing Revisited: Insider Attacks and Multi-Traitor Tracing*

Jan Bormet Sebastian Faust Hussien Othman

Technische Universität Darmstadt, Darmstadt, Germany
{jan.bormet, sebastian.faust}@tu-darmstadt.de, hussien.othman@gmail.com

Abstract

Threshold traitor tracing [Boneh et al., Crypto'24] enhances threshold decryption with anti-collusion guarantees via a tracing mechanism. In this model, a set of colluders constructs a decryption box, called a decoder, and sells it to an external buyer. Collusion-resilience is guaranteed by a tracing algorithm that, given only black-box access to the decoder, can identify at least one of the colluders.

Since its introduction, threshold traitor tracing has attracted significant attention, leading to a number of new constructions that have been proven secure within this model [Asiacrypt'25, CiC'25, CCS'25, Eurocrypt'26], focusing on optimizing parameter sizes and removing trusted authority. However, in this paper, we argue that the state-of-the-art definition and constructions are not resilient against natural attacks where the decoder is sold to an insider, i.e., a member of the decryption committee.

Motivated by this gap, we introduce new definitions to model decoder boxes that are sold to insiders. We show that existing threshold traitor tracing techniques are inherently vulnerable to insider attacks. Then, we introduce a novel approach to achieve insider resilience through *multi-traitor tracing*, i.e., identifying multiple traitors. We present compilers that amplify the number of traitors that can be found, thereby achieving insider-resilience with efficient parameters, in particular, sublinear ciphertext size. These compilers may also be of independent interest.

*This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 and Horizon Europe research and innovation programs (grant CRYPTOLAYER-101044770).

Contents

1 Introduction **4**

2 Technical Overview **7**
2.1 The Starting Point of This Paper . . . 7
2.2 Defining Traceability Against Insiders . . . 8
2.2.1 Insider Resilience from Multi-Traitor Tracing . . . 9
2.3 Compiler from τ-Traceability to $(K \cdot \tau)$-Traceability . . . 10
2.3.1 Robustness via Multi-Instance Expansion . . . 11
2.4 Tracing $\Theta(t)$ Traitors . . . 13
2.5 Compiling from 1-traceability to $\Theta(n^a)$-traceability for $a \in (0,1)$ . . . 13
2.6 A Construction with Maximal Partial Insider Resilience . . . 15
2.7 Additional Related Work . . . 16
2.8 Roadmap . . . 17

3 Preliminaries **17**
3.1 Partitions of Parties . . . 18
3.2 (Traceable) Threshold KEMs . . . 18

4 Selling Decoders to Insiders – a Stronger Definition **20**
4.1 The New Traceability Definition . . . 20
4.2 Remarks on Giving the Tracer Access to Insiders' Secret Keys . . . 23
4.3 Full Insider Tracing without Involving Secret Keys while Tracing is Impossible . . . 23
4.3.1 Implications of Theorem 4.6 on the Traceability Definition . . . 26

5 Insider Resilience from Multi-Traitor Tracing **26**
5.1 Multi-Traitor to Full Insiders . . . 27
5.2 Full-Insider Implies Partial-Insider Resilience . . . 29

6 Achieving K-traceability for Constant K **30**
6.1 The 1-Parition Compiler . . . 30
6.2 Multi-Instance Expansion . . . 31

7 On $\Theta(t)$-Traceability **33**

8 On $\Theta(n^a)$-Traceability for $a \in (0,1)$ **34**
8.1 The Tracing Algorithm . . . 37
8.2 A Proof of Lemma 8.6 . . . 40

9 Achieving Maximal Partial Insider Resilience **46**
9.1 Bipartite Threshold Encryption with Simulated Partial Decryptions . . . 47
9.2 A PD-BT-KEM from Pairings . . . 49
9.3 From PD-BT-KEM to Traceability with Maximal Partial Insider Resilience . . . 51
9.4 Traceability Proof for Construction 8 . . . 53

10 Generic Threshold Traitor Tracing and Gap Elimination Compiler **60**
10.1 A Proof of Theorem 10.1 . . . 62

A Relevance of Insider Resilience in the Blockchain Setting **69**

B Proofs of Partition Lemmas **70**

C Limit Elimination Compiler **75**

D Deferred Proofs for our Compilers **77**

E Naive t-Traceable Construction with Large Ciphertext **92**

F A Proof of Theorem 9.6 **94**

G Deferred Proof of Assumption 1 **98**

H Full-Insider Attacks in Existing Schemes **101**

1 Introduction

Threshold encryption [29, 30] enables secure encryption among a set of n parties by generating a succinct ciphertext decryptable by any subset of at least $t \leq n$ parties, while remaining private against coalitions of up to $t-1$ malicious parties. Motivated by applications such as encrypted mempools [2, 33], recent works have explored numerous variants of threshold encryption [39, 53, 45, 26, 19, 17, 1, 8, 7]. However, all existing constructions remain vulnerable to collusion attacks, in which committee members cooperate to help the adversary decrypt ciphertexts. Motivated by this limitation, Boneh, Partap, and Rotem [11] recently introduced *threshold traitor tracing*, extending threshold encryption with a *traceability* guarantee. Informally, a coalition of t or more parties may embed their secret keys into a *decoder* capable of decrypting ciphertexts. Given only black-box access to such a decoder, a tracing algorithm must identify at least one colluder. Subsequent works studied threshold traitor tracing under various tradeoffs, including constructions without a trusted dealer or interaction [20, 23], CCA security [3], public traceability [22], and polylogarithmic parameters [28].

The existing definition of threshold traitor tracing [11] models collusion via a *standalone* decoder, namely one that decrypts correctly with noticeable probability ε. Such a decoder can be used by anyone to decrypt arbitrary ciphertexts, capturing the setting where colluders sell a decoder to an external buyer, who may then submit it for tracing. Collusion deterrence arises from the risk that such a buyer acts as a whistleblower (see, e.g., [47]). That is, the buyer may blow the whistle by forwarding the decoder to the tracer, who can then trace at least one colluder and punish them, thus disincentivizing collusion. However, we identify a fundamental limitation of this model when the buyer of the decoder is an *insider*, i.e., a party that can obtain additional non-trivial information useful for decryption, such as valid secret-key shares or partial decryptions. In the simplest case, an insider can be thought of as a committee member. Then, colluders may construct a decoder D that decrypts correctly only when provided, say, with an insider's secret key share sk_i. While such a decoder can be used by the insider to decrypt, tracing it necessarily requires supplying this insider information during tracing (otherwise, the decoder could simply refuse to decrypt). Thus, the colluders may design the decoder so that tracing *always accuses the insider*. In Section 2.1, we show that this vulnerability is inherent to a widely used design pattern underlying several existing constructions [11, 3, 23, 22]. As a result, insiders have little incentive to submit such decoders for tracing, since tracing only implicates themselves.

Insider attacks fundamentally separate threshold traitor tracing from the non-threshold (broadcast) setting [25]. In non-threshold traitor tracing, insider attacks are irrelevant: an "insider" already possesses full decryption capability and therefore does not need a decoder to decrypt. In contrast, under threshold encryption, committee members cannot independently decrypt arbitrary ciphertexts and may therefore purchase a decoder to obtain such capability.

This motivates us to study threshold traitor tracing in the presence of insiders. We consider two types of insiders: *full insiders*, who input their entire secret-key share to the decoder, and *partial insiders*, who provide only partial decryptions. While we consider the notion of partial insiders to be a relaxation of full insiders, it captures realistic scenarios on its own. First, users may be unwilling to enter their entire secret key into untrusted software and thus prefer to enter only a partial decryption. Second, this models a setting where some partial decryptions for ciphertexts that are supposed to remain private are exposed during normal protocol execution [7, 37], e.g., in encrypted mempools. We provide a more detailed blockchain-focused discussion in Appendix A.

Related Models. To the best of our knowledge, traceability in the presence of insiders for threshold traitor tracing has not been studied previously. Existing works address related, but different settings.

Threshold Traitor Tracing with Small Coalitions. In [11], the authors study an additional model in which decoders require to be fed with partial decryptions. This model differs from ours in three significant respects. First, in their setting, such decoders are constructed by small

coalitions consisting of fewer than t traitors. In contrast, we consider decoders that may be constructed by coalitions of arbitrary size, including coalitions larger than the threshold t. This distinction is crucial: the techniques of [11] rely on the assumption that the decoder cannot decrypt without the additional partial decryptions, where in our setting the decoder may already be capable of decrypting, and the additional partial decryptions are used instead to evade tracing. Second, their decoders are required to function for *any* set of partial decryptions. By contrast, we consider decoders that designate a specific set of insiders. We note that [11] claim that such decoders cannot be traced in their model; however, we overcome this limitation through the introduction of the whistleblowing mechanism. Third, they consider only decoders that receive partial decryptions, where we also study decoders that are given the insiders' secret keys.

Secret Sharing with Strong Traceability. In [43], the authors consider traitor tracing in the context of secret sharing, where a reconstruction box may specify a designated set of additional shares required for reconstruction. In their model, they also employ a whistleblowing mechanism to enable tracing. However, their setting differs from ours in two significant respects. First, their reconstruction boxes are again restricted to those constructed from small coalitions, as is inherent to the secret-sharing setting. Second, they focus exclusively on secret sharing, which is of a different nature than threshold encryption and introduces different challenges. In particular, in secret sharing the main objective is to achieve short shares, whereas in threshold encryption we aim to minimize all parameter sizes, in particular the ciphertext size. Furthermore, it is generally unclear how to extend techniques from traceable secret sharing to threshold encryption. Finally, their efficient constructions (in terms of share size) rely on indistinguishability obfuscation (iO), whereas our approach avoids the use of such heavy cryptographic primitives.

Instantiation	Traitors found	Full	Partial	Robust.	$\lvert\mathsf{ct}\rvert$	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{sk}_i\rvert$	Notes
Naive	$n/2$	1-to-1	1-to-1	$n/2$	$O(n)$	$O(n)$	$O(1)$	$\lvert\mathsf{ct}\rvert$ prohibitive
Cor. 2.10	$\tau - x$	1-to-1	$(x+1)n/2\tau$	$n/3$	$O(\tau^2)$	$O(\tau^2)$	$O(n^2/\tau)$	constant τ, $\forall x$ s.t. $0 \le x < \tau$
Cor. 2.11	$\Theta(n^a)$	1-to-1	$n/3$	$n/3$	$O(n^a)$	$O(n^{1-a})$	$O(n^{1+a})$	any $a \in (0,1)$
Thm. 2.7	$5n/22$	1-to-1	1-to-1	$\sqrt{n}/2$	$O(\sqrt{n})$	$O(n)$	$O(\sqrt{n})$	–

Figure 1: Instantiations of our compilers for $t = n/2$. The 1-to-1 indicates that full/partial insider resilience is guaranteed up to a 1-to-1 tradeoff from traitors found as per Corollary 5.1. Robustness refers to the maximum number of maliciously chosen crash failures the scheme can tolerate and still decrypt correctly.

Our Contributions. We next describe our contributions.

- We show that existing definition of threshold traitor tracing is insufficient, as it admits constructions that are vulnerable to insider attacks, revealing a conceptual gap in the current model. To address this gap, we present a generalized definition of *traceability against insider attacks* (Section 4).
- We establish an approach for achieving traceability against insiders through *multi-traitor tracing*, i.e., ensuring that the tracing algorithm identifies not just one, but at least $\tau \le t$ traitors (which we refer to as τ-traceability). In particular, we show that τ-traceability implies resilience against $\tau - 1$ full insiders (Section 5), which in turn implies resilience against $\tau - 1$ partial insiders (Corollary 5.1). Henceforth, we refer to this as a *1-to-1 tradeoff*.
- We present three compilers for multi-traitor tracing (and thus insider resilience). To the best of our knowledge, we are the first to introduce constructions that achieve multi-traitor tracing with succinct ciphertexts,[1] which we believe to be of independent interest.

[1]There exists a naive construction with ciphertext size linear in n that is t-traceable.

  - The first compiler amplifies the number of traitors identified by a given scheme by a multiplicative constant factor K (Theorems 2.2 and 2.5).
  - The second compiler reduces the ciphertext size of any $n/2$-traceable scheme by decreasing the number of traitors found to $cn/2$ for a constant $0 < c < 1$ (Theorem 2.7). The resulting scheme, however, is only robust against $O(\sqrt{n})$ maliciously chosen crash failures.
  - The third compiler amplifies the number of traitors identified by any given scheme to be $O(n^a)$ for any $0 < a < 1$. The resulting scheme incurs a ciphertext size blow-up of $O(n^a)$ (Theorem 2.8).
- We extend a construction of [11] to achieve resilience against up to $t - 1$ (i.e, maximal) partial insiders under a relaxed decoder model, while maintaining $O(1)$-size ciphertexts (Theorem 2.9). This result is notable since instantiating our compilers with a scheme that already provides partial-insider resilience yields a better tradeoff between the number of traitors identified and partial-insider resilience, surpassing the 1-to-1 tradeoff guaranteed by Corollary 5.1.
- Finally, we instantiate our compilers to obtain constructions that are multi-traitor and resilient against insiders. The results are summarized in Fig. 1. We provide the results for threshold $t = n/2$.

Remark 1.1 (Relaxed Correctness). Our compilers achieve a *relaxed* notion of correctness compared to the standard definition Specifically, we require that a uniformly distributed authorized set can decrypt except with negligible probability, whereas the standard notion requires that *every* authorized set can decrypt. Nevertheless, we argue that analyzing schemes that are restricted to our relaxed notion remains meaningful for three reasons: First, in certain practical settings such as encrypted mempools, it is reasonable to assume that the (authorized) subset of parties contributing partial decryptions is effectively sampled at random under honesty assumptions. Second, in adversarial settings where an attacker can choose to crash an arbitrary subset of parties, preventing them from participating in decryption, we naturally fall back to our robustness notion; we achieve close to optimal robustness up to an arbitrary constant factor c, i.e., can tolerate up to $n/2 - cn$ arbitrary crash failures. In more detail, $n/2 - cn$-robustness guarantees that no malicious adversary (even given all secret keys) can output a set of size $\geq n/2 + cn$ that is unable to decrypt, except with negligible probability. Third, to the best of our knowledge, relaxing correctness in this way does not trivialize the problem of constructing insider-resilient threshold traitor tracing schemes.

Takeaways. Our paper has two important takeaways:

Beyond Non-threshold Traitor Tracing. The threshold requirement must be enforced not only for decryption, but also for tracing. Existing constructions primarily focus on thresholdizing the decryption of non-threshold schemes. As we show, however, this approach is insufficient: we present a generic construction that is fundamentally not traceable in the presence of insiders.

Existing Constructions Remain Relevant. While existing schemes are not resilient against full insiders, some of them can be proven (with possible modifications) to be resilient against partial insiders. We can then achieve resiliency against full insiders using our compilers. Therefore, we view the existing literature on threshold traitor tracing as a crucial step toward achieving resiliency against insiders.

Extensions and Open Problems. Our work leaves interesting open problems.

Relationship Between Insiders and Multi-Traitor Tracing. In this paper, we show that multi-traitor tracing implies traceability against insiders. Whether the converse holds remains an open question. More broadly, it is worth investigating alternative approaches to achieving traceability against insiders.

Better Roubstenss. A particularly interesting direction for future work is improving the robustness of our $O(n)$-traceable construction. Our current construction achieves only $O(\sqrt{n})$ robustness. Such constructions may require completely new techniques.

New Constructions against Partial Insiders. We present the first construction resilient against $t-1$ partial insiders. Our construction, however, relies on a relaxed decoder model in which the decoder outputs the full decryption. An interesting direction for future work is to develop constructions that can trace distinguishing decoders. Interestingly, these constructions can also be instantiated within our compilers.

Generalizing the compilers. Our compilers assume reconstruction threshold $n/2$. However, they can be straightforwardly extended to any threshold cn for $c \in (0,1)$. Additionally, we believe that upgrading our compilers to achieve the classical correctness notion is an interesting direction for future work.

2 Technical Overview

We next give a high-level exposition of our techniques. For simplicity, we describe all constructions in this paper as key-encapsulation mechanisms (KEM). Extending KEM to encryption follows standard techniques.

Throughout this work, we consider *distinguishing decoders*, which, given a ciphertext c and a candidate KEM key k, output a bit indicating whether k is encapsulated in c. This is a stronger model than *decrypting decoders*, which output k when given c. One of our results (Theorem 2.9), however, holds only for decrypting decoders. We next present the original definition of traceability due to [11].

The Original Traceability Definition [11]. A traceable threshold KEM is essentially a standard threshold KEM with two modifications. (1) key generation outputs an additional secret tracing key tk, and (2) a Trace^D algorithm that, given the tracing key tk and black-box access to a decoder D with noticeable decryption advantage, outputs a set of traitors.

Traceability is defined through a security game between a challenger and an adversary. Initially, the challenger performs key generation on behalf of all parties. Then, the adversary can corrupt an arbitrary set of traitors $\mathcal{T}$ through a corruption oracle (learning their secret key shares) and outputs a decoder D. We say that a decoder D is *good*, if it has sufficient advantage to distinguish whether for a given tuple (k, c), the KEM key k is either encapsulated in ciphertext c, or uniformly random from the key space. The challenger then runs the tracing algorithm $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk})$, which uses a tracing key tk and has black-box access to the decoder D, outputting the set of alleged traitors $\mathcal{T}'$. The adversary is said to break traceability if (a) the challenger fails to identify at least one real traitor for good decoders, or (b) the adversary manages to frame an honest party with non-negligible probability ($\mathcal{T}' \not\subseteq \mathcal{T}$).

2.1 The Starting Point of This Paper

Our starting observation is that, under the original definition of [11], traceable threshold KEM ($\mathsf{TT\text{-}KEM}$) can already be obtained generically from any *non-threshold* traitor tracing scheme (in which every individual party with a secret key can decrypt on its own), which shows that the current definition admits constructions whose traceability relies entirely on a non-threshold component. We will see later why this is problematic for insider resilience.

Thresholdizing Traitor Tracing Generically. Let Π_{TT} be any (non-threshold) traitor tracing scheme, and Π_{TE} any (non-traceable) *threshold* KEM. We construct a *traceable threshold* KEM Π_{TTE} by running key generation in both Π_{TT} and Π_{TE} (the latter with threshold t), and assigning each party secret keys for both schemes $\mathsf{sk}_i = (\mathsf{sk}_i^{\mathsf{TT}}, \mathsf{sk}_i^{\mathsf{TE}})$. To encrypt, sample independently

$$(k^{\mathsf{TT}}, c^{\mathsf{TT}}) \xleftarrow{\$} \Pi_{\mathsf{TT}}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{TT}}) \text{ and } (k^{\mathsf{TE}}, c^{\mathsf{TE}}) \xleftarrow{\$} \Pi_{\mathsf{TE}}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{TE}}),$$

set the KEM key $k \leftarrow k^{\mathsf{TT}} \oplus k^{\mathsf{TE}}$, and assemble the ciphertext as $c \leftarrow (c^{\mathsf{TT}}, c^{\mathsf{TE}})$. Any t parties can recover k^{TE} from c^{TE}, while any single party can decrypt c^{TT} to obtain k^{TT}, hence reconstructing k.

Traceability follows via a wrapping argument. Given a good decoder D that distinguishes valid encapsulations of Π_{TTE} with advantage ε, the tracer (holding the tracing key tk of Π_{TT}) builds a decoder D' for Π_{TT} as follows: On input $(c^{\mathsf{TT}}, k^{\mathsf{TT}})$, D' samples random $(k^{\mathsf{TE}}, c^{\mathsf{TE}}) \xleftarrow{\$} \Pi_{\mathsf{TE}}.\mathsf{Enc}(\mathsf{pk}^{\mathsf{TE}})$, sets $k \leftarrow k^{\mathsf{TT}} \oplus k^{\mathsf{TE}}$, and $c \leftarrow (c^{\mathsf{TT}}, c^{\mathsf{TE}})$, queries D on (c, k), and outputs the same bit as D. Note that if D is a good decoder, i.e., it can distinguish with advantage ε whether k is encapsulated in c, then so is D' for Π_{TT}. The tracer can then trace D' using the tracing algorithm of Π_{TT}. We formalize the following theorem in Section 10 (Corollary 10.2).

Theorem 2.1 (Generic Threshold Traitor Tracing (Informal)). *If there exists a non-threshold traitor-tracing scheme Π_{TT} and a (non-traceable) threshold KEM Π_{TE}, then there exists a traceable threshold KEM Π_{TTE} whose public key, secret key, and ciphertext sizes are the sums of the corresponding sizes of Π_{TT} and Π_{TE}.*

Due to the broad literature on non-threshold traitor tracing, the above generic compiler recovers several recent results in threshold traitor tracing [11, 3, 23, 28], all of which effectively thresholdize a non-threshold traitor tracing scheme. However, it exposes a critical weakness in the presence of *insider attacks*.

Insider Attacks on the Generic Compiler and Existing Schemes. Observe that the traceable component c^{TT} can be *decrypted by a single party*, independently of the threshold component. We refer to this pattern as an *unlocking mechanism*: a single party can unlock the traceable part of the ciphertext, bypassing enforcement of the threshold requirement. Colluders can exploit this mechanism by constructing a decoder that functions only when provided with the secret key or partial decryption of a *designated insider*. This input suffices to decrypt c^{TT}, while the colluders' keys decrypt c^{TE}. Tracing such a decoder inevitably accuses the insider, never the actual colluders, eliminating any incentive for the insider to cooperate with the tracer.

More generally, any construction exhibiting the unlocking mechanism discussed above cannot be secure against full or even partial insiders. Indeed, several existing schemes suffer from this limitation, including [23][2] and [11, §5.1][3]. In contrast, constructions such as [20, 28] as well as the construction in [11, §5.2] avoid unlocking. We conjecture that these constructions can be made secure against *partial insiders*. We confirm this for the latter construction [11, §5.2], proving partial-insider resilience under a relaxed decoder model. Nevertheless, achieving full-insider resilience directly (without going via multi-traitor tracing) seems out of reach for even these constructions. The intuition is that, given the insider's secret key, the colluders can engineer the decoder's behavior so that the tracing algorithm implicates the insider. We elaborate on this in Appendix H, where we present concrete attacks.

2.2 Defining Traceability Against Insiders

To capture insider attacks, we introduce a strengthened notion of traceability against (σ, ρ)-insiders. In our definition, the adversary outputs together with the decoder D a set of full insiders $\mathcal{I}_{\mathsf{sk}}$ and partial insiders $\mathcal{I}_{\mathsf{pd}}$, with $|\mathcal{I}_{\mathsf{sk}}| \leq \sigma$ and $|\mathcal{I}_{\mathsf{pd}}| \leq \rho$. The decoder D is considered *good* if, given (k, c), the full insiders' secret keys $\{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}}$, and the partial insiders' partial decryptions $\{\mathsf{PDec}(\mathsf{sk}_i, c)\}_{i \in \mathcal{I}_{\mathsf{pd}}}$, it distinguishes whether k is encapsulated in c with non-negligible advantage ε; without this insider information, D may refuse to decrypt.

A scheme is *τ-traceable against (σ, ρ)-insiders*[4] if, for every adversary producing $(D, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}})$, it holds that $\Pr[\mathsf{GoodTr}] \geq \Pr[\mathsf{GoodDec}] - \mathsf{negl}(\lambda)$ and $\Pr[\mathsf{BadTr}] \leq \mathsf{negl}(\lambda)$, where the events

[2] This construction can be viewed as an instance of the generic compiler.

[3] We discuss an attack on this construction in Section 9.

[4] We restrict $\tau + \sigma + \rho \leq t$. If t parties can decrypt, and a decoder receives insider information from $\sigma + \rho$ parties, we can expect to find at most $\tau \leq t - \sigma - \rho$ traitors.

GoodTr, GoodDec, and BadTr are defined as follows with respect to running the tracing algorithm on D.

- GoodTr occurs, whenever at least τ *correct* traitors are found during tracing.
- GoodDec occurs whenever the adversary produces $(D, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}})$ such that D is *good* for insiders $\mathcal{I}_{\mathsf{sk}}$ and $\mathcal{I}_{\mathsf{pd}}$. More formally, if given $(k, c, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{\mathsf{PDec}(\mathsf{sk}_i, c)\}_{i\in\mathcal{I}_{\mathsf{pd}}})$, D can distinguish with noticeable advantage whether k is random, or encapsulated in the ciphertext c.
- BadTr occurs whenever the tracing algorithm *frames* an honest party when tracing D. Importantly, insiders count as honest parties, unless they were specifically corrupted by the adversary as traitors before.

We say that a TT-KEM is *maximally resilient* against full (resp., partial) insiders, if it is τ-traceable against $t - \tau$ full (resp., partial) insiders for any $0 < t \leq n$. Our definition naturally generalizes the original definition due to [11]: their definition is equivalent to 1-traceability against $(0, 0)$-insiders.

Intuitively, tracing a decoder with full insiders must somehow make use of the full insiders' secret keys, as without them, the decoder might refuse to work at all. Indeed, in Section 4.3, we show an impossibility result that no TT-KEM can be both semantically secure and 1-traceable against even a single full insider *without involving the full insiders' secret keys in the tracing process.*[5] One way to "involve" the full insiders' secret keys is to have the full insiders "complete" the decoder by hardcoding their secret keys before forwarding it to the tracer. In this paper, however, we decide to instead give the tracer direct access to the full insiders' secret keys *during tracing* (but not before). That is, we give their secret keys as input to the tracing algorithm. This models that insiders forward their secret key *alongside* the decoder to the tracer when blowing the whistle. We elaborate on this modeling choice and compare it to the alternative definition in Section 4.3.

Our formal definition can be found in Section 4. We next outline how to achieve insider-resilience.

2.2.1 Insider Resilience from Multi-Traitor Tracing

Our main approach derives insider resilience from *multi-traitor tracing*. In Section 5, we show that any τ-traceable scheme is resilient against up to $\tau-1$ full insiders at the cost of reducing the number of traitors found by the same amount (1-to-1 tradeoff).[6] We show in a similar manner how full insider resilience implies partial insider resilience (Corollary 5.1). Having reduced the problem of insider resilience to that of finding multiple traitors, we next introduce compilers that transform a traceable threshold KEM into one that identifies *more* traitors. Our compilers achieve different tradeoffs with respect to number of traitors found, robustness, and ciphertext size. We denote by $P(n)$, $S(n)$, and $C(n)$ the public key, secret share, and ciphertext size of a construction as a function of the number of parties.

Ramp Setting. We note that our compilers yield *ramp* schemes [5], requiring a gap between the security threshold t and the reconstruction threshold $r > t$. For simplicity, we carry out our analysis for reconstruction threshold $r = n/2$. In Section 10, we show how to eliminate this gap, i.e., achieving $t = r$, while preserving traceability. This is achieved via encrypting in a 2-out-of-2 fashion to the traceable ramp scheme and any (non-traceable) threshold KEM without ramp. This compiler is of independent interest; e.g., applying it to the recent construction of [28] yields a post-quantum threshold traitor tracing scheme without ramp.

[5] This is related to a result for traceable secret sharing [43], showing that reconstruction boxes that can ask for additional shares of even corrupted parties cannot be traced while maintaining secrecy against the tracer. The underlying intuition is similar to ours: A tracer cannot get the box to work unless it can produce a valid share, which would also allow it to break secrecy.

[6] Intuitively, the tracer can feed the insiders' secret keys into the decoder, and if τ traitors are found, only $\tau - 1$ of them can be insiders.

2.3 Compiler from τ-Traceability to $(K \cdot \tau)$-Traceability

We present a compiler that transforms any τ-traceable scheme (including $\tau = 1$) into a $(K \cdot \tau)$-traceable scheme for an arbitrary constant K, at the cost of a K-fold increase in ciphertext size. For simplicity, let the reconstruction threshold $t = n/2$. We partition parties into K disjoint bins $B_1, \ldots, B_K$ of size $n_{\mathsf{bin}} = n/K$. Henceforth, such a partition is called a (K, n)-balanced partition. For each bin B_i, we instantiate an independent instance of a τ-traceable threshold KEM Π with threshold $t_{\mathsf{bin}} = n_{\mathsf{bin}}/2$. To encrypt, one iterates over all bins $i \in [K]$ and generates (k_i, c_i) using the instance of bin i, computes the KEM key as $k \leftarrow \bigoplus_{i \in [K]} k_i$, and assembles the ciphertext as $c \leftarrow \{c_i\}_{i \in [K]}$. Decryption succeeds iff at least t_{bin} parties from *every* bin participate. The intuition is that, as decryption requires decrypting all components c_i, any successful decoder D must succeed in every bin, and can thus be transformed into a decoder D_i for every bin $i \in [K]$. Tracing each D_i identifies τ traitors per bin, and disjointness of the bins implies $K \cdot \tau$ distinct traitors. Unfortunately, this construction has two limitations: (1) there exist many authorized sets S with $|S| \geq t = n/2$ that do not meet the threshold t_{bin} in *every bin*, and thus would be unable to decrypt.[7] And (2), crashing $n_{\mathsf{bin}} - t_{\mathsf{bin}} \approx n/2K$ parties in any bin prevents decryption, thus breaking *robustness*.

To address (1), we relax correctness to *probabilistic correctness*, requiring that a uniformly random authorized set decrypts with overwhelming probability. Since a random set of size $n/2$ is unlikely to be perfectly balanced, we lower the per-bin threshold to $t_{\mathsf{bin}} = \frac{n}{2K} - O(\sqrt{n/K})$. By a Chernoff bound, such a set intersects every bin in at least t_{bin} parties with overwhelming probability. This yields a *ramp* threshold scheme: For reconstruction threshold $r = n/2$, we obtain a security threshold $t = n/2 - O(\sqrt{Kn})$. To deal with this, as discussed in Section 2.2.1, we apply our gap elimination compiler introduced in Section 10.

For details on our probabilistic correctness notion, we refer the reader to Definition 3.8 and Remark 1.1. Issue (2) can be mitigated using *multi-instance expansion*, discussed later in this section. The above compiler, henceforth called $\Pi_{\mathsf{1\text{-}part}}$, is formalized in the following theorem.

Theorem 2.2 (1-Partition Compiler). *Let Π be a τ-traceable* TT-KEM*. Then, there exists a $(K \cdot \tau)$-traceable ramp* TT-KEM *of sizes $(P'(n), S'(n), C'(n)) = (KP_\Pi(\frac{n}{K}), S_\Pi(\frac{n}{K}), KC_\Pi(\frac{n}{K}))$, gap $g = O(\sqrt{Kn})$, and $(\frac{n}{2K} + g/K)$-robustness.*

To trace D in each bin, we next show how to transform D into decoders D_i for Π ciphertexts for every bin $i \in [K]$. The tracer constructs a decoder D_i for all bins $i \in [K]$ as follows: On input (k_i, c_i), the decoder D_i generates encryptions in the remaining bins as $(k_j, c_j) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_j)$ for all $j \neq i \in [K]$. Then, it computes $k \leftarrow \bigoplus_{j \in [K]} k_j$, queries $D(k, c := \{c_j\}_{j \in [K]})$, and forwards D's output. It is easy to see that every decoder D_i has the same advantage against Π as the decoder D has against $\Pi_{\mathsf{1\text{-}part}}$: If k_i is encapsulated in c_i, then so is k in ciphertext c. If k_i is uniform random and independent of c_i, then so is k by the perfect security of the one-time pad. Since D_i forwards D's output, both decoders have identical distinguishing advantage. Formally proving the above is technically involved, and we summarize the main challenges below. We formalize the above in Section 6 and give a brief proof sketch here.

Proof Idea of Theorem 2.2. At a high level, we aim to show that if D is good, then tracing succeeds in all bins, and thus $K \cdot \tau$ traitors are found. A standard technique for proving this is a hybrid argument. We define hybrid $\mathcal{H}_0$ to be identical to the traceability game, and in $\mathcal{H}_j$, we modify the adversary's winning condition such that the adversary cannot win by breaking traceability in any bin $u \leq j$. As a first attempt, consider the following definition of $\mathcal{H}_j$: If, in any bin $u \leq j$, tracing identifies fewer than τ traitors, we overwrite $\mathsf{GoodTr} \leftarrow 1$, and if tracing in any bin $u \leq j$ falsely accuses a non-traitor, we overwrite $\mathsf{BadTr} \leftarrow 0$.

[7] As an extreme example, consider the set $S = B_1 \cup B_2 \cup \ldots \cup B_{K/2}$. Clearly it holds that $|S| \geq n/2$, so S should be authorized, but S cannot decrypt in any of the bins $B_{K/2+1}$ to B_K.

Observe that traceability holds by definition in the final hybrid $\mathcal{H}_K$: Either the additional condition is hit and GoodTr is set to 1, or τ traitors were found in every bin, again implying $\mathsf{GoodTr} = 1$. Likewise, BadTr is set to 0 whenever a non-traitor is falsely accused *in any bin*.

The challenge is to show that our modifications *decreased* $\Pr[\mathsf{BadTr}]$, and *increased* $\Pr[\mathsf{GoodTr}]$ by at most a negligible amount, which we aim to prove by a reduction to traceability of Π. Here, however, we run into a problem: Consider an adversary that outputs a good decoder only with probability, say $1/2$, and thus might cause $\mathsf{GoodTr} = 1$ with that same probability in $\mathcal{H}_0$. In $\mathcal{H}_K$, however, GoodTr is always 1. Clearly, this means that for such an adversary, we have increased $\Pr[\mathsf{GoodTr}]$ by $1/2$ across the hybrids, which is non-negligible. To address this issue, we do the following instead. At the beginning of tracing, we empirically test whether the decoder D is good, setting $\mathsf{test} \leftarrow 1$ if it passes and $\mathsf{test} \leftarrow 0$ otherwise. Then, whenever the modified condition in $\mathcal{H}_j$ is hit, we set $\mathsf{GoodTr} \leftarrow \mathsf{test}$ instead of $\mathsf{GoodTr} \leftarrow 1$. Assume for now that the test perfectly distinguishes good decoders from bad ones. Then, we avoid the above issue, since in the final hybrid, $\Pr[\mathsf{GoodTr}]$ would be elevated to $\Pr[\mathsf{GoodDec}] = 1/2$, and not 1. Also, observe that traceability still automatically holds in the last hybrid $\mathcal{H}_K$, since $\Pr[\mathsf{GoodTr}] \geq \Pr[\mathsf{test}] = \Pr[\mathsf{GoodDec}]$.

Finally, we design the empirical test such that ε-good decoders cannot test bad, and decoders that test good are at least $\varepsilon/4$-good, except with negligible probability, making them traceable in the base scheme Π.

Amplifying Partial Insider Resilience. So far, we have relied solely on the 1-to-1 tradeoff between traitors found and partial insiders to guarantee partial-insider resilience. To improve on this, we show next that we can achieve a better tradeoff by instantiating our compiler with a base scheme that is already resilient against ρ partial insiders (see Section 2.6 for a construction of such a scheme). In particular, instead of the generic 1-to-1 tradeoff between traitors found and partial-insider resilience, this construction achieves a 1-to-$(\rho+1)$ tradeoff. The following proposition extends Theorem 2.2.

Proposition 2.3 (1-Partition Compiler with Partial Insiders (Informal)). *Let $K \in \mathbb{N}$ be the number of bins. If Π is τ-traceable against ρ partial insiders for $\tau + \rho \leq t_{\mathsf{bin}}$, then $\Pi_{\mathsf{1\text{-}part}}$ is $((K-x)\cdot\tau)$-traceable against $(\rho + x\cdot(\rho+1))$-partial insiders for all $0 \leq x < K$.*

To see this, observe that an adversary must concentrate at least $\rho+1$ partial insiders in a bin, to cause tracing in that bin to fail. Thus, we still find τ traitors in all bins $i \in [K]$, where $|\mathcal{I}_{\mathsf{pd}} \cap B_i| \leq \rho$.

Example 2.4 (1-to-$(\rho+1)$ Tradeoff). Let Π be a 1-traceable scheme that is resilient against $\rho = t_{\mathsf{bin}} - 1$ partial insiders. Let $K = 3$. Then, the scheme $\Pi_{\mathsf{1\text{-}part}}$ is

- 3-traceable against $(t_{\mathsf{bin}}-1)$-partial insiders, and
- 2-traceable against $(2t_{\mathsf{bin}}-1)$-partial insiders, and
- 1-traceable against $(3t_{\mathsf{bin}}-1)$-partial insiders.

2.3.1 Robustness via Multi-Instance Expansion

As discussed, the above scheme only achieves weak robustness: crashing $n_{\mathsf{bin}} - t_{\mathsf{bin}} \approx n/2K$ parties in any bin causes decryption to fail. To improve robustness, we instantiate ℓ independent copies of the 1-partition scheme $\Pi_{\mathsf{1\text{-}part}}$, sampling ℓ (K,n)-balanced partitions independently at key generation. To encrypt, sample a random key $k \xleftarrow{\$} \{0,1\}^\lambda$, and for each partition $j \in [\ell]$, generate $(k_j, c_j) \xleftarrow{\$} \Pi_{\mathsf{1\text{-}part}}.\mathsf{Enc}(\mathsf{pk}_j)$. Set $u_j \leftarrow k \oplus k_j$ and $c \leftarrow \{(c_j, u_j)\}_{j\in[\ell]}$, which forms a 1-out-of-ℓ sharing of k across partitions.

The key intuition is that decryption succeeds if the honest parties successfully decrypt in at least one partition. Consequently, an adversary must block decryption in all ℓ partitions to break robustness, which, intuitively, should require crashing more than $n/(2K) - t_{\mathsf{bin}}$ parties. In

fact, we show that we can achieve robustness against $n/2 - cn$ maliciously chosen crash failures for any constant $c \in (0, 1/2)$, at the cost of increasing ciphertext size by a factor $O(\frac{K(1-c)}{c^2})$.

Tracing via a PLBE-style Technique. At a high level, the tracing in our $\Pi_{\ell\text{-part}}$ compiler relies on a technique originating from Private Linear Broadcast Encryption (PLBE) [15], where we view the partitions as users. A similar idea was introduced by Zhandry [54] to extend traceable broadcast encryption schemes to support more users. We note that we extend this idea to the threshold setting in Section 8.

To build intuition, consider a decoder D against the multi-instance scheme $\Pi_{\ell\text{-part}}$. To decrypt a ciphertext $c = \{c_j, u_j\}_{j\in[\ell]}$ successfully, it must decrypt successfully in at least one partition $j \in [\ell]$. Hence, if D has advantage ε, there must exist at least one partition, where D has advantage ε/ℓ.

More formally, given a decoder D that has advantage ε against $\Pi_{\ell\text{-part}}$, we construct a decoder D_j against the scheme $\Pi_{1\text{-part}}$ in partition j for all $j \in [\ell]$ as follows. On input (c_j, k_j), D_j generates ciphertexts for all remaining partitions $q \in [\ell] \setminus \{j\}$ as $(k_q, c_q) \xleftarrow{\$} \Pi_{1\text{-part}}.\mathsf{Enc}(\mathsf{pk}_q)$. Then, it samples two random KEM keys $k^0, k^1 \xleftarrow{\$} \{0,1\}^\lambda$, and shares k^0 among all $q \geq j$ partitions, while revoking all partitions q with $q < j$. More precisely, for $q < j$, sample $u_q \xleftarrow{\$} \{0,1\}^\lambda$, and for $q \geq j$, set $u_q \leftarrow k_q \oplus k^0$. Then, the decoder D_j samples a random bit b and outputs $b \oplus D(\{c_q, u_q\}_{q\in[\ell]}, k^b)$.

We argue that, if D is ε-good, then there must exist at least one partition $j \in [\ell]$, for which D_j is $\varepsilon/2\ell$-good. Denote with $X_j(b)$ the distribution $(k^b, \{(c_q, u_q\}_{q\in[\ell]})$, generated by sampling random k^0, k^1, and encrypting k^0 while revoking all u_q for $q \leq j$ by replacing them with random values. Denote with p_j the decoder D's probability to distinguish $X_j(0)$ from $X_j(1)$, i.e.,

$$p_j \coloneqq \Pr\left[D(X_j(b)) = b\right].$$

Observe that, since D is ε-good, it must hold that $p_0 \geq 1/2 + \varepsilon$. Further, since in $X_\ell(b)$ all partitions are revoked, it holds that $X_\ell(0) \equiv X_\ell(1)$, and thus $p_\ell = 1/2$. Hence, by pigeonhole principle, there must exist some $j \in [\ell]$ such that the distinguishing probability *drops significantly*, i.e., $p_j \leq p_{j-1} - \varepsilon/\ell$.

Now, let j^* be such j. When k_{j^*} is valid, i.e., k_{j^*} is encapsulated in c_{j^*}, D_{j^*} effectively runs D on $X_{j^*-1}(b)$, and outputs 0, whenever D guesses b *correctly*, and 1 otherwise. In other words, D_{j^*} *correctly* guesses that k_{j^*} is valid with probability p_{j^*-1}. If, however, k_{j^*} is invalid, then D_{j^*} runs D on $X_{j^*}(b)$, and outputs 1, whenever D guesses b *incorrectly*. Thus, D_{j^*} *correctly* guesses that k_{j^*} is invalid with probability $1 - p_{j^*}$. Overall, we conclude that D_{j^*} correctly *predicts* the validity of k_{j^*} with probability

$$\frac{1}{2}(p_{j^*-1} + 1 - p_{j^*}) = \frac{1}{2} + \frac{p_{j^*-1} - p_{j^*}}{2} \geq \frac{1}{2} + \frac{\varepsilon}{2\ell}.$$

This means that D_{j^*} is $(\varepsilon/2\ell)$-good.

Tracing the *good* decoder D_{j^*} then identifies at least τ traitors. A full formalization appears in Section 6.2, where we show that traceability is preserved for any τ-traceable TT-KEM as base scheme, not only $\Pi_{1\text{-part}}$. The following theorem shows that multi-instance expansion as above maintains traceability.

Theorem 2.5 (Multi-Instance Expansion (Informal)). *Let Π be a TT-KEM that is τ-traceable against ρ-partial insiders. Then, expanding Π to ℓ instances retains τ-traceability against ρ-partial insiders. The public key, secret key, and ciphertext sizes grow by a factor ℓ.*

Consider instantiating Theorem 2.2 with a 1-traceable scheme Π across K bins. To achieve robustness against $n/2 - cn$ maliciously chosen crash failures for $0 < c < 1/2$, it suffices to have $\ell = \Theta((1-c)K/c^2)$ partitions. We prove this by showing that, with this choice of ℓ, except with

negligible probability, no set S with $|S| \leq n/2 - cn$ intersects a bin in at least $\frac{n}{2K} - t_{\mathsf{bin}}$ parties in all partitions. Then, applying Theorem 2.5 with $\ell = \frac{2K(1-c)}{c^2}$ yields the following Corollary.

Corollary 2.6 *For any $K \in \mathbb{N}$ and $c \in (0, 1/2)$ there exists a K-traceable* TT-KEM *with $(\frac{n}{2} - cn)$-robustness and parameter sizes $(P'(n), S'(n), C'(n)) = \frac{2K^2(1-c)}{c^2} \cdot \Big(P(n/K), \frac{1}{K}S(n/K), C(n/K)\Big)$.*

2.4 Tracing $\Theta(t)$ Traitors

Our next goal is to identify a linear number of traitors while maintaining *succinct* ciphertexts. Starting from a t-traceable scheme with $O(n)$-size ciphertexts, our approach is to design a compiler that reduces the ciphertext size of the scheme while maintaining a linear number of traitors found. To do so, we modify the 1-partition scheme as follows. We first fix $K = \sqrt{n}$ bins, and thus $n_{\mathsf{bin}} = \sqrt{n}$ and $t_{\mathsf{bin}} = \frac{\sqrt{n}}{2} - O(n^{1/4})$. As building blocks, we use any (not necessarily traceable) threshold KEM with ciphertext size $O(1)$, denoted by TE, and the naive t-traceable scheme Π_{naive} with ciphertext size $O(n)$ (discussed in Appendix E).

During key generation, we set up one instance of Π_{naive} and one instance of TE in each of the $\sqrt{n}$ bins. Then, to encrypt, one samples a random bin $i \in [\sqrt{n}]$, and encrypts as in the $\Pi_{\mathsf{1\text{-}part}}$ compiler, but using the instance of Π_{naive} in the i'th bin, and the instances of TE in all remaining bins. Clearly, since in each bin we instantiate with $n_{\mathsf{bin}} = \sqrt{n}$, and only one bin contains the Π_{naive} ciphertext, the overall ciphertext size is $\Theta(\sqrt{n})$, assuming that C_{TE} is constant in n. Denote with *type-i ciphertexts* a ciphertext where Π_{naive} is in the i'th bin.

Tracing Intuition. Suppose, for now, that D is *perfect*, i.e., that it always decrypts correctly. Then, D also perfectly decrypts type-i ciphertexts for all $i \in [\sqrt{n}]$. Similarly to $\Pi_{\mathsf{1\text{-}part}}$, we can transform D into a decoder D_i that is good against the Π_{naive} instance in each bin $i \in [\sqrt{n}]$, and tracing D_i in Π_{naive} will yield t_{bin} traitors. As bins are disjoint, we find $t = \sqrt{n} \cdot t_{\mathsf{bin}} = n/2 - o(n)$ traitors in total.

The above argument assumes that the decoder D is perfect, and thus successfully decrypts ciphertexts of all types. However, a decoder may choose to decrypt only ciphertexts of, say, a single type (corresponding to a fixed bin i), in which case tracing would succeed for only one bin. To address this issue, we first consider decoders that are ε-good for some constant $\varepsilon \in (0, 1/2)$. We show by a counting argument that such a decoder must successfully decrypt type-i ciphertexts for sufficiently many bins i. As an example, if $\varepsilon = 1/4$, we show that there must be at least $5\sqrt{n}/11$ bins i, where D_i is at least $\varepsilon' = 1/24$ good. Tracing in these bins yields $\frac{5n}{22} - o(n)$ traitors. Finally, to remove the limitation that decoders must be at least $\varepsilon \geq 1/4$-good, we can apply the limit elimination compiler (Theorem C.2) due to [54, 28], which upgrades traceability to hold for decoders with arbitrarily small non-negligible advantage. We extend it straightforwardly to the multi-traitor and insider setting in Appendix C. The parameter sizes increase only by a constant amount. Unfortunately, this approach achieves at most $\Theta(\sqrt{n})$-robustness. Applying multi-partition expansion does not help, because, to achieve $n/2 - cn$ robustness, one would need to instantiate $\Theta(\sqrt{n})$ partitions. As each partition already has ciphertext size $\Theta(\sqrt{n})$, the overall ciphertext size would again be $\Theta(n)$.

The traceability proof essentially extends that of the 1-partition compiler by the analysis that any good decoder must be sufficiently good when running against Π_{naive} in sufficiently many bins. We defer formalization to Section 7.

Theorem 2.7 *There exists a $\frac{5n}{22}$-traceable ramp* TT-KEM *with $\Theta(\sqrt{n})$-robustness and ciphertext size $\Theta(\sqrt{n})$.*

2.5 Compiling from 1-traceability to $\Theta(n^a)$-traceability for $a \in (0, 1)$

So far, we have shown how to trace a constant number of traitors with strong robustness (Section 2.3), and how to identify $\Theta(t)$ traitors with limited robustness (Section 2.4). Our goal is

now to combine these guarantees: namely, to achieve robustness close to $n/2$ while tracing a non-constant number of traitors (e.g., $\Theta(\sqrt{n})$), with only sublinear ciphertext blowup. Moreover, we aim to maintain almost maximal resilience against partial insiders whenever the underlying scheme itself is maximally resilient.

A natural approach is to instantiate the $\Pi_{\mathsf{1\text{-}part}}$ compiler with $K = \sqrt{n}$ bins, which yields $\Theta(\sqrt{n})$ traitors. However, maintaining $(n/2 - cn)$-robustness requires multi-instance expansion with $\sqrt{n}$ partitions (by Corollary 2.6), resulting in linear ciphertext blowup, since we incur $\sqrt{n}$ partitions $\times$ $\sqrt{n}$ bins. The first attempt to achieve stronger robustness is to require threshold decryption across bins within a single partition (rather than K-out-of-K). This, however, comes at the cost of reduced resilience against partial insiders. Achieving $(\frac{n}{2} - cn)$-robustness in this manner provides resilience against at most cn partial insiders, for $c \in (0, 1/2)$. For example, achieving $\frac{n}{3}$-robustness is possible only if the scheme tolerates at most $\frac{n}{6}$ partial insiders. Hence, our next goal is to achieve both $(n/2 - cn)$-robustness and $(n/2 - cn)$-resilience against partial insiders, for *small* $c \in (0, 1/2)$ at the same time, while tracing $\Theta(n^a)$ traitors with sublinear ciphertext blowup.

To avoid the linear blowup incurred by large partitions, we reduce the number of bins per partition, K, to $o(\sqrt{n})$, specifically to a constant. In $\Pi_{\mathsf{1\text{-}part}}$, however, K directly bounds the number of traitors found (assuming a base scheme with $\tau = 1$). Hence, the challenge is to find $\Theta(\sqrt{n})$ traitors while using partitions with only a constant number of bins. The key idea is to trace across $\ell = \Theta(\sqrt{n})$ partitions, rather than relying on a single partition with $\Theta(\sqrt{n})$ bins. Achieving this requires forcing the decoder to succeed in many partitions. We do so by sharing the KEM key in a *threshold* manner across partitions, rather than using the 1-out-of-ℓ approach of the multi-instance expansion compiler. Concretely, with an $\ell/2$-out-of-ℓ threshold, any good decoder must succeed in at least $\ell/2$ partitions, enabling tracing in each. Our first challenge is to show that any good decoder can be converted into one that is good in at least $\ell/2$ partitions, so that tracing can be applied to each. In more detail, we generalize the PLBE-style approach used in the multi-instance compiler (c.f. Section 2.3), where partitions are treated as parties, to the threshold setting. The intuition is that the decoder must successfully decrypt in at least $\ell/2$ partitions. Therefore, after revoking $\ell/2 + 1$ partitions, its distinguishing advantage must be 0. This allows us to identify traitors among the first $\ell/2$ partitions in the revocation order. Accordingly, in the tracing algorithm, whenever we identify a traitor in some partition, we move that partition to the end of the revocation order. In other words, this partition will no longer be among the first $\ell/2$ partitions in subsequent iterations, forcing the algorithm to identify a traitor from a different partition in every step. Overall, the process runs for $\ell/2$ iterations and therefore identifies $\ell/2$ traitors from distinct partitions. We note that the proof requires the underlying scheme $\Pi_{\mathsf{1\text{-}part}}$ to remain traceable even against decoders whose advantage is as low as $\Omega(1/\ell)$. Since some constructions only support tracing for larger advantages, one may first apply our limit-elimination compiler (which generalizes the compiler of [54] to the insider setting), described in Appendix C.

However, even after resolving the previous challenge and tracing across $\Theta(\sqrt{n})$ partitions, a critical issue remains: most found traitors may be duplicates. For example, if $K = 2$ with each bin of size $n/2$, a party appears (in expectation) in half of the partitions in the first bin and half in the second. An adversary can then select a small set S (of constant size) whose members appear in *both* bins in $\ell/2$ partitions, so tracing may yield only $|S|$ distinct traitors. To overcome this, the key idea is to use *unbalanced partitions*, consisting of a small bin of size $\sqrt{n}$ and a large bin of size $n - \sqrt{n}$. By a simple probabilistic argument, a party can appear in the small bin only in a small fraction $2/\sqrt{n}$ of partitions. Consequently, tracing a traitor from the small bin in $\ell/2$ partitions (for $\ell = \Theta(\lambda \cdot \sqrt{n})$) guarantees that at least $\frac{0.5\ell}{2\ell/\sqrt{n}} = \frac{\sqrt{n}}{4}$, of the traced traitors are *distinct*.

It remains to show that the construction achieves the required level of robustness and resilience against partial insiders. The intuition is that, with an appropriate choice of ℓ, the adversary

cannot select a subset of size smaller than $n/2 - cn$ that prevents decryption or tracing in a large fraction of the partitions. The following theorem is proven in Section 8.

Theorem 2.8 (Unbalanced-Partitions Compiler (Informal)). *Let* Π *be a* 1*-traceable* TT-KEM*. For all constants* $a \in (0,1)$ *and* $c \in (0,1/2)$ *there exists an* $\Theta(n^{1-a})$*-traceable* TT-KEM *with* $(n/2-cn)$*-robustness and parameter sizes* $(P'(n), S'(n), C'(n)) = \ell \cdot (P_\Pi(n^a), S_\Pi(n^a), C_\Pi(n^a))$*, where* $\ell = \Theta(\lambda n^{1-a}/c^2)$*. If* Π *is maximally partial-insider resilient, the resulting scheme is* $(n/2-cn)$*-partial-insider resilient.*

2.6 A Construction with Maximal Partial Insider Resilience

Next, we discuss how to extend the construction in [11, §5.2] to achieve traceability against up to $(t-1)$ partial insiders.

Tracing in [11] builds on the non-threshold traitor tracing construction of [9, 4], which samples a fingerprinting code [14] with codewords $(\mathsf{cw}^{(1)}, \ldots, \mathsf{cw}^{(n)}) \in \{0,1\}^\ell$. For each position $j \in [\ell]$ and bit $v \in \{0,1\}$, a key pair $(sk_{j,v}, pk_{j,v})$ is generated using a standard public key encryption scheme. User i receives the secret keys corresponding to its codeword $\{\mathsf{sk}_{j,\mathsf{cw}_j^{(i)}}\}_{j\in[\ell]}$. To encrypt, a random position $j \in [\ell]$ is chosen and the same KEM key is encrypted under both left and right public keys $\mathsf{pk}_{j,0}$ and $\mathsf{pk}_{j,1}$, yielding a ciphertext $((c_0, c_1), j)$, where c_0 is the *left* ciphertext, and c_1 is the *right* ciphertext. A decoder that holds only left (resp. right) keys at position j can decrypt only c_0 (resp. c_1). This asymmetry allows extraction of a codeword from any successful decoder, which can then be traced using the fingerprinting code. The work of [11] extends this approach to the threshold setting via a bipartite threshold KEM (BT-KEM) satisfying two properties: *Two-sided correctness* ensures that any t parties, possibly holding mixed keys, can decrypt. *One-sided security* ensures that adversaries who only hold keys from one side at position j cannot distinguish if the ciphertext of the other side is consistent with the given KEM-key (i.e., valid) or not (i.e., invalid).

Our new BT-KEM (PD-BT-KEM)**.** We extend the result of [11] to consider *decoders that receive partial decryptions*. To handle this, we allow the tracer to simulate partial decryptions and define two new security properties, replacing one-sided security:

Property 1 (Right-Sided SPD Security) ensures that adversaries who hold only right keys for a position j cannot decrypt the KEM-key encapsulated in the *left ciphertext*, even if they obtain up to $t-1$ simulated partial decryptions consistent with the left ciphertext.

Property 2 (Left-Sided SPD Security) ensures that adversaries with only left keys for a position j cannot distinguish whether left and right ciphertexts encapsulate *the same* key, even if they receive up to $t-1$ simulated partial decryptions consistent with the left ciphertext.

We show how to realize the above notions by proving that the BT-KEM construction in [11, §5.2] can be made Right- and Left-SPD secure with only minimal modifications. Our proof relies on an interactive assumption, which we prove to hold in the GGM [52, 48]. Interestingly, we show that the other BT-KEM construction in [11, §5.1] is not secure under our Right-SPD notion, indicating that not every BT-KEM is also PD-BT-KEM. We defer the discussion to Section 9.2.

Partial-Insider Resilient Tracing from PD-BT-KEM**.** Suppose the adversary outputs a decoder D and a set of partial insiders $\mathcal{I}_{\mathsf{pd}}$ with $|\mathcal{I}_{\mathsf{pd}}| < t$. At a high level, tracing follows the pattern of [9, 11], i.e., we extract for which positions the decoder holds left (resp. right) keys, and then feed this information into the fingerprinting code to identify a traitor. Now, however, we need to simulate partial decryptions for $\mathcal{I}_{\mathsf{pd}}$ when querying the decoder, relying on our new properties of Right-SPD and Left-SPD security. For each position j, the tracer queries D on valid and invalid ciphertexts while simulating partial decryptions consistent with the left ciphertext. For each position, we count how many times the decoder outputs a key that is consistent with the left part. Let p_j count the number of trials where this happens for valid ciphertexts, and q_j for invalid ciphertexts, respectively. We then extract information by the following two arguments:

(1) If $q_j > 0$, i.e., if the decoder outputs the key encapsulated in the left ciphertext *at least once* for invalid ciphertexts, we conclude that it must have a left secret key in position j. Otherwise, the decoder would break Right-SPD security (Property 1).

(2) If q_j and p_j are significantly far apart, we conclude that the decoder must have a right secret key in position j, since valid and invalid ciphertexts are indistinguishable without right keys due to Left-SPD security (Property 2).

Traceability then follows by reduction to these properties, extending the analysis of [9]. We formalize the above approach in Section 9, where we establish the following theorem.

Theorem 2.9 ($(t-1)$-Partial Insiders (Informal)). *There exists a 1-traceable* TT-KEM *that is secure against $(t-1)$-partial insiders, where t is the threshold. The scheme has secret key size $O(n^2)$, and constant-size public keys and ciphertexts. Partial insider resilience holds against decrypting decoders.*[8]

We instantiate our compilers using the construction from Theorem 2.9. A comparison of the resulting schemes for concrete parameters appears in Figure 1. The first corollary follows from the 1-partition compiler together with a multi-instance expansion for robustness (Theorems 2.2 and 2.5), while the second follows from our unbalanced compiler (Theorem 2.8).

Corollary 2.10 *For any constant K and $c \in (0, 1/2)$ there exists a* TT-KEM *that is $(K-x)$-traceable against $(x \cdot t_{\mathsf{bin}} + t_{\mathsf{bin}} - 1)$ partial insiders for $t_{\mathsf{bin}} = n/2K - O(\sqrt{n/K})$ for all $0 \le x < K$. The scheme achieves $(n/2 - cn)$-robustness and parameter sizes $(P(n), S(n), C(n)) = \frac{2K^2(1-c)}{c^2} \cdot (O(1), O(n^2/K^2), O(1))$. Partial insider resilience is guaranteed against decrypting decoders.*

Corollary 2.11 *For arbitrary $a \in (0, 1)$ and $c \in (0, 1/2)$, there exists a* TT-KEM *that is $\Theta(n^{1-a})$-traceable with $n/2 - cn$ robustness and sizes $(P(n), S(n), C(n)) = \ell \cdot (O(1), O(n^{2a}), O(1))$, for $\ell = \Theta(\lambda n^{1-a}/c^2)$. Furthermore, it is resilient against $(n/2) - cn$ partial insiders for decrypting decoders.*

2.7 Additional Related Work

Traitor tracing, introduced by Chor et al. [25] for piracy protection, has been extensively studied in broadcast encryption. Notably, Zhandry [55] constructed a pairing-based scheme with parameters independent of the number of users, with similar results via LWE [41, 24] and iO-based techniques [16, 42].

To the best of our knowledge, traceability in the presence of insiders was not studied previously. In [20], the authors construct a multi-traitor tracing scheme that achieves *amortized* constant-size ciphertexts in an epoch-based setting, but linear size in the worst case; in contrast, our compilers always achieve sublinear ciphertext size.

Other works study traitor tracing in related primitives such as secret sharing [12, 44, 36], and threshold VRFs [13], however, they work only with small coalitions, do not consider insider attacks, and it is unclear how to extend these techniques to traceable threshold encryption.

In [32, 18], the authors propose an entirely different collusion model where committee members can collude in MPC (without a decoder). However, their constructions are currently limited to secret-sharing schemes and rely on a strong, non-standard assumption, in particular, the existence of MPC-hard functions.

In [45], the authors use related techniques to construct silent ramp threshold encryption. In their construction, a ciphertext is encrypted to a small random committee of size $O(\lambda)$, independent of n. Security and robustness are then derived from the fact that the exact committee is *hidden*: without a party's secret key, it is infeasible to determine whether that party

[8] Unfortunately, our construction only guarantees partial-insider resilience for *decrypting decoders*: While Right-SPD security prevents adversaries without left keys from decrypting the left ciphertext, simulated partial decryptions may still allow them to *distinguish* valid from invalid ciphertexts.

belongs to the committee. We view it as an interesting direction for future work to consider collusion resilience in this setting, especially given that even a small coalition of constant size $\ll t$ can adaptively compromise secrecy and robustness on a per-ciphertext basis.

2.8 Roadmap

The remainder of the paper is organized as follows. After introducing preliminary notation and definitions in Section 3, we present our definition of threshold (multi-)traitor tracing with insider resilience in Section 4. In Section 5, we show that multi-traitor tracing implies full insider resilience, which in turn implies partial insider resilience. We then move on to introduce our compilers for multi-traitor tracing:

- Section 6.1: Our 1-partiotion compiler to identify a constant number of traitors.
- Section 6.2: Our Multi-instance expansion for improved robustness.
- Section 7: Our $\Theta(t)$-traceable compiler.
- Section 8: Our unbalanced-bins compiler for $\Theta(n^a)$-traceability with $n/2 - cn$ robustness.

In Section 9, we show our fingerprinting-based construction that achieves maximal partial insider-resilience, albeit for decrypting decoders. Finally, in Section 10, we formalize both our generic compiler for thresholdizing traitor tracing schemes *without* insider-resilience, and our gap-elimination compiler to eliminate the gap between the security threshold t and the reconstruction threshold r without compromising traceability.

3 Preliminaries

Notation. We denote the security parameter by λ, and write $\mathsf{negl}(\lambda)$ (resp. $\mathsf{poly}(\lambda)$) for functions that are negligible (resp. polynomials) in λ. We denote the number of parties by n. When describing asymptotic properties of our constructions, we ignore factors other than n in the O-notation. For integers $a, b \in \mathbb{N}$, we denote the set $\{1, \ldots, b\}$ by $[b]$ and the set $\{a, \ldots, b\}$ by $[a, b]$. We write $y \leftarrow A(x; r)$ to execute the algorithm A on input x with randomness r and assign the result to y. If A is executed with uniform randomness, we abbreviate $y \xleftarrow{\$} A(x)$ to indicate that A is probabilistic. We write $S \xleftarrow{\$} \binom{M}{t}$ to denote the process of sampling a subset S of size t uniformly at random from a set M.

Lagrange Interpolation. Some of our constructions rely on Lagrange interpolation. Denote with $\lambda_{i,S}^{(x)} = \prod_{j \in S \setminus \{i\}} \frac{x-j}{i-j}$ the Lagrange coefficient for interpolating a polynomial at position x using index i among the set of indices S. Then, given points y_i for $i \in S$, the polynomial $p(x) = \sum_{i \in S} \lambda_{i,S}^{(x)} \cdot y_i$ is of degree $|S| - 1$, and $p(i) = y_i$ for all $i \in S$.

Shamir's Secret Sharing. We use Shamir's secret sharing $(\mathsf{Share}, \mathsf{Rec})$ to share a secret s across a threshold t of n shares.

Definition 3.1 (Shamir Secret Sharing). For a field $\mathbb{F}$, Shamir's t-out-of-n secret sharing is a tuple of algorithms $(\mathsf{Share}, \mathsf{Rec})$.

- $\{s_i\}_{i \in [n]} \xleftarrow{\$} \mathsf{Share}(s, n, t)$. To share a secret $s \in \mathbb{F}$ into $n \in \mathbb{N}$ shares with threshold $0 < t \leq n$, the dealer samples a uniformly random polynomial p of degree $t - 1$ over $\mathbb{F}$ such that $p(0) = s$ and sets $s_i = p(i)$.
- $s \leftarrow \mathsf{Rec}(t, S, \{s_i\}_{i \in S})$. To reconstruct the secret s from at least t shares $\{s_i\}_{i \in S}$ corresponding to the set of indices S with $|S| \geq t$, one interpolates $s = p(0) = \sum_{i \in S} \lambda_{i,S}^{(0)} \cdot s_i$.

Any set of $t - 1$ shares does not leak information about the secret and appears uniformly random over $\mathbb{F}^{t-1}$.

Tail Bounds. Throughout this work, we use the following formulation of the Chernoff bound [50].

Lemma 3.2 (Chernoff Bound). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0,1\}$. Let $X = \sum_{i=1}^{n} X_i$, and $\mu = \mathbb{E}[X]$. Then, for any $\delta > 0$ it holds that*

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right),$$

and for $0 < \delta < 1$ it holds that

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2}\right).$$

We also make use of Hoeffding's inequality.

Lemma 3.3 (Hoeffding Inequality). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[a,b]$. Let $X = \sum_{i=1}^{n} X_i$, and $\mu = \mathbb{E}[X]$. Then, for any $t > 0$ it holds that*

$$\Pr[X \leq \mu - t] \leq \exp\left(-\frac{2t^2}{n(b-a)^2}\right) \text{ and } \Pr[X \geq \mu + t] \leq \exp\left(-\frac{2t^2}{n(b-a)^2}\right).$$

3.1 Partitions of Parties

Definition 3.4 ((K,n)-Partition). For $K, n \in \mathbb{N}$, a (K,n)-partition of a set of n parties is a collection $P = (B_1, \ldots, B_K)$ of K pairwise disjoint subsets of parties, i.e., $B_i \subseteq [n]$ for all $i \in [K]$, $B_i \cap B_j = \emptyset$ for all $i \neq j \in [K]$, and $\bigcup_{i\in[K]} B_i = [n]$. We refer to each subset B_i as a *bin*. We say that P is *balanced* if all bins have the same size, i.e., $|B_i| = |B_j|$ for all $i, j \in [K]$. Otherwise, P is said to be *unbalanced*. We denote $\mathsf{bin}(P, i) = j$ to indicate that $i \in B_j$ in partition P.

We will frequently need to argue whether a given set of parties controls sufficiently many members of a bin, a partition, or a set of partitions. For this, we introduce the following notation.

Definition 3.5 (Domination). Let $K, n \in \mathbb{N}$ and let $P = \{B_j\}_{j\in[K]}$ be a (K,n)-partition. For $S \subseteq [n]$, $r, \beta, \ell \in \mathbb{N}$, and $\mathbf{r} = (r_1, \ldots, r_K) \in \mathbb{N}^K$, we say that

- S *r-dominates a bin B* if $|S \cap B| \geq r$.
- S *$\mathbf{r}$-dominates the partition P* if the set S r_j-dominates B_j for all $j \in [K]$.
- S *$(\beta, \mathbf{r})$-dominates a collection of ℓ (K,n)-partitions $\{P_i\}_{i\in[\ell]}$* if there exist indices $i_1, \ldots, i_\beta \in [\ell]$ such that S $\mathbf{r}$-dominates partition P_{i_j} for all $j \in [\beta]$.
- S *$\mathbf{r}$-blocks partition P* if there exists a $j \in [K]$ such that S r_j-dominates B_j.
- S *$(\beta, \mathbf{r})$-blocks a collection of ℓ (K,n)-partitions $\{P_i\}_{i\in[\ell]}$* if there exist indices $i_1, \ldots, i_\beta \in [\ell]$ such that S $\mathbf{r}$-blocks partition P_{i_j} for all $j \in [\beta]$.

3.2 (Traceable) Threshold KEMs

Next, we define TT-KEMs. A definition of traceability is deferred to Section 4.

Definition 3.6 ((Traceable) Threshold KEMs). A traceable threshold KEM is a tuple of algorithms $\mathsf{TT\text{-}KEM} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{PDec}, \mathsf{Combine}, \mathsf{Trace})$. It has key-space $\mathcal{K}_\lambda$, randomness space $\mathcal{R}_\lambda$, and for $n \in \mathbb{N}$, it supports reconstruction thresholds in $\mathcal{S}_n \subseteq [n]$. For all $r \in \mathcal{S}_n$ it supports security thresholds $t(r) \leq r$. In more detail, let

- $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ generates parameters that are public input to all algorithms.
- $(\mathsf{pk}, \{sk_i\}_{i\in[n]}, \mathsf{tk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$. Given the number of parties n, reconstruction threshold $r \in \mathcal{S}_n$, security threshold $0 < t \leq t(r)$, and the limit ε, KGen outputs a public key pk, n secret key shares $\{\mathsf{sk}_i\}_{i\in[n]}$, and a tracing key tk. Sometimes we replace n with a concrete set of parties for convenience.

- $(k, c) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}; r_{\mathsf{Enc}})$. Given public key pk and randomness $r_{\mathsf{Enc}} \in \mathcal{R}_\lambda$, the encryption algorithm outputs a KEM key $k \in \mathcal{K}_\lambda$ and a ciphertext c.
- $d_i/\perp \leftarrow \mathsf{PDec}(\mathsf{sk}_i, c)$. Given a secret key sk_i and a ciphertext c, the partial decryption algorithm outputs a partial decryption d_i or $\perp$ to indicate failure.
- $k/\perp \leftarrow \mathsf{Combine}(\mathsf{pk}, S, \{d_i\}_{i\in S}, c)$. Given the public key pk and decryption shares d_i from a set of at least r parties S on a ciphertext c, the $\mathsf{Combine}$ algorithm returns a KEM key $k \in \mathcal{K}$ or $\perp$ to indicate failure.
- $\mathcal{T}' \xleftarrow{\$} \mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \ldots)$. The Trace algorithm will be defined in Section 4.

We consider a non-traceable threshold KEM T-KEM to be a TT-KEM without the parts highlighted in gray, i.e., where $\varepsilon = \mathsf{tk} = \perp$ and Trace is undefined. We say that a TT-KEM has *no ramp* if $t(r) = r$.

We move on to define correctness, robustness, and security.

Definition 3.7 (Perfect Correctness). We say that a TT-KEM Π is perfectly correct if for all $\lambda \in \mathbb{N}$, all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, all $n \in \mathsf{poly}(\lambda)$, all $r \in \mathcal{S}_n$, and all $0 < t \le t(r)$, all $\varepsilon \in (0, 1/2]$, all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, all $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}) \in \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$, all $(k, c) \in \mathsf{Enc}(\mathsf{pk})$, and all $S \subseteq [n]$ with $|S| \ge r$ it holds that

$$\mathsf{Combine}(\mathsf{pk}, S, \{\mathsf{PDec}(\mathsf{sk}_i, c)\}_{i\in S}, c) = k.$$

Some of our constructions rely on relaxing the standard correctness property of threshold encryption to what we call *probabilistic correctness*. Intuitively, probabilistic correctness states that a random authorized set can decrypt with overwhelming probability.

Definition 3.8 (Probabilistic Correctness). We say that a TT-KEM Π is probabilistically correct if there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$ there exists some $n_0 \in \mathsf{poly}(\lambda)$, such that for all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, all $n \ge n_0$, all $r \in \mathcal{S}_n$, and all $0 < t \le t(r)$, all $\varepsilon \in (0, 1/2]$, all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, all $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}) \in \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$, all $(k, c) \in \mathsf{Enc}(\mathsf{pk})$, and all r' with $r \le r' \le n$, it holds that

$$\Pr_{S \xleftarrow{\$} \binom{[n]}{r'}} \left[\mathsf{Combine}(\mathsf{pk}, S, \{\mathsf{PDec}(\mathsf{sk}_i, c)\}_{i\in S}, c) = k\right] \ge 1 - \mathsf{negl}(\lambda).$$

We also define robustness against up to c maliciously chosen crash failures. The definition below argues that no PPT adversary can cause decryption to fail by crashing at most c parties, except with negligible probability.

Definition 3.9 (c-robustness). For a function $\mathsf{c}(n, r)$ with $0 \le \mathsf{c}(n, r) \le n - r$, we say that a TT-KEM is robust against c maliciously chosen crash failures if for all PPT adversaries $\mathcal{A}$ there exists some $n_0 \in \mathsf{poly}(\lambda)$ such that for all $n \ge n_0$, all $r \in \mathcal{S}_n$, and all $0 < t \le t(r)$, all $\varepsilon \in (0, 1/2]$, it holds that

$$\Pr\left[\begin{array}{c} |S| \ge n - \mathsf{c}(n, r) \wedge \\ \mathsf{Combine}\left(\mathsf{pk}, S, \{\mathsf{PDec}(\mathsf{sk}_i, c)\}_{i\in S}, c\right) \ne k \end{array} : \begin{array}{c} (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon) \\ (k, c) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}) \\ S \xleftarrow{\$} \mathcal{A}(1^\lambda, n, r, t, \mathsf{pk}, \mathsf{sk}, \varepsilon, \mathsf{tk}, \mathsf{c}) \end{array}\right]$$

is negligible in λ.

We say that a TT-KEM is *perfectly* robust, if it is $(n - r)$-robust.

For simplicity, we only consider semantic security with static corruptions in this paper.

Definition 3.10 (Semantic Security). A TT-KEM Π is semantically secure if for all PPT adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$, all $n \in \mathsf{poly}(\lambda)$, all $r \in \mathcal{S}_n$, all $0 < t \le t(r)$, and all $\varepsilon \in (0, 1/2]$ it holds that

$$\Pr[\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi}_{\mathcal{A}}(1^\lambda, n, r, t, \varepsilon) = 1] \le \frac{1}{2} + \mathsf{negl}(\lambda),$$

where $\mathsf{Game\text{-}IND\text{-}CPA}$ is defined in Figure 2.

$\mathsf{Game\text{-}IND\text{-}CPA}_{\mathcal{A}}(1^\lambda, n, r, t, \varepsilon)$

$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$

$(\mathsf{Cor}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{pp}, n, r, t, \varepsilon)$

$(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$

if $\mathsf{Cor} \not\subseteq [n] \ \vee |\mathsf{Cor}| \geq t$ **then return** 0

$(k_0, c) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk})$

$k_1 \xleftarrow{\$} \mathcal{K}_\lambda, b \xleftarrow{\$} \{0,1\}$

$b' \xleftarrow{\$} \mathcal{A}(\mathsf{st}, \mathsf{pk}, \mathsf{tk}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}}, k_b, c)$

return $b \stackrel{?}{=} b'$

Figure 2: Semantic security game of TT-KEMs

Definition 3.11 (Perfect TT-KEM). We say that a (traceable) threshold KEM is *perfect* if it is perfectly correct, perfectly robust, semantically secure, has no ramp, and $\mathcal{S}_n = [n]$.

4 Selling Decoders to Insiders – a Stronger Definition

In this section, we provide our definition of traceability against insiders. We begin by presenting the new definition in Section 4.1. Then, in Section 4.2 we discuss the design choices regarding the tracer's access to insiders' secret keys. Finally, in Section 4.3 we present an impossibility result for tracing full insiders without direct or indirect access by the tracer to the secret keys during tracing, and discuss in Section 4.3.1 an alternative definition in which the tracer *indirectly* accesses the secret keys.

4.1 The New Traceability Definition

First, we define the Trace algorithm to have the following syntax:

$\mathcal{T}' \xleftarrow{\$} \mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$. The tracing algorithm receives the public key pk, the tracing key tk, the limit ε, the set of full insiders $\mathcal{I}_{\mathsf{sk}}$, the set of partial insiders $\mathcal{I}_{\mathsf{pd}}$, the secret keys of all insiders,[9] as well as black-box access to a decoder algorithm D. It outputs a set of traitors $\mathcal{T}'$.

Consider Definition 4.1 and the corresponding experiment in Figure 3. Our definition generalizes and strengthens the existing definition due to [11] in the following ways:

- We parameterize the number of traitors that must be identified by a parameter τ, modifying the event GoodTr accordingly. The definition of [11] corresponds to the special case $\tau = 1$.
- We explicitly capture *security against insiders*. In Game-Tr, the adversary specifies two sets, $\mathcal{I}_{\mathsf{sk}}$ and $\mathcal{I}_{\mathsf{pd}}$, along with a decoder D. The set $\mathcal{I}_{\mathsf{sk}}$ consists of full insiders who provide their secret keys, while $\mathcal{I}_{\mathsf{pd}}$ consists of partial insiders who supply partial decryptions of a given ciphertext to the decoder. For a given scheme, the parameters σ and ρ denote the maximum number of permissible full insiders and partial insiders, respectively.
- The challenger runs the tracing algorithm on D, which makes use of the secret keys of both full and partial insiders, to determine the set of traitors found $\mathcal{T}'$.
- We refine the event GoodDec to reflect this notion of traceability. A decoder D is ε-good if it has an advantage ε when run with insider information from the specified sets $\mathcal{I}_{\mathsf{sk}}$ and $\mathcal{I}_{\mathsf{pd}}$.

[9] In fact, tracing may be possible even without the secret keys of the partial insiders (see Remark 4.5). Furthermore, an alternative definition for tracing full insiders is discussed in Section 4.3.1.

In GoodTr, the identified traitors may include insiders, but only if they are also traitors, i.e., if they were corrupted by the adversary via the $\mathcal{O}^{\mathsf{corrupt}}$ oracle. This models collusions where some traitors act as insiders to deflect blame.

- We optionally grant the adversary access to a partial decryption oracle for honestly generated ciphertexts $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$. Unlike the standard corruption oracle $\mathcal{O}^{\mathsf{corrupt}}$, which allows the adversary to adaptively corrupt *traitors*, this oracle models the scenario where colluding parties might observe partial decryptions of ciphertexts from honest parties (i.e., non-traitors). This captures a very realistic threat in certain scenarios (such as encrypted mempools where all committee members publish their partial decryptions every block), and essentially requires schemes to guarantee that learning partial decryptions from non-traitors does not allow colluders to frame honest parties. Additionally, it models that the adversary could decide on who to corrupt as traitor based on information leaked through partial decryptions. Unless stated otherwise, statements about traceability apply to traceability with and without $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ access.

Overall, if we say that a scheme is τ-traceable against (σ, ρ)-insiders, informally speaking, this means that tracing a decoder that works for a set of at most σ full insiders and ρ partial insiders is guaranteed to identify at least τ traitors.

$\mathsf{Game\text{-}Tr}_{\mathcal{A},\varepsilon,\sigma,\rho}(1^\lambda, n, r, t)$

$\mathcal{T} \leftarrow \emptyset$

$\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$

$(\mathsf{pk}, \{\mathsf{sk}_i\}_{[n]}, \mathsf{tk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$

$(D, (\mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}})) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{corrupt}}, \mathcal{O}^{\mathsf{PDec\text{-}Hon}}}(1^\lambda, \mathsf{pk})$

assert $\mathcal{I}_{\mathsf{sk}} \cap \mathcal{I}_{\mathsf{pd}} = \emptyset \wedge |\mathcal{I}_{\mathsf{sk}}| \leq \sigma \wedge |\mathcal{I}_{\mathsf{pd}}| \leq \rho$

$\mathcal{T}' \xleftarrow{\$} \mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}} \cup \mathcal{I}_{\mathsf{pd}}})$

return $(\mathsf{pk}, D, \mathcal{T}, \mathcal{T}', \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}} \cup \mathcal{I}_{\mathsf{pd}}})$

$\mathcal{O}^{\mathsf{corrupt}}(i)$

if $i \notin [n]$ **return** $\bot$

Update $\mathcal{T} \leftarrow \mathcal{T} \cup \{i\}$

return sk_i

$\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}})$

if $r_{\mathsf{Enc}} \notin \mathcal{R}_{\mathsf{Enc}}$ **return** $\bot$

$(k, c) \leftarrow \mathsf{Enc}(\mathsf{pk}; r_{\mathsf{Enc}})$

return $d_i \leftarrow \mathsf{PDec}(\mathsf{sk}_i, c)$

Figure 3: Traceability game for $\mathsf{TT\text{-}KEM}$. Changes accommodating insider attacks are marked in gray. Note that, optionally, the adversary has access to the $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle. If the assertion fails, the game sets D to the Dummy decoder that always outputs 0.

Definition 4.1 (τ-Traceability against (σ, ρ)-Insiders with Limit ε). Let $\varepsilon(\lambda) \in (0, 1/2]$, $\tau(\lambda)$, $\sigma(\lambda)$, and $\rho(\lambda)$ be functions in the security parameter. A $\mathsf{TT\text{-}KEM}$ scheme is τ-traceable against (σ, ρ)-insiders, if for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$, all $n \in \mathsf{poly}(\lambda)$, all $r \in \mathcal{S}_n$, all t with $0 < t \leq t(r)$ and $\tau + \sigma + \rho \leq t$

$$\Pr[\mathsf{GoodTr}] \geq \Pr[\mathsf{GoodDec}] - \mathsf{negl}(\lambda) \quad \text{and} \quad \Pr[\mathsf{BadTr}] \leq \mathsf{negl}(\lambda),$$

where $(\mathsf{pk}, D, \mathcal{T}, \mathcal{T}', \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i \in \mathcal{I}}) \xleftarrow{\$} \mathsf{Game\text{-}Tr}_{\mathcal{A},\varepsilon,\sigma,\rho}(1^\lambda, n, r, t)$ (Fig. 3), and the three events GoodTr, BadTr and $\mathsf{GoodDec}$ are defined as follows:

- GoodTr occurs if $|\mathcal{T}'| \geq \tau$ and $\mathcal{T}' \subseteq \mathcal{T}$, which indicates that at least τ real traitors were traced and no non-traitors were framed by the tracing algorithm. In particular, this means that insiders may only be framed if they are originally corrupted by the adversary (i.e., if they are also in $\mathcal{T}$).

- BadTr occurs if at least one non-traitor was framed in the tracing algorithm ($\mathcal{T}' \neq \emptyset$ and $\mathcal{T}' \not\subseteq \mathcal{T}$).

- $\mathsf{GoodDec}$ indicates that the decoder is good for pk, when given the secret keys of parties in $\mathcal{I}_{\mathsf{sk}}$ and partial decryptions on behalf of parties in $\mathcal{I}_{\mathsf{pd}}$. More formally, when $P(D) \geq 1/2 + \varepsilon(\lambda)$ where

$$P(D) := \Pr\left[D\Big(c, k_b, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big) = b : \begin{array}{c} (c, k_0) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}) \\ \forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{PDec}(\mathsf{sk}_i, c) \\ k_1 \xleftarrow{\$} \mathcal{K}_\lambda, b \xleftarrow{\$} \{0,1\} \end{array} \right].$$

  In particular, $P(\mathsf{Dummy}) = 1/2$, hence if the assertion in $\mathsf{Game\text{-}Tr}$ fails, then $\mathsf{GoodDec} = 0$.

$\mathsf{TestDec}^{D}_{\Pi}\Big(\mathsf{pk}, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}}, Q, \varepsilon\Big)$

Initialize $ctr \leftarrow 0$ **and repeat** Q **times**
  $(k_0, c) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk})$; $k_1 \xleftarrow{\$} \mathcal{K}$
  $\forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c)$
  $b \xleftarrow{\$} \{0,1\}$
  $b' \xleftarrow{\$} D\big(k_b, c, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\big)$
  if $b' = b$ **then** $ctr \leftarrow ctr + 1$
if $ctr \geq Q \cdot (1/2 + \varepsilon)$ **then return** 1 **else return** 0

Figure 4: The $\mathsf{TestDec}$ algorithm to test a decoder D.

Testing Decoders. Throughout this paper, we frequently test decoders. As shorthand, we define the algorithm $\mathsf{TestDec}$ in Figure 4. This algorithm makes Q queries to a decoder D, supplying a KEM key k, ciphertext c, and secret keys and partial decryptions on behalf of full and partial insiders. It then counts the number of times that D correctly guesses whether k is embedded in c. It returns 1, if D guesses correctly in at least a fraction $1/2 + \varepsilon$ of trials, indicating that, at least empirically, the decoder D has advantage ε.

On public tracing. In this paper, we work in the *private tracing* setting, where tracing relies on a secret tracing key, as most threshold traitor tracing schemes in the literature support only private tracing [11, 3, 20, 28]. Insider resilience, however, is equally relevant in a public tracing setting, where the tracing key is public. In fact, public threshold tracing schemes such as [23] and [22] exhibit exactly the *unlocking mechanism* that makes threshold traitor tracing schemes vulnerable to insiders. Our definition can be straightforwardly amended to support public tracing by granting the adversary access to the tracing key tk. Moreover, none of our compilers inherently rely on the fact that the tracing key is kept secret, and can be securely instantiated with public tracing schemes, yielding public tracing schemes as a result.

Limtless. Finally, we define the term *limitless*, referring to $\mathsf{TT\text{-}KEM}$s that can trace decoders with arbitrary but noticeable decryption advantage.

Definition 4.2 (Limitless). We say that a $\mathsf{TT\text{-}KEM}$ is *limitless*, if Definition 4.1 applies for any $\varepsilon \in 1/\mathsf{poly}(\lambda)$ and the KGen algorithm does not rely on ε.

In [54], Zhandry showed how to compile any *limited* traitor tracing scheme to a *limitless* scheme. This result was extended in [28] to the traceable threshold encryption setting. We further extend Zhandry's limit elimination compiler to support multi-traitor tracing and insiders in Appendix C.

4.2 Remarks on Giving the Tracer Access to Insiders' Secret Keys

Observe that we grant the tracer access to the full-insiders' secret keys during tracing (see Section 4.3 for details on this decision). We also grant the tracer access to the *partial insiders'* secret keys, thereby viewing *partial insiders as a relaxation of full insiders*. Under this choice, full-insider resilience implies partial-insider resilience (Theorem 5.3). While one could instead require the tracer to simulate partial decryptions for partial insiders without their secret keys, it is unclear whether such a model preserves this implication, since we would be removing information from the tracer (i.e., the partial insiders' secret keys) that may have been crucial for tracing to succeed with full insiders in the first place.

Remark 4.3 (On the adversary outputting the set of insiders). In our model, the adversary outputs the sets of insiders $\mathcal{I}_{\mathsf{sk}}$ and $\mathcal{I}_{\mathsf{pd}}$, along with the decoder D. This models the following scenario in the real world:

If a set of colluders decides to sell a decoder to a set of one or more insiders, we argue that at least the set of insiders themselves must know who the insiders are to make the decoder work (i.e., if they do not know whose keys to input into the decoder to make it work, it is essentially useless and has no decryption advantage). Since they are the ones who "blow the whistle" and forward the decoder to the tracer, they forward the set of insiders together with the decoder.

Another way of seeing this is as follows: If any set of less than t committee members can make a decoder work (such that it has a noticeable decryption advantage) by inputting their secret keys or partial decryptions, then, by definition, they are a valid set of insiders and could blow the whistle by forwarding the decoder to the tracer. A similar logic was used in a recent work [43] in the collusion-resilient secret sharing setting, where decoders come with a "user manual" (i.e., the set of parties from which additional shares are required).

Remark 4.4 (Requiring insiders' secret keys is justified). We also argue that it is *justified* in practice to require insiders' secret keys for tracing, since insiders already act as whistleblowers. Hence, they have a vested interest in tracing to succeed, so they can be assumed to forward their secret keys to the tracer along with the decoder. Alternatively, one could give the tracer the authority to force committee members to decommit their secret keys once a good decoder is found, since secrecy is broken at this point anyway.[10]

Remark 4.5 (Tracing is possible without partial insiders' secret keys) We note that it is not strictly necessary to require that the tracer also gets access to the *partial insiders'* secret keys. In fact, our maximally partial-insider resilient construction in Section 9 allows the tracer to simulate partial decryptions to the decoder, provided he knows the randomness used to generate the ciphertext, and thus, it does not need partial insiders' secret keys for tracing.

Nevertheless, in this work, we treat partial insiders as a relaxation of full insiders, which we consider the more natural perspective. To prove that full-insider resilience generically implies partial-insider resilience, the tracer must be given access to partial insiders' secret keys. This is because we proved that full insiders' secret keys are *necessary* for tracing. Consequently, if resilience against full insiders is to imply resilience against an equal number of partial insiders, the tracer must retain access to the partial insiders' secret keys. Indeed, resilience against full insiders may inherently rely on the tracer's access to the corresponding secret keys. Removing this information in the partial-insider setting could eliminate essential information for tracing to succeed.

4.3 Full Insider Tracing without Involving Secret Keys while Tracing is Impossible

In this section, we show that tracing in the presence of insiders is impossible without involving the secret keys of full insiders $\{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}$. To capture the notion of "no secret key involvement",

[10] This could be realized, e.g., by forcing committee members to lock a stake that is slashed upon non-compliance.

we consider a variant of Definition 4.1 and its security game Game-Tr (Figure 3), where the Trace^D algorithm no longer receives access to the insiders' secret keys. We sketch the security game for "traceability against full insiders without involving their secret keys" in Figure 5.

$\mathsf{Game\text{-}Tr\text{-}NoSk}_{\mathcal{A},\varepsilon,\sigma}(1^\lambda, n, r, t)$	$\mathcal{O}^{\mathsf{corrupt}}(i)$
$\mathcal{T} \leftarrow \emptyset$	**if** $i \notin [n]$ **return** $\bot$
$(\mathsf{pk}, \{\mathsf{sk}_i\}_{[n]}, \mathsf{tk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$	**Update** $\mathcal{T} \leftarrow \mathcal{T} \cup \{i\}$
$(D, \mathcal{I}_{\mathsf{sk}}) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{corrupt}}}(1^\lambda, \mathsf{pk})$	**return** sk_i
assert $\lvert\mathcal{I}_{\mathsf{sk}}\rvert \leq \sigma$	
$\mathcal{T}' \xleftarrow{\$} \mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}})$	
return $(\mathsf{pk}, D, \mathcal{T}, \mathcal{T}', \mathcal{I}_{\mathsf{sk}})$	

Figure 5: Traceability against full insiders without involving insiders' secret keys during tracing

Theorem 4.6 *There does not exist a* TT-KEM *that is both semantically secure and* 1*-traceable against* 1 *full insider without involving the full insider's secret key in the tracing process.*

For simplicity, we prove the theorem for a perfect TT-KEM, i.e., a TT-KEM with perfect correctness and no ramp.

Proof Idea. We construct an adversary $\mathcal{A}$ against 1-traceability that corrupts $t-1$ parties, designates one additional party p_{ins} as full insider, and produces a good decoder. Thus, by assumption, the decoder it produces must be traced successfully with overwhelming probability. On any given input $(k, c, \mathsf{sk}_{p_{\mathsf{ins}}})$, the decoder produced by $\mathcal{A}$ first checks whether the secret key of the full insider is "valid" by checking whether it can be used to decrypt a new random ciphertext (alongside the traitors' secret keys). If this check fails, i.e., the secret key is invalid, the decoder always outputs 0 instead.

Then, we argue as follows: First, a decoder that *always* outputs 0 cannot be traced, except with negligible probability due to unframability. Hence, the tracer must make at least one query to the decoder, where the given secret key passes the validity test.

Thus, we can use $\mathcal{A}$ to break semantic security of Π by internally running the tracer on D using the tracing key (which is known to the adversary in the semantic security game). Assuming the tracer makes a maximum of $Q \in \mathsf{poly}(\lambda)$ queries to D, $\mathcal{A}$ picks a random query to embed the IND-CPA challenge (k^*, c^*) into the validity check, and outputs the result to its challenger. $\mathcal{A}$ has probability $1/Q$ to hit the valid query by the tracer, in which case it will win the semantic security game. We next formalize this approach.

Proof of Theorem 4.6. We prove the theorem by assuming that Π is a perfect TT-KEM that is 1-traceable against 1 full insider without involving the insiders' secret key in the tracing process.

Fix an arbitrary $\varepsilon \in (0, 1/2]$. Let $\mathcal{A}$ be the following PPT adversary against $\mathsf{Game\text{-}Tr\text{-}NoSk}^{\Pi}$: $\mathcal{A}$ initially corrupts $t-1$ random parties $\mathcal{T} = \{p_1, \ldots, p_{t-1}\}$ as traitors and designates another random party p_{ins} as full insider. Then, $\mathcal{A}$ constructs a decoder D as follows: On input $(k, c, \mathsf{sk}_{p_{\mathsf{ins}}})$, the decoder samples $(k', c') \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk})$ and computes partial decryptions $d'_i \leftarrow \mathsf{PDec}(\mathsf{sk}, c')$ for all traitors $i \in \mathcal{T}$. Then it computes a partial decryption on behalf of the insider: $d'_{p_{\mathsf{ins}}} \leftarrow \mathsf{PDec}(\mathsf{sk}_{p_{\mathsf{ins}}}, c')$. Next, the decoder checks whether

$$\mathsf{valid} := \left[k' \stackrel{?}{=} \mathsf{Combine}\Big(\mathsf{pk}, \mathcal{T} \cup \{p_{\mathsf{ins}}\}, \{d'_i\}_{i \in \mathcal{T} \cup \{p_{\mathsf{ins}}\}}, c'\Big)\right].$$

Denote with valid the event that the above check succeeds. If $\mathsf{valid} = 1$, behave like a good decoder: Decrypt c using the traitors' secret keys alongside $\mathsf{sk}_{p_{\mathsf{ins}}}$, output 0 if the resulting key is

equal to k and 1 otherwise. Else, if $\mathsf{valid} = 0$, always output 0, essentially acting like the Dummy decoder. Finally, $\mathcal{A}$ outputs $(D, \mathcal{I}_{\mathsf{sk}} \coloneqq \{p_{\mathsf{ins}}\})$ to the $\mathsf{Game\text{-}Tr\text{-}NoSk}$ challenger.

We next prove two claims that will allow us to show that $\mathcal{A}$ can be used to break semantic security.

Claim 4.7 It holds that $\Pr[\mathsf{GoodDec}_{\mathcal{A}}] = 1$.

Proof of Claim 4.7. First, observe that $\sigma = 1 = |\mathcal{I}_{\mathsf{sk}}|$, hence the assertion in $\mathsf{Game\text{-}Tr\text{-}NoSk}$ does not fail, and $\mathsf{GoodDec}$ is not set to 0 by default. Due to perfect correctness of Π, it must hold that whenever D is queried with a valid secret key $\mathsf{sk}_{p_{\mathsf{ins}}}$, then $\mathsf{valid} = 1$, and thus, again by perfect correctness, D guesses correctly whether k is real or random. Hence, it holds that $P(D) = 1 \geq 1/2 + \varepsilon$ for any $\varepsilon \in (0, 1/2]$. ∎

Claim 4.8 Let $Q \in \mathsf{poly}(\lambda)$ be an upper bound on the number of queries to the decoder made by the Trace algorithm. When running $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}})$, it holds that $\mathsf{valid} = 1$ for at least one query made to D, except with negligible probability.

Proof of Claim 4.8. We prove the claim by contradiction. Let γ be the probability that $\mathsf{valid} = 0$ across all of the queries made by Trace^D, and assume that γ is non-negligible. Observe that, if no valid queries are made, then D always outputs 0, and thus acts like a dummy decoder. Also, note that by Claim 4.7 and the assumption that Π is traceable, it must hold that $\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq 1 - \mathsf{negl}(\lambda)$ for a negligible function $\mathsf{negl}(\lambda)$, which implies that

$$\Pr\left[\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}) \neq \emptyset \colon \begin{array}{c} \left(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}\right) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \varepsilon) \\ \mathcal{I}_{\mathsf{sk}} \xleftarrow{\$} \binom{[n]}{1}; \quad \mathcal{T} \xleftarrow{\$} \binom{[n]\setminus\mathcal{I}_{\mathsf{sk}}}{t-1} \\ \text{Construct } D \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Together with the assumption that D acts like a dummy decoder with probability γ, we can conclude that

$$\Pr\left[\mathsf{Trace}^{\mathsf{Dummy}}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}) \neq \emptyset \colon \begin{array}{c} \left(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}\right) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \varepsilon) \\ \mathcal{I}_{\mathsf{sk}} \xleftarrow{\$} \binom{[n]}{1} \end{array}\right] \geq \gamma - \mathsf{negl}(\lambda),$$

which is a contradiction with unframability of Π. To see this, consider an adversary $\mathcal{B}$ that corrupts no party and outputs the dummy decoder Dummy alongside a random full insider. Then $\Pr[\mathsf{BadTr}_{\mathcal{B}}] \geq \gamma - \mathsf{negl}(\lambda)$, which we initially assumed to be non-negligible. ∎

Equipped with Claims 4.7 and 4.8, we now show that we can construct a successful adversary $\mathcal{B}$ against $\mathsf{Game\text{-}IND\text{-}CPA}$ of Π.

First, $\mathcal{B}$ samples the set $\mathsf{Cor} \xleftarrow{\$} \binom{[n]}{t-1}$, forwards it to its challenger, and receives as a response the tuple $(\mathsf{pk}, \mathsf{tk}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}}, k^*, c^*)$. Then, $\mathcal{B}$ samples $\mathcal{I}_{\mathsf{sk}} \xleftarrow{\$} \binom{[n]\setminus\mathsf{Cor}}{1}$, and constructs the decoder D for $\mathcal{T} \coloneqq \mathsf{Cor}$ and $\mathcal{I}_{\mathsf{sk}}$. Finally, $\mathcal{B}$ chooses a random $q \in Q$ and, using tk, runs $\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}})$, simulating the behavior of D perfectly up until the q-th query. In the q-th query, $\mathcal{B}$ embeds the IND-CPA challenge (k^*, c^*) into the validity check: $\mathcal{B}$ computes $d_i \leftarrow \mathsf{PDec}(\mathsf{sk}_i, c^*)$ for all $i \in \mathcal{T}$ as well as $d_{p_{\mathsf{ins}}} \leftarrow \mathsf{PDec}(\mathsf{sk}_{p_{\mathsf{ins}}}, c^*)$, where $\mathsf{sk}_{p_{\mathsf{ins}}}$ is parsed from the input of the q-th query to D. Finally, $\mathcal{B}$ outputs 0 if $k^* = \mathsf{Combine}(\mathsf{pk}, \mathcal{T} \cup \{p_{\mathsf{ins}}\}, \{d_i\}_{i\in\mathcal{T}\cup\{p_{\mathsf{ins}}\}}, c^*)$, indicating that the challenge key k^* is real, and 1 otherwise. If tracing issues less than q queries, $\mathcal{B}$ instead outputs a random bit.

Analysis. Observe that if tracing issues at least one query such that $\mathsf{valid} = 1$, then $\mathcal{B}$ guesses this query with probability at least $1/Q$. Applying Claim 4.8, we can conclude that

$$\Pr\left[\mathsf{Game\text{-}IND\text{-}CPA}_{\mathcal{B}}(1^\lambda, n, t, \varepsilon) = 1\right] \geq \frac{1}{2} + \frac{1 - \mathsf{negl}(\lambda)}{2Q} - \mathsf{negl}(\lambda),$$

and thus $\mathcal{B}$ breaks IND-CPA security with a non-negligible advantage. ∎

4.3.1 Implications of Theorem 4.6 on the Traceability Definition

In order to properly define and realize traceability against full insiders, we must somehow circumvent the impossibility laid out in Theorem 4.6. This can be done in several ways, all of which essentially involve the full insiders' secret keys in the tracing procedure. We next compare two possible definitions of traceability against full insiders that avoid Theorem 4.6 by making use of the full insiders' secret keys during tracing. (In the following, we assume there are no partial insiders for simplicity.)

- **Insiders complete the decoder.** After the adversary in the traceability game outputs $(D, \mathcal{I}_{\mathsf{sk}})$, the *challenger* (i.e., the security game) could complete the decoder by wrapping it with the full insiders' secret keys. More formally, the security game could construct a decoder D' that, on input (k, c) queries $D(k, c, \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}})$ and outputs the same bit. Thus, the security game would essentially *hardcode* the full insider's secret keys into the decoder. The decoder D' is then traced by the tracer without access to secret keys. In practice, this roughly models the fact that full insiders could build a new decoder by hardcoding their secret keys in the same manner, and the forward D' to the tracer. This circumvents the impossibility result outlined above, since the adversary $\mathcal{B}$ cannot run the tracing algorithm on the decoder D directly. Claims 4.7 and 4.8 would only apply to the wrapped decoder D', which $\mathcal{B}$ cannot simulate when playing against Game-IND-CPA without knowledge of $\{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}}$. We highlight that the main advantage of such a model is that the tracer never directly learns the secret keys of the full insiders. However, we judge this benefit to be fairly limited, since the scheme's security must already be considered compromised once a good decoder exists, and thus, we do not gain much by preventing the tracer from learning some secret keys *during* tracing.

- **Tracer gets access to secret keys.** Instead, one could just give the tracer direct access to the secret keys during tracing, such that the tracer can complete D by itself. This also circumvents the impossibility result, since the adversary $\mathcal{B}$ now cannot simulate tracing D, as the tracing algorithm may now inherently rely on secret keys $\{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}}$ which are unknown to $\mathcal{B}$ playing in Game-IND-CPA.

  The advantage of this definition is that it gives the tracer more power and thus may give us more leverage in designing insider-resilient threshold traitor tracing (i.e., it potentially admits more constructions). In fact, it is easy to see that this definition is implied by the one above: Instead of having the security game construct D', one could just use $\{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}}$ to construct D' in the very first step of the Trace algorithm and then continue to trace D'. The key difference is that this definition admits tracing schemes, where the tracer can play with the secret keys of the full insiders. For example, the tracer might query the decoder D using *malformed* secret keys on behalf of the full insiders, and observe the decoder's behavior to obtain useful information.

  We recall that constructing *natively* full-insider resilient threshold traitor tracing (i.e., without relying on multi-traitor tracing) remains an interesting open problem. Allowing the tracer to exploit secret keys fed to the decoder on behalf of the full insiders during tracing may constitute a useful ingredient for such constructions.

Given its more general nature (i.e., it admits more constructions), we chose to define insider-resilient threshold traitor tracing in accordance with the latter option, giving the tracer access to the insiders' secret keys (c.f. Definition 4.1).

5 Insider Resilience from Multi-Traitor Tracing

Next, we formalize our approach to traceability against insider collusion by utilizing schemes that enable *tracing multiple traitors*: First, we show that one can accept identifying x *fewer*

traitors as a trade-off to gain resilience against x *more full insiders* (Theorem 5.2). Second, we show that one can accept resilience against y *fewer full insiders*, to achieve resilience against y *more partial insiders* (Theorem 5.3). The following corollary summarizes this result.

Corollary 5.1 *Any* TT-KEM *that is* τ*-traceable against* (σ, ρ)*-insiders is also* $(\tau - x)$*-traceable against* $(\sigma + x - y, \rho + y)$*-insiders for all* $x, y \in \mathbb{N}$ *with* $0 \leq x \leq \tau$ *and* $0 \leq y \leq \sigma + x$.

In particular, Corollary 5.1 implies that any τ-traceable scheme *without* insider resilience is also 1-traceable against (σ, ρ)-insiders, provided that $\sigma + \rho < \tau$. Importantly, we show that one does not have to fix this tradeoff at the time of instantiation: The same scheme is traceable against all supported trade-offs among τ, σ, and ρ *at the same time*.

5.1 Multi-Traitor to Full Insiders

We next show the first step, i.e., trading traitors found for full insiders. Let Π be a TT-KEM that is τ-traceable against (σ, ρ)-insiders with $\tau > 1$. Our compiler allows us to tolerate up to $\tau - 1$ *additional* full insiders by hardcoding their secret keys into the decoder during tracing, and removing the additional full insiders from the set of identified traitors at the end. More generally, we show that such a scheme is also $(\tau - x)$-traceable against $(\sigma + x, \rho)$ for all $0 \leq x < \tau$. We denote $\tau' = \tau - x$ and $\sigma' = \sigma + x$.

Construction 1 (Mult-Traitor to Full Insiders)**.** Let Π be a TT-KEM that is τ-traceable against (σ, ρ)-insiders. We construct Π' which is identical to Π except for a modification to the Trace algorithm:
$\Pi'.\mathsf{Trace}^{D'}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}', \mathcal{I}_{\mathsf{pd}}', \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}' \cup \mathcal{I}_{\mathsf{pd}}'})$: If $|\mathcal{I}_{\mathsf{sk}}'| \geq \tau + \sigma$, abort and output $\mathcal{T}' = \emptyset$. Otherwise, proceed with the following steps:

- Sample the set $\mathcal{I}_{\mathsf{sk}}$ as an arbitrary subset of $\mathcal{I}_{\mathsf{sk}}'$ of size $\min(\sigma, |\mathcal{I}_{\mathsf{sk}}'|)$, i.e., $\mathcal{I}_{\mathsf{sk}} \subseteq \mathcal{I}_{\mathsf{sk}}'$ and $|\mathcal{I}_{\mathsf{sk}}| = \min(\sigma, |\mathcal{I}_{\mathsf{sk}}'|)$. Also, let the set of full insiders for Π' that are not insiders for Π be $\mathcal{N}_{\mathsf{sk}} \leftarrow \mathcal{I}_{\mathsf{sk}}' \setminus \mathcal{I}_{\mathsf{sk}}$, and let $\mathcal{I}_{\mathsf{pd}} \leftarrow \mathcal{I}_{\mathsf{pd}}'$.
- Construct a decoder D for Π. On input $(c, k, \{\overline{\mathsf{sk}_i}\}_{i \in \mathcal{I}_{\mathsf{sk}}}, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}})$, the decoder D outputs $D'(c, k, \{\overline{\mathsf{sk}_i}\}_{i \in \mathcal{I}_{\mathsf{sk}}} \cup \{\mathsf{sk}_i\}_{i \in \mathcal{N}_{\mathsf{sk}}}, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}})$.[11]
- Run $\mathcal{T}_{\mathsf{ins}}' \xleftarrow{\$} \Pi.\mathsf{Trace}^{D}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}} \cup \mathcal{I}_{\mathsf{pd}}})$, output $\mathcal{T}' \leftarrow \mathcal{T}_{\mathsf{ins}}' \setminus \mathcal{N}_{\mathsf{sk}}$.

Theorem 5.2 *If* Π *is* τ*-traceable against* (σ, ρ)*-insiders, then Construction 1 is* $(\tau - x)$*-traceable against* $(\sigma + x, \rho)$*-insiders for all* $0 \leq x \leq \tau$.

Since the construction does not rely on specifying x ahead of time, Theorem 5.2 implies that the tracing algorithm always returns the *maximum* number of traitors, i.e., we do not have to limit ourselves to a specific choice of τ' and σ' at the time of setup or key generation. As an example, suppose we instantiate our theorem with a scheme Π where $\sigma = 0$ and $\tau = t$. If the tracer gets access to a decoder that does not require full insiders, he will identify τ traitors. If, instead, the decoder requires, say, 2 full insiders, it will identify $\tau - 2$ traitors. We address this subtle detail in the following proof.

Proof of Theorem 5.2. Fix arbitrary, τ, σ, and ρ for an arbitrary base scheme Π such that $0 < \tau \leq t$ and $0 < \tau + \sigma + \rho \leq t$ and assume that Π is τ-traceable against (σ, ρ)-insiders. Let $\mathcal{A}$ be any PPT adversary against traceability of Π'.

In the following, we introduce a number of random variables for $\mathcal{A}$ running in Game-Tr. Denote with $\mathcal{T}_{\mathcal{A}}$ the random variable capturing the set of traitors corrupted by $\mathcal{A}$ through its corruption oracle in Game-Tr, and with $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ as well as $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$ the set of full and partial insiders

[11] Recall that the decoder D' expects to obtain secret keys on behalf of $\mathcal{I}_{\mathsf{sk}}'$. By essentially hardcoding the secret keys of $\mathcal{N}_{\mathsf{sk}}$, D now only expects to obtain secret keys from $\mathcal{I}_{\mathsf{sk}}$, which is of size at most σ and thus traceable by $\Pi.\mathsf{Trace}$.

output by $\mathcal{A}$. Denote $\mathcal{I}'_{\mathcal{A}} = \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \cup \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$. W.l.o.g., assume that $\mathcal{I}'_{\mathcal{A}} \subseteq [n]$, $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \cap \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}} = \emptyset$ and $|\mathcal{I}'_{\mathcal{A}}| < \sigma + \rho + \tau$. Further, require that $|\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}| \leq \rho$ as well as $|\mathcal{I}_{\mathsf{sk}}{}'| < \sigma + \tau$. Also, denote with D' the decoder produced by $\mathcal{A}$ and by $\mathcal{T}'_{\mathcal{A}}$ the set of traitors found when tracing D' against $\mathcal{I}'_{\mathcal{A}}$.

We aim to show that there exists a negligible function negl such that

$$\Pr\overbrace{\left[\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}} \ \wedge \ |\mathcal{T}'_{\mathcal{A}}| \geq \tau - \max\left(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0\right)\right]}^{\mathsf{GoodTr}_{\mathcal{A}} \text{ for arbitrary } 0 \leq x \leq \tau} \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda).$$

Intuitively, the above statement expresses the following guarantee: First, we must only find *real traitors* ($\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}}$), and second, we find at least $\tau - x$ traitors, whenever the adversary specifies $\sigma + x$ full insiders. If the adversary should specify less than σ full insiders, we identify τ traitors. The above must happen with probability at least $\Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$.

Additionally, we will show that

$$\Pr[\mathsf{BadTr}_{\mathcal{A}}] \leq \mathsf{negl}(\lambda).$$

We construct a reduction $\mathcal{B}$ for τ-traceability against (σ, ρ)-inisiders of Π as follows: When $\mathcal{B}$ receives a public key and public parameters from its challenger, it forwards those to $\mathcal{A}$. When $\mathcal{A}$ issues a corruption query, $\mathcal{B}$ forwards it to its challenger and returns the resulting secret key to $\mathcal{A}$. Similarly, $\mathcal{B}$ answers queries to $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ by querying its own challenger's $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle. As a result, when $\mathcal{A}$ outputs $(D', \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}, \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}})$, it holds that $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$.

Given a decoder D', $\mathcal{B}$ must produce a "working" decoder D to return to its Π challenger. However, $\mathcal{B}$ can pick at most σ secret-key insiders from $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ to act as insiders to forward to its own challenger. Hence, $\mathcal{B}$ must hardcode the remaining keys into D' to produce D. Since $|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| < \sigma + \tau$, $\mathcal{B}$ hardcodes up to $\tau - 1$ keys.

To do so, $\mathcal{B}$ sets $\mathcal{I}_{\mathsf{pd}\mathcal{B}} \leftarrow \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$ and samples an arbitrary subset $\mathcal{I}_{\mathsf{sk}\mathcal{B}}$ of $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ of size $\min(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}|, \sigma)$ (as mandated by $\Pi'.\mathsf{Trace}$). Also $\mathcal{B}$ computes the set $\mathcal{N}_{\mathsf{sk}} \leftarrow \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \setminus \mathcal{I}_{\mathsf{sk}\mathcal{B}}$. Note at this point that it must hold that $|\mathcal{N}_{\mathsf{sk}}| = \max(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0)$. Next, $\mathcal{B}$ issues a corruption query to $\mathcal{O}^{\mathsf{corrupt}}$ to its challenger on behalf of each party in $\mathcal{N}_{\mathsf{sk}}$, thereby learning their secret keys. Now it holds that $\mathcal{T}_{\mathcal{B}} = \mathcal{T}_{\mathcal{A}} \cup \mathcal{N}_{\mathsf{sk}}$. Set $\mathcal{I}_{\mathcal{B}} = \mathcal{I}_{\mathsf{sk}\mathcal{B}} \cup \mathcal{I}_{\mathsf{pd}\mathcal{B}}$. Finally, $\mathcal{B}$ hardcodes $\{\mathsf{sk}_i\}_{i \in \mathcal{N}_{\mathsf{sk}}}$ into D' to produce D and outputs $(D, \mathcal{I}_{\mathsf{sk}\mathcal{B}}, \mathcal{I}_{\mathsf{pd}\mathcal{B}})$ to its challenger. Note that by construction it must hold that $|\mathcal{I}_{\mathsf{sk}\mathcal{B}}| \leq \sigma$ and $|\mathcal{I}_{\mathsf{pd}\mathcal{B}}| \leq \rho$. Denote with $\mathcal{T}'_{\mathcal{B}}$ the set of traitors as computed by $\mathcal{B}$'s challenger when running $\mathcal{T}'_{\mathcal{B}} \xleftarrow{\$} \Pi.\mathsf{Trace}^{D}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}\mathcal{B}}, \mathcal{I}_{\mathsf{pd}\mathcal{B}}, \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathcal{B}}})$. Then, it holds that $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}} \setminus \mathcal{N}_{\mathsf{sk}}$.

Analysis. First, it is easy to see that if D' is good for $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ and $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$, then D is good for $\mathcal{I}_{\mathsf{sk}\mathcal{B}}$ and $\mathcal{I}_{\mathsf{pd}\mathcal{B}}$ and thus $\Pr[\mathsf{GoodDec}_{\mathcal{A}}] = \Pr[\mathsf{GoodDec}_{\mathcal{B}}]$.

We start with BadTr: Observe that if $\mathsf{BadTr}_{\mathcal{A}}$ happens, then there must exist some party $i \in \mathcal{T}'_{\mathcal{A}}$ such that $i \notin \mathcal{T}_{\mathcal{A}}$. In this case, however, it must hold that $i \in \mathcal{T}'_{\mathcal{B}}$ and $i \notin \mathcal{N}_{\mathsf{sk}}$, because $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}} \setminus \mathcal{N}_{\mathsf{sk}}$. Since $\mathcal{T}_{\mathcal{B}} = \mathcal{T}_{\mathcal{A}} \cup \mathcal{N}_{\mathsf{sk}}$ and $i \notin \mathcal{T}_{\mathcal{A}}$ as well as $i \notin \mathcal{N}_{\mathsf{sk}}$, it must hold that $i \notin \mathcal{T}_{\mathcal{B}}$. In this case, the event $\mathsf{BadTr}_{\mathcal{B}}$ happens and thus

$$\Pr[\mathsf{BadTr}_{\mathcal{A}}] = \Pr[\mathsf{BadTr}_{\mathcal{B}}],$$

which is negligible by assumption.

Next, we look at GoodTr: Suppose that $\mathsf{GoodTr}_{\mathcal{B}}$ happens. In this case, it holds that $\mathcal{T}'_{\mathcal{B}} \subseteq \mathcal{T}_{\mathcal{B}}$. Also, by construction, for all $i \in \mathcal{T}'_{\mathcal{A}}$ it holds that $i \in \mathcal{T}'_{\mathcal{B}} \setminus \mathcal{N}_{\mathsf{sk}}$ and thus $i \in \mathcal{T}_{\mathcal{B}} \setminus \mathcal{N}_{\mathsf{sk}}$. Since $\mathcal{T}_{\mathcal{B}} = \mathcal{T}_{\mathcal{A}} \cup \mathcal{N}_{\mathsf{sk}}$, we get that $i \in (\mathcal{T}_{\mathcal{A}} \cup \mathcal{N}_{\mathsf{sk}}) \setminus \mathcal{N}_{\mathsf{sk}} = \mathcal{T}_{\mathcal{A}} \setminus \mathcal{N}_{\mathsf{sk}} \subseteq \mathcal{T}_{\mathcal{A}}$ and thus $\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}}$.

Finally, we will show that $|\mathcal{T}'_{\mathcal{A}}| \geq \tau - \max(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0)$. If $\mathsf{GoodTr}_{\mathcal{B}}$ happens, then it must hold that $|\mathcal{T}'_{\mathcal{B}}| \geq \tau$. Since $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}} \setminus \mathcal{N}_{\mathsf{sk}}$, we get that

$$|\mathcal{T}'_{\mathcal{A}}| \geq |\mathcal{T}'_{\mathcal{B}}| - |\mathcal{N}_{\mathsf{sk}}| = |\mathcal{T}'_{\mathcal{B}}| - \max(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0) \geq \tau - \max(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0).$$

We can conclude that

$$\begin{aligned}
&\Pr\left[\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}} \ \wedge \ |\mathcal{T}'_{\mathcal{A}}| \geq \tau - \max\left(|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| - \sigma, 0\right)\right] \\
&\geq \Pr[\mathsf{GoodTr}_{\mathcal{B}}] \\
&\geq \Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \mathsf{negl}(\lambda) \\
&= \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda).
\end{aligned}$$

■

5.2 Full-Insider Implies Partial-Insider Resilience

Additionally, we show that resilience against full insiders implies resilience against partial insiders, i.e., that any scheme that is τ-traceable against (σ, ρ) insiders is also τ-traceable against $(\sigma - x, \rho + x)$ for all x with $0 \leq x \leq \sigma$. In our compiler, if the tracer receives a decoder D' that works for $\rho + x$ partial insiders $\mathcal{I}_{\mathsf{pd}}{}'$, he designates a subset $\mathcal{I}_{\mathsf{pd}} \subseteq \mathcal{I}_{\mathsf{pd}}{}'$ of size ρ to be partial insiders against the base scheme, and the remaining x partial insiders as full insiders instead $\mathcal{I}_{\mathsf{sk}} = \mathcal{I}_{\mathsf{sk}}{}' \cup (\mathcal{I}_{\mathsf{pd}}{}' \setminus \mathcal{I}_{\mathsf{pd}})$. Then, he constructs a decoder D against the base scheme. When receiving partial decryptions on behalf of $\mathcal{I}_{\mathsf{pd}}$, and secret keys on behalf of $\mathcal{I}_{\mathsf{pd}}{}' \setminus \mathcal{I}_{\mathsf{pd}}$, D uses the given secret keys to compute partial decryptions on behalf of $\mathcal{I}_{\mathsf{pd}}{}' \setminus \mathcal{I}_{\mathsf{pd}}$, and runs D' with the secret keys of $\mathcal{I}_{\mathsf{sk}}{}'$, and the computed partial decryptions of $\mathcal{I}_{\mathsf{pd}}{}' \setminus \mathcal{I}_{\mathsf{pd}}$ alongside the given partial decryptions of $\mathcal{I}_{\mathsf{pd}}$.

Construction 2 (Full Insiders to Partial Insiders)**.** Let Π be a threshold KEM that is τ-traceable against (σ, ρ)-insiders. We construct Π' which is identical to Π except for a modification to the Trace algorithm:

$\Pi'.\mathsf{Trace}^{D'}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}{}', \mathcal{I}_{\mathsf{pd}}{}', \{\mathsf{sk}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}{}' \cup \mathcal{I}_{\mathsf{pd}}{}'})$**:** Let $I'_{\mathsf{sk}} = |\mathcal{I}_{\mathsf{sk}}{}'|$ and $I'_{\mathsf{pd}} = |\mathcal{I}_{\mathsf{pd}}{}'|$. Abort and output $\mathcal{T}' = \emptyset$ unless all of the following conditions are met:

- $\mathcal{I}_{\mathsf{sk}}{}' \cap \mathcal{I}_{\mathsf{pd}}{}' = \emptyset$
- $I'_{\mathsf{sk}} \leq \sigma$, i.e., there are at most a tolerable amount of full insiders.
- $I'_{\mathsf{sk}} + I'_{\mathsf{pd}} \leq \sigma + \rho$, i.e., let $x = \sigma - I'_{\mathsf{sk}}$ be the difference between how many full insiders we could tolerate and how many we received. Then, we can tolerate at most $I'_{\mathsf{pd}} \leq \rho + x$ partial insiders.

Otherwise, proceed with the following steps to sample appropriate sets of full insiders $\mathcal{I}_{\mathsf{sk}}$ and partial insiders $\mathcal{I}_{\mathsf{pd}}$ as well as construct a decoder D to trace using $\Pi.\mathsf{Trace}$:

- Sample the set $\mathcal{I}_{\mathsf{pd}}$ for Π as an arbitrary subset of size $\min(\rho, I'_{\mathsf{pd}})$ of the set $\mathcal{I}_{\mathsf{pd}}{}'$, i.e.,

  $$\mathcal{I}_{\mathsf{pd}} \subseteq \mathcal{I}_{\mathsf{pd}}{}' \text{ and } |\mathcal{I}_{\mathsf{pd}}| = \min(\rho, I'_{\mathsf{pd}}).$$

  Also, let the set of partial insiders for Π' that are not partial insiders for Π be $\mathcal{N}_{\mathsf{pd}} \leftarrow \mathcal{I}_{\mathsf{pd}}{}' \setminus \mathcal{I}_{\mathsf{pd}}$. These are essentially the set of partial insiders that are treated as full insiders.
- Set the set of full insiders $\mathcal{I}_{\mathsf{sk}} \leftarrow \mathcal{I}_{\mathsf{sk}}{}' \cup \mathcal{N}_{\mathsf{pd}}$.
- Construct a decoder D that given a KEM-key k, a ciphertext c, a set of secret keys $\{\overline{\mathsf{sk}}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}}$ from parties in $\mathcal{I}_{\mathsf{sk}}$ and partial decryptions $\{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}$ from parties $\mathcal{I}_{\mathsf{pd}}$ does the following:
  - Compute $\{d_i \leftarrow \mathsf{PDec}(\overline{\mathsf{sk}}_i, c)\}_{i \in \mathcal{N}_{\mathsf{pd}}}$, i.e., compute partial decryptions on behalf of the full insiders that are converted to partial insiders.
  - Output $D'\Big(c, k, \{\overline{\mathsf{sk}}_i\}_{i \in \mathcal{I}_{\mathsf{sk}}{}'}, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}{}'}\Big)$.

Finally, run and output $\mathcal{T}' \xleftarrow{\$} \Pi.\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$.

Theorem 5.3 *Let Π be a* TT-KEM *that is τ-traceable against (σ, ρ)-insiders. Then Construction 2 is τ traceable against $(\sigma - x, \rho + x)$ insiders for all $0 \le x \le \sigma$.*

The proof of Theorem 5.3 is similar to that of Theorem 5.2, and can be found in Appendix D.1.

6 Achieving K-traceability for Constant K

In this section, we present a compiler that allows us to identify a constant number of traitors K by partitioning parties into K bins.

We first introduce the following lemma, which we will use to argue the correctness and robustness of our compiler. Its proof is given in Appendix B.2.

Lemma 6.1 *Let $n = \mathsf{poly}(\lambda)$ and $K = o(n^a)$ for $a < 1$. Further, let $P_j = \{B_i^{(j)}\}_{i\in[K]}$ for $j \in [\ell]$ be a collection of ℓ independently and uniformly sampled (K, n)-balanced partitions, where $|B_i^{(j)}| = n/K$ for every $i \in [K]$ and every $j \in [\ell]$. Let $t_{\mathsf{bin}} = \frac{n}{2K} - \sqrt{\frac{\lambda n}{K}}$. Then,*

1. *For any $\ell \ge 1$, for uniformly distributed $S \subseteq [n]$ where $|S| \ge n/2$, there exists some $j \in [\ell]$ such that for all $i \in [K]$,*
$$|S \cap B_i^{(j)}| \;\ge\; t_{\mathsf{bin}},$$
except with negligible probability in λ.

2. *Let $r = n/K - t_{\mathsf{bin}}$, $\mathbf{r} = r \cdot (1, \ldots, 1) \in \mathbb{N}^K$, and $c \in (0, 1/2)$. Then, for $\ell \ge 2(1-c)K/c^2$, the probability that there is a subset $S \subseteq [n]$, where $|S| \le n/2 - cn$, which $(\ell, \mathbf{r})$-blocks $\{P_j\}_{j\in[\ell]}$ is negligible in λ.*

6.1 The 1-Parition Compiler

Construction 3 (1-Partition Compiler)**.** Let Π be a perfect τ-traceable threshold KEM against ρ partial insiders for key space $\mathcal{K} = \{0,1\}^\lambda$. Let $0 < \delta < 1$. Then, Π' with reconstruction threshold $\mathcal{S}_n = \{n/2\}$ and security threshold $t(r) = (1-\delta)r = (1-\delta)n/2$ is constructed as follows.

$\Pi'.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$**:** Sample a (K, n)-balanced partition $P = \{B_1, \ldots, B_K\}$ of parties in $[n]$ uniformly at random. Then, run the key generation algorithm of Π in each bin with threshold $t_{\mathsf{bin}} = (1-\delta)n/2K$:

$$\left\{\left(\mathsf{pk}_j, \{\mathsf{sk}_i\}_{i\in[B_j]}, \mathsf{tk}_j\right) \xleftarrow{\$} \Pi.\mathsf{KGen}\left(1^\lambda, B_j, t_{\mathsf{bin}}, \varepsilon/4\right)\right\}_{j\in[K]}$$

Output $\mathsf{pk} \coloneqq \{\mathsf{pk}_j\}_{j\in[K]}$, $\{\mathsf{sk}_i\}_{i\in[n]}$, and $\mathsf{tk} \coloneqq \{\mathsf{tk}_j\}_{j\in[K]}$.

$\Pi'.\mathsf{Enc}(\mathsf{pk})$**:** Run encryption $(k_j, c_j) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_j)$ in each bin $j \in [K]$. Compute $k \leftarrow \bigoplus_{j\in[K]} k_j$ and $c \leftarrow (c_1, \ldots, c_K)$. Output (k, c).

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i, (c_1, \ldots, c_K))$**:** Let $j = \mathsf{bin}(P, i)$ and output $d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_j)$.

$\Pi'.\mathsf{Combine}(\mathsf{pk}, S, \{d_i\}_{i\in[n]}, (c_1, \ldots, c_K))$**:** Given partial decryptions from a set of parties S, verify that $|S \cap B_j| \ge t_{\mathsf{bin}}$ for all $j \in [K]$. Then, recover $k_j \leftarrow \Pi.\mathsf{Combine}(S \cap B_j, \{d_i\}_{i\in S\cap B_j}, c_j)$ for all $j \in [K]$. Output $k \leftarrow \bigoplus_{j\in[K]} k_j$.

$\Pi'.\mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$**:** Let $\mathcal{I}_{\mathsf{pd}}{}^{(j)} = \mathcal{I}_{\mathsf{pd}} \cap B_j$ for all $j \in [K]$. Given black-box access to a decoder D we first test it by running $\mathsf{test} \leftarrow \mathsf{TestDec}(\mathsf{pk}, \emptyset, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}, Q := 8\lambda/\varepsilon^2, \varepsilon/2)$. (See Figure 4). If $\mathsf{test} = 1$, we say that D tests good. If $\mathsf{test} = 0$, we set all D_j to the Dummy decoder that always returns 0.

Otherwise, for all $j \in [K]$ where $\mathcal{I}_{\mathsf{pd}}{}^{(j)} \le \rho$, construct the decoder D_j as follows: On input the tuple $(k_j, c_j, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j)}})$, compute for all $\ell \in [K] \setminus \{j\}$:

$$(k_\ell, c_\ell) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_\ell) \text{ and } \forall i \in \mathcal{I}_{\mathsf{pd}}{}^{(\ell)} : d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_\ell).$$

Compute $k \leftarrow \bigoplus_{u\in[K]} k_u$ and return $D\Big(k, (c_1, \ldots, c_j, \ldots, c_K), \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)$ as D_j's output.

Finally, for all $j \in [K]$, where $\mathcal{I}_{\mathsf{pd}}{}^{(j)} \le \rho$, run $\mathcal{T}_j \xleftarrow{\$} \Pi.\mathsf{Trace}^{D_j(\cdot)}(\mathsf{pk}_j, \mathsf{tk}_j, \varepsilon/4, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j)}})$. For all $j \in [K]$ where $\mathcal{I}_{\mathsf{pd}}{}^{(j)} > \rho$, set $\mathcal{T}_j \leftarrow \emptyset$. Output $\mathcal{T} \leftarrow \bigcup_{j\in[K]} \mathcal{T}_j$.

Correctness, Roubstenss, and Semantic Security. We instantiate the construction with $\delta = 2\sqrt{\lambda K/n}$. Consider a random set $|S| \ge n/2$. We prove in Part 1 of Lemma 6.1 that S has at least t_{bin} parties from every one of the K bins. Thus, correctness follows. For robustenss, clearly, an adversary can only break robustness by causing sufficient crashes in at least one bin to break decryption. Since Π is assumed to be perfectly robust, any adversary must crash more than

$$\frac{n}{K} - t_{\mathsf{bin}} = \frac{n}{2K} \cdot (1 + \delta)$$

parties in any of the bins.

Finally, semantic security follows from the security of the underlying scheme Π. For completness, we provide a formal proof in Appendix D.2.2.

Parameter Sizes. If Π has ciphertext size $C(n)$, public key size $P(n)$, and secret key size $S(n)$, then Π' has sizes $P'(n) = K \cdot P(n/K)$, $S'(n) = S(n/K)$, and $C'(n) = K \cdot C(n/K)$. Π' has keyspace $\mathcal{K}_\lambda = \{0,1\}^\lambda$.

Remark 6.2 In Construction 3, we instantiate the underlying scheme in each bin with limit $\varepsilon/4$ to allow for a small margin of error when testing decoders. Essentially, in traceability proof, we need to claim that if D is ε-good against Π', then, by Chernoff's bound, D must test to be at least $(\varepsilon/2)$-good except with negligible probability, and if any decoder D *tests* at least $(\varepsilon/2)$-good, then the decoders D_j must be at least $(\varepsilon/4)$-good, which, again, holds by Chernoff's bound. Note that, if the underlying scheme is unable to satisfy this requirement due to a restriction on its limit, one can run the limit-elimination compiler (Theorem C.2) first.

Theorem 6.3 *Let $K \in \mathbb{N}$ be the number of bins used in Construction 3. Let $\tau > 0$ and $\tau + \rho \le t_{\mathsf{bin}}$. If the underlying* TT-KEM *Π is τ-traceable against ρ partial insiders then Π' (Construction 3) is $(\tau' = (K - x) \cdot \tau)$-traceable against $(\rho' = \rho + x \cdot (\rho + 1))$ partial insiders for all $0 \le x < K$. Whenever $\rho' > 0$, we require that Π offers a partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$. If Π offers a partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$, then so does Π'.*

A formal proof of Theorem 6.3 can be found in Appendix D.2.1.

6.2 Multi-Instance Expansion

In this section, we prove Theorem 2.5 as well as Corollary 2.6.

We start our analysis by showing that we can retain τ-traceability against ρ partial insiders of a scheme Π, when encrypting using a 1-out-of-ℓ threshold across ℓ instances, each instantiated with Π. Then, we show that applying this compiler to our 1-partition scheme allows us to make an improved robustness guarantee.

We note that this compiler follows the Zhandry's user expansion compiler [54], which allows for extending the number of users at the cost of increased ciphertext size in traceable broadcast encryption.

Construction 4. Let $\ell \in \mathsf{poly}(\lambda)$ and let Π be a τ-traceable TT-KEM against ρ partial insiders. We construct Π' as follows:

$\Pi'.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$**:** Run ℓ instances of $\Pi.\mathsf{KGen}$:

$$\left\{\left(\mathsf{pk}^{(j)}, \{\mathsf{sk}_i^{(j)}\}_{i\in[n]}, \mathsf{tk}^{(j)}\right) \xleftarrow{\$} \Pi.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon/4\ell)\right\}_{j\in[\ell]}$$

Set $\mathsf{pk} := \{\mathsf{pk}^{(j)}\}_{j\in[\ell]}$, $\mathsf{sk}_i := \{\mathsf{sk}_i^{(j)}\}_{j\in[\ell]}$, and $\mathsf{tk} := \{\mathsf{tk}^{(j)}\}_{j\in[\ell]}$. Output $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk})$.

$\Pi'.\mathsf{Enc}(\mathsf{pk})$**:** Sample a random KEM key $k \xleftarrow{\$} \mathcal{K}$, and run encryption in every instance to generate $(k_j, c_j) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^{(j)})$. Compute $u_j \leftarrow k \oplus k_j$ for all $j \in [\ell]$. Finally, assemble $c := \{(c_j, u_j)\}_{j\in[\ell]}$ and output (k, c).

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i, c)$**:** Compute $d_i^{(j)} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(j)}, c_j)$ for all instances $j \in [\ell]$ and output $d_i := \{d_i^{(j)}\}_{j\in[\ell]}$.

$\Pi'.\mathsf{Combine}(\mathsf{pk}, S, \{d_i\}_{i\in[n]}, c)$**:** Given partial decryptions from a set of parties in S, verify that there exists an instance $j \in [\ell]$ such that S is sufficient to decrypt in j. Then compute $k_j \leftarrow \Pi.\mathsf{Combine}(S, \{d_i^{(j)}\}_{i\in S}, c_j)$ for such j and output $k \leftarrow k_j \oplus u_j$.

$\Pi'.\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$**:** For each $j \in [\ell]$, construct a decoder D_j for the j-th instance of Π and run $\mathcal{T}'_j \xleftarrow{\$} \Pi.\mathsf{Trace}^{D_j}(\mathsf{pk}^{(j)}, \mathsf{tk}^{(j)}, \varepsilon/4\ell, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}}})$. Finally, output $\mathcal{T}' \leftarrow \bigcup_{j\in[\ell]} \mathcal{T}'_j$. Above, the decoder D_j is constructed from D as follows. On input $(c_j, k_j, \{d_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}}})$, D_j does the following:

- For $q \in [\ell]\backslash\{j\}$, generate $(c_q, k_q) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^{(q)})$ and compute $d_i^{(q)} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(q)}, c_q)$ for all $i \in \mathcal{I}_{\mathsf{pd}}$.
- Sample k^0 and k^1 uniformly from $\mathcal{K}$, as well as a random bit $b \xleftarrow{\$} \{0,1\}$.
- For $q < j$, sample $u_q \xleftarrow{\$} \mathcal{K}$, for $q \geq j$ (in particular, for $q = j$), set $u_q \leftarrow k_q \oplus k^0$.
- Assemble $c := \{(c_q, u_q)\}_{q\in[\ell]}$ and $d_i := \{d_i^{(q)}\}_{q\in[\ell]}$ for all $i \in \mathcal{I}_{\mathsf{pd}}$. The output of D_j is $b \oplus D\left(c, k^b, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\right)$.

Parameter Sizes: If Π has ciphetext size $C(n)$, public key size $P(n)$, and secret key size $S(n)$, then Π' has sizes $C'(n) = \ell \cdot C(n)$, $P'(n) = \ell \cdot P(n)$, and $S'(n) = \ell \cdot S(n)$.

Theorem 6.4 (Formal Version of Theroem 2.5). *If Π is τ-traceable against ρ partial insiders, then so is the multi-instance scheme Π'. Whenever $\rho > 0$, we require that Π offers a partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$. If Π offers a partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$, then so does Π'.*

It is intuitive to see why traceability is retained: Regarding unframability (i.e., BadTr), if a decoder manages to frame a non-traitor, then in at least one instance j, tracing must have returned a non-traitor on input D_j, hence a non-traitor has been framed in the base scheme Π. This can happen only with negligible probability due to unframability of Π. Similarly, if D is a good decoder, we can show that, à la PLBE, for at least one instance j, the constructed decoder D_j must also be a good decoder. As a result, the tracing in instance j on D_j will yield at least τ real traitors.

The proof of Theorem 6.4 follows closely the proof of user's expansion compiler [54], and can be found in Appendix D.3.

Proving Corollary 2.6. To arrive at Corollary 2.6, we apply multi-instance expansion (Theorem 2.5) to a 1-partition scheme (Theorem 2.2). It remains to show that the resulting scheme is robust.

Consider instantiating the 1-partition compiler scheme across K bins. Let Π be the resulting scheme.

Lemma 6.5 *For any constant $c \in (0, 1/2)$, let $\ell = 2(1-c)K/c^2$, let Π' be a multi-instance expansion (Construction 4) of Π across $\ell = 2(1-c)K/c^2$ partitions. Then, Π' is $(n/2-cn)$-robust.*

Proof. An adversary can break robustness only by causing decryption to fail in all ℓ partitions. This requires the adversary to identify a set $\mathsf{Cor} \subseteq [n]$ with $|\mathsf{Cor}| \leq n/2 - cn$ such that Cor is $(\ell, \mathbf{r})$-blocking where $\mathbf{r} = (n/K - t_{\mathsf{bin}}, \ldots, n/K - t_{\mathsf{bin}})$. By Lemma 6.1, the probability that such a set exists is negligible. This completes the proof. ∎

7 On $\Theta(t)$-Traceability

In this section, we prove Theorem 2.7. We start by presenting the following construction, which can be viewed as compiler that reduces the ciphertext from any t-traceable threshold KEM.

Construction 5. Let $\mathcal{K} = \{0,1\}^\lambda$, $\Pi_{\mathsf{t\text{-}tr}}$ be a perfect t-traceable threshold KEM, and TE any perfect (non-traceable) threshold KEM. The construction is parameterized by $\gamma \in (0, 1)$.

$\Pi'.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$**:** Set $K := \sqrt{n}$ and sample a uniformly random (K, n)-balanced partition $P = \{B_1, \ldots, B_K\}$ of $[n]$. Let $n_{\mathsf{bin}} := n/K = \sqrt{n}$ and $t_{\mathsf{bin}} := n_{\mathsf{bin}}/2 - \sqrt{\lambda n_{\mathsf{bin}}}$. Let $\varepsilon' = 1/2 - \frac{0.5-\varepsilon}{\gamma}$ For each $j \in [K]$, run

$$(\mathsf{pk}_1^{(j)}, \{\mathsf{sk}_{1,i}\}_{i \in B_j}, \mathsf{tk}_j) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{KGen}(1^\lambda, |B_j|, t_{\mathsf{bin}}, \varepsilon'/4),$$

$$(\mathsf{pk}_2^{(j)}, \{\mathsf{sk}_{2,i}\}_{i \in B_j}) \xleftarrow{\$} \mathsf{TE}.\mathsf{KGen}(1^\lambda, |B_j|, t_{\mathsf{bin}}).$$

Output $\mathsf{pk} := \{(\mathsf{pk}_1^{(j)}, \mathsf{pk}_2^{(j)})\}_j$, $\{\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i}\}_{i \in [n]}$, and $\mathsf{tk} := \{(\mathsf{tk}_j)\}_j$.

$\Pi'.\mathsf{Enc}(\mathsf{pk})$**:** Sample $u \xleftarrow{\$} [K]$. For $j \neq u$, compute $(k_j, c_j) \xleftarrow{\$} \mathsf{TE}.\mathsf{Enc}(\mathsf{pk}_2^{(j)})$; for $j = u$, compute $(k_u, c_u) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)})$. Set $k := \bigoplus_j k_j$ and $c := (c_1, \ldots, c_K)$. Return $(k, (c, u))$.

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i, (c, u))$**:** Parse sk_i as $(\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i})$. If $i \in B_u$, output $\Pi_{\mathsf{t\text{-}tr}}.\mathsf{PDec}(\mathsf{sk}_{1,i}, c_u)$; otherwise output $\mathsf{TE}.\mathsf{PDec}(\mathsf{sk}_{2,i}, c_j)$ where $j = \mathsf{bin}(P, i)$.

$\Pi'.\mathsf{Combine}(S, (c, u), \{d_i\}_{i \in S})$**:** Verify $|S \cap B_j| \geq t_{\mathsf{bin}}$ for all $j \in [K]$. For $j = u$, compute $k_u \leftarrow \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Combine}(S \cap B_u, \{d_i\}_{i \in S \cap B_u})$; for $j \neq u$, compute $k_j \leftarrow \mathsf{TE}.\mathsf{Combine}(S \cap B_j, \{d_i\}_{i \in S \cap B_j})$. Output $k := \bigoplus_j k_j$.

$\Pi'.\mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon)$**:** Let $\varepsilon' = 1/2 - \frac{0.5-\varepsilon}{\gamma}$. Given black-box access to a decoder D we do the following:

- Let $Q = 8\lambda/\varepsilon'^2$ and initialize a counter $ctr_u \leftarrow 0$ for every $u \in [K]$.
- For every $u \in [K]$:
  - Do Q times: Sample $(k_0, (c, u)) \xleftarrow{\$} \Pi'.\mathsf{Enc}(\mathsf{pk}; u)$ and $k_1 \xleftarrow{\$} \mathcal{K}$ as well as a random bit b. That is, the random position is fixed to u. Let $b' = D(k_b, (c, u))$. If $b = b'$, then $ctr_u = ctr_u + 1$.
  - Set $\mathsf{test}_u = 1$ if $ctr_u \geq Q \cdot (1/2 + \varepsilon'/2)$ and $\mathsf{test}_u = 0$ otherwise.

- Construct the decoder D_u as follows: On input (k_u, c_u), where k_u is a KEM key and c_u is a ciphertext in $\Pi_{\mathsf{t\text{-}tr}}$, compute for all $j \in [K] \setminus \{u\}$:

$$(k_j, c_j) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}_2^{(j)}, B_j, t_{\mathsf{bin}}).$$

  Compute $k \leftarrow \bigoplus_{j\in[K]} k_j$ and return $D(k, ((c_1, \ldots, c_u, \ldots, c_K), u))$ as D_u's output.

Finally, for all $j \in [K]$, where $\mathsf{test}_j = 1$, run $\mathcal{T}_j \xleftarrow{\$} \Pi.\mathsf{Trace}^{D_j(\cdot)}(\mathsf{pk}_j, \mathsf{tk}_j, \varepsilon'/4)$, and all $j \in [K]$ where $\mathsf{test}_j = 0$, set $\mathcal{T}_j \leftarrow \emptyset$. Output $\mathcal{T} \leftarrow \bigcup_{j\in[K]} \mathcal{T}_j$.

Theorem 7.1 *Let $\varepsilon, \varepsilon' \in (0, 0.5)$ such that $\varepsilon' < \varepsilon$ and $\gamma = \frac{0.5-\varepsilon}{0.5-\varepsilon'}$. If $\Pi_{\mathsf{t\text{-}tr}}$ is t_{bin}-traceable with limit $\varepsilon'/4$ then Π' is $(1-\gamma)Kt_{\mathsf{bin}}$-traceable with limit ε.*

The Proof of Theorem 7.1. The proof follows the argument of Theorem 6.3, with one key difference: we can no longer guarantee the existence of a decoder D_i for every bin i. Nevertheless, Lemma 7.2 shows that decoders can be extracted for at least $(1-\gamma)K$ of the bins. The theorem then follows by demonstrating that tracing succeeds for each of these decoders, using an argument similar to that in Theorem 6.3. The full proof is presented in Appendix D.4.

Before discussing Lemma 7.2, we introduce the following definition. We say that D is ε'-good for poisition u if it holds that:

$$P_{\Pi',u} \coloneqq \Pr[D(k^b, ((c_1, \ldots, c_K), u)) = b] \geq \frac{1}{2} + \varepsilon',$$

where $(k_u, c_u) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)}, B_u)$, $(k_j, c_j) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}_2^{(j)}, B_j)$ for all $j \in [K] \setminus \{u\}$, $k^0 \xleftarrow{\$} \bigoplus_{j=1}^{K} k_j$, $k^1 \xleftarrow{\$} \mathcal{K}$, and $b \xleftarrow{\$} \{0,1\}$.

Lemma 7.2 *Suppose that D is ε-good and let $\gamma = \frac{0.5-\varepsilon}{0.5-\varepsilon'}$. Then, for any $\varepsilon' < \varepsilon$ it holds that D is ε'-good for at least $(1-\gamma)\cdot K$ positions $j \in [K]$.*

Proof of Lemma 7.2. We first note that if D is ε-good, then it must hold that

$$P_{\Pi'}(D) = \Pr\left[D(k^b, c) = b \colon (k^0, c) \xleftarrow{\$} \Pi'.\mathsf{Enc}(\mathsf{pk}),\ k^1 \xleftarrow{\$} \mathcal{K},\ b \xleftarrow{\$} \{0,1\}\right] \geq \frac{1}{2} + \varepsilon.$$

Assume towards contradaction that D is ε'-good for less than $(1-\gamma)\cdot K$ indices, then it holds that:

$$P_{\Pi'}(D) < (1-\gamma)\cdot 1 + \gamma \cdot \left(\frac{1}{2} + \varepsilon'\right) = 1 - \gamma\left(\frac{1}{2} - \varepsilon'\right) = 1 - \frac{0.5-\varepsilon}{0.5-\varepsilon'}(0.5-\varepsilon') = \frac{1}{2} + \varepsilon.$$

This contradicts the assumption that $P_{\Pi'}(D) \geq \frac{1}{2} + \varepsilon$. ∎

Putting it all together. By Lemma 7.2, choosing $\varepsilon = 1/4$ and $\varepsilon' = 1/24$ ensures that D is ε'-good for at least $\frac{5}{11}\cdot K$ of the bins. Instantiating Theorem 7.1 with the t-traceable threshold KEM that has $O(n)$ ciphertexts (see Appendix E) and setting $K = \sqrt{n}$ yields a $5n/22$-traceable threshold KEM with ciphertext size $O(\sqrt{n})$ and $O(\sqrt{n})$-robustness. This scheme has a limit ε. We remove this limt by applying the generalized limit elimination compiler from Appendix C, thereby proving Theorem 2.7.

8 On $\Theta(n^a)$-Traceability for $a \in (0,1)$

In this section, we prove Theorem 2.8. That is, we show a compiler to achieve $\Theta(n^a)$-traceability. The compiler is depicted in Construction 6, whose properties are summarized in the following theorem.

Theorem 8.1 *Let $a \in (0,1)$ and $c_1, c_2 \in (0, 1/2)$. Further, let $\delta_1 \coloneqq 6n^{-a/2}$, $\delta_2 \coloneqq 6\sqrt{n}/(n-n^a)$, $t_1 \coloneqq (1-\delta_1)n^a/2$, and $t_2 \coloneqq (1-\delta_2)(n-n^a)/2$. Let*

$$\ell \coloneqq \max\left\{3\lambda n^{1-a}, \frac{10n^{1-a}(1-c_1)}{c_1^2}, \frac{20n^{1-a}(2-3c_2)}{c_2^2}\right\}, \qquad t_{\mathsf{parts}} \coloneqq 0.8\ell.$$

Let Π be a perfect limitless 1-traceable TT-KEM *that offers a $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ oracle*[12] *and let* TE *a perfect* T-KEM*. Then, Π' (Construction 6) is probabilistically correct, semantically secure, $(n/2 - c_1 n)$-robust, and $(0.3n^{1-a})$-traceable for reconstruction threshold $r = n/2$ and security threshold $t = t_1 + t_2 = n/2 - o(n)$. If Π is resilient against (t_1-1)-partial insiders and offers a $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle, then Π' is resilient against $(n/2-c_2 n)$ partial insiders and offers a $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle.*

Before introducing our unbalanced construction, we state a lemma that will be useful in proving its correctness, robustness, and traceability. A proof of the following Lemma can be found in Appendix B.3.

Lemma 8.2 *Let $P_j = \{B_1^j, B_2^j\}$ for $j \in [\ell]$ be a collection of ℓ independently and uniformly sampled $(2,n)$-unbalanced partitions, where for $a \in (0,1)$, $|B_1^j| = n^a$ and $|B_2^j| = n - n^a$ for every $j \in [\ell]$. Let $\delta_1 = 6n^{-a/2}$, $\delta_2 = 6\frac{\sqrt{n}}{n-n^a}$, $t_1 = (1-\delta_1)\frac{n^a}{2}$, $t_2 = (1-\delta_2)\frac{(n-n^a)}{2}$. Then,*

1. *If $\ell \geq 6\lambda$, for uniformly distributed $S \subseteq [n]$ where $|S| \geq n/2$, it holds that S $(0.8\ell, (t_1, t_2))$-dominates the partitions $\{P_j\}_{j\in[\ell]}$ with overwhelmning probability.*

2. *Let $c \in (0, 0.5)$, $\ell \geq 10n^{1-a}(1-c)/(c^2)$, $\mathbf{r}_1 = (n^a - t_1, n - n^a - t_2)$ and $\beta = 0.2\ell$. Then, the probability that there exists a subset $S \subseteq [n]$, such that $|S| \leq n/2 - cn$ and $(\beta, \mathbf{r}_1)$-blocks $\{P_j\}_{j\in[\ell]}$ is negligible in λ.*

3. *Let $c \in (0, 0.5)$, $\ell \geq 20n^{1-a}(2-3c)/c^2$, $\mathbf{r}_2 = (t_1, 0)$, and $\beta = 0.2\ell$. Then, the probability that there exists a subset $S \subseteq [n]$, such that $|S| \leq n/2 - cn$ and $(0.2\ell, \mathbf{r}_2)$-dominates $\{P_j\}_{j\in[\ell]}$ is negligible in λ.*

4. *If $\ell \geq 3n^{1-a}\lambda$, then the probability that there exists a $u \in [n]$ such that $u \in B_1^j$ for more than $2\ell/n^{1-a}$ indices j is negligible.*

Construction 6. Let $\ell \in \mathsf{poly}(\lambda)$ and $0 < t_{\mathsf{parts}} \leq \ell$. Let Π be a perfect 1-traceable TT-KEM against ρ partial insiders. Let TE be a perfect threshold KEM. Let the key-space of both Π and TE be $\{0,1\}^\lambda$. We construct Π', parameterized by a constant $a \in (0,1)$, and functions $\delta_1(\lambda, n, a)$ and $\delta_2(\lambda, n, a)$.

$\Pi'.\mathsf{KGen}(1^\lambda, n, r \coloneqq n/2, t, \varepsilon)$**:** For key generation, we sample ℓ random $(2,n)$-*unbalanced* partitions of $[n]$ parties $\{P_1, \ldots, P_\ell\}$ such that for all partitions $j \in [\ell]$ it holds that $P_j = \{B_1^{(j)}, B_2^{(j)}\}$, where $|B_1^{(j)}| = n^a$ and $|B_2^{(j)}| = n - n^a$. Let $t_1 \coloneqq \frac{n^a}{2} \cdot (1-\delta_1)$ and $t_2 \coloneqq \frac{n-n^a}{2} \cdot (1-\delta_2)$ and assert that $t \leq t_1 + t_2$. Let $\varepsilon' = \varepsilon/16\ell$. In each partition, run key generation of Π for parties in $B_1^{(j)}$ with threshold t_1 and in $B_2^{(j)}$ we run key generation of TE with threshold t_2:

$$\forall j \in [\ell]\colon (\Pi.\mathsf{pk}^{(j)}, \{\mathsf{sk}_i^{(j)}\}_{i\in B_1^{(j)}}, \mathsf{tk}^{(j)}) \xleftarrow{\$} \Pi.\mathsf{KGen}(1^\lambda, B_1^{(j)}, t_1, t_1, \varepsilon')$$

$$\forall j \in [\ell]\colon (\mathsf{TE}.\mathsf{pk}^{(j)}, \{\mathsf{sk}_i^{(j)}\}_{i\in B_2^{(j)}}) \xleftarrow{\$} \mathsf{TE}.\mathsf{KGen}(1^\lambda, B_2^{(j)}, t_2, t_2)$$

Set $\mathsf{pk} \coloneqq \{(\Pi.\mathsf{pk}^{(j)}, \mathsf{TE}.\mathsf{pk}^{(j)})\}_{j\in[\ell]}$, $\mathsf{sk}_i \coloneqq \{\mathsf{sk}_i^{(j)}\}_{j\in[\ell]}$, and $\mathsf{tk} \coloneqq \{\mathsf{tk}^{(j)}\}_{j\in[\ell]}$. Output the tuple $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk})$.

[12] The $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ oracle is defined in Section 8.1, Figure 7.

$\Pi'.\mathsf{Enc}(\mathsf{pk})$**:** Sample KEM key $k \xleftarrow{\$} \mathcal{K}$ and share $(s_1, \ldots, s_\ell) \xleftarrow{\$} \mathsf{Share}(k, \ell, t_{\mathsf{parts}})$. Next, for all $j \in [\ell]$, do the following:

- Generate $(c_j^{\Pi}, k_j^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(j)})$ and $(c_j^{\mathsf{TE}}, k_j^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE}.\mathsf{Enc}(\mathsf{TE}.\mathsf{pk}^{(j)})$ and assemble $c_j \coloneqq (c_j^{\Pi}, c_j^{\mathsf{TE}})$.
- Compute the mask $u_j \leftarrow s_j \oplus k_j^{\Pi} \oplus k_j^{\mathsf{TE}}$.

Finally, assemble and output $(k, c \coloneqq \{(c_j, u_j)\}_{j\in[\ell]})$.

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i, c)$**:** For all $j \in [\ell]$, do the following: If $i \in B_1^{(j)}$, then compute $d_i^{(j)} \xleftarrow{\$} \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_j^{\Pi})$, else, compute $d_i^{(j)} \xleftarrow{\$} \mathsf{TE}.\mathsf{PDec}(\mathsf{sk}_i, c_j^{\mathsf{TE}})$. Finally, assemble and output $d_i \xleftarrow{\$} \{d_i^{(j)}\}_{j\in[\ell]}$.

$\Pi'.\mathsf{Combine}(\mathsf{pk}, S, \{d_i\}_{i\in S}, c)$**:** Initialize $M = \emptyset$. For all $j \in [\ell]$, let $S_1^{(j)} = S \cap B_1^{(j)}$ and $S_2^{(j)} = S \cap B_2^{(j)}$ and do the following:

- If $|S_1^{(j)}| < t_1$ or $|S_2^{(j)}| < t_2$, continue with the next index.
- Otherwise, compute

$$k_j^{\Pi} \leftarrow \Pi.\mathsf{Combine}\left(S_1^{(j)}, \{d_i^{(j)}\}_{i\in S_1^{(j)}}, c_j^{\Pi}\right) \text{ and } k_j^{\mathsf{TE}} \leftarrow \mathsf{TE}.\mathsf{Combine}\left(S_2^{(j)}, \{d_i^{(j)}\}_{i\in S_2^{(j)}}, c_j^{\mathsf{TE}}\right).$$

- Undo the mask $s_j \leftarrow u_j \oplus k_j^{\Pi} \oplus k_j^{\mathsf{TE}}$.
- Insert $M \leftarrow M \cup \{j\}$.

Finally, if $|M| \geq t_{\mathsf{parts}}$, reconstruct and output $k \leftarrow \mathsf{Rec}(t_{\mathsf{parts}}, M, \{s_j\}_{j\in M})$. Return $\perp$ otherwise.

$\Pi'.\mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$**:** See Figure 6 (and explained in Section 8.1).

Parameter Sizes. Π' has ciphertext size $C'(n) = \ell \cdot (C_\Pi(n^a) + C_{\mathsf{TE}}(n - n^a))$, public key size $P'(n) = \ell \cdot (P_\Pi(n^a) + P_{\mathsf{TE}}(n - n^a))$, and secret key size $S'(n) = \ell \cdot \max(S_\Pi(n^a), S_{\mathsf{TE}}(n - n^a))$. Π' has key space $\mathcal{K}_\lambda = \{0,1\}^\lambda$.

Correctness and Security. We show in Lemma 8.2 that, using a Chernoff bound, when $\ell = \Omega(\lambda)$, a random subset S with $|S| \geq n/2$ (t_1, t_2)-dominates (Definition 3.5) at least 0.8ℓ of the partitions except with negligible probability, implying correctness. Semantic security follows straightforwardly from that of the underlying schemes. The formal proof is deferred to Appendix D.5.

Lemma 8.3 (Robustness of Construction 6). *If* Π *and* TE *are perfectly robust, construction* Π' *is* $(n/2 - c_1 n)$*-robust for arbitrary* $c_1 \in (0, 1/2)$*.*

Proof. An adversary can only break robustness by causing decryption to fail in at least $\ell - t_{\mathsf{parts}} = 0.2\ell$ partitions. This requires the adversary to find a set $\mathsf{Cor} \subseteq [n]$ with $|\mathsf{Cor}| \leq n/2 - c_1 n$ such that Cor $(0.2\ell, \mathbf{r})$-blocks the set $\{P_i\}_{i\in[\ell]}$ for $\mathbf{r} = (n^a - t_1, n - n^a - t_2)$.[13]

We show via a probabilistic argument that, for an appropriate choice of ℓ (specified in Theorem 8.1), the probability that such a set exists is negligible. Suppose for now the adversary fixes Cor ahead of time, then the expected number of parties from Cor that fall into the first bin of a given partition is $(\frac{1}{2} - c_1)n^a$. For Cor to block the first bin, it must cover more than $n^a/2$ parties in that bin, which requires a deviation of $\Theta(c_1 n^a)$ from the expectation. By a standard Chernoff bound, the probability of such a deviation for a fixed partition is $\Theta(\exp(-n^a))$. To account for an adversary who chooses Cor *after seeing the partitions*, we apply a union bound over all possible choices of Cor to show that the probability that there exists a set Cor for which this event occurs in at least 0.2ℓ partitions is negligible. For details, see Part 2 of Lemma 8.2. ■

[13] We consider only the first bin, as blocking decryption in the second bin requires $O(n - n^a)$ parties, which, for sufficiently large n, is larger than $n/2 - cn$ for any constant c.

8.1 The Tracing Algorithm

The tracing algorithm is depicted in Figure 6. We next give an overview. Let us fix $t_{\mathsf{parts}} = 0.8\ell$, i.e., during encryption, we share the KEM key across partitions in a 0.8ℓ-out-of-ℓ manner. Thus, intuitively, any successful decoder must decrypt in at least 0.8ℓ of the partitions, which will allow us to identify one traitor from the small bin of each such partition. While these traitors may not all be distinct, we later use a counting argument to bound the maximum number of duplicate traitors to achieve τ' *unique* traitors found as promised in Theorem 8.1. We elaborate on this in the proof later, and focus here on how to find a traitor from sufficiently many partitions.

Notation. For each step i, let $\mathsf{Ord}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,\ell})$ be an ordering over the partition indices such that $v_{i,j} \in [\ell]$, and $v_{i,j} \neq v_{i,q}$ for all $j \neq q$. For all $x \in [\ell]$, we define access to the x-th element $\mathsf{Ord}_i[x] = v_{i,x}$, and the prefix $\mathsf{Ord}_i[\leq x] = (v_{i,1}, v_{i,2}, \ldots, v_{i,x})$, and analogous notation for operators $<, \geq, >$.

We proceed in up to 0.8ℓ steps. In each step i, we run the subroutine TrStep, which outputs a set $\mathcal{T}_i'$ of traitors found in step i, and bookkeeping information Found_i, which denotes the *first* partition, from which a traitor was found.

Tracing in the i-th Step. At a high level, in TrStep we apply a PLBE-style tracing procedure similar to the multi-instance expansion compiler (Theorem 2.5), but with two crucial modifications: (1) we carefully control the *order in which partitions are revoked*, and (2) we make use of the fact that the key is shared in a 0.8ℓ-out-of-ℓ manner instead of 1-out-of-ℓ.

In step i, we revoke partitions in the order fixed in Ord_i. This is achieved by designing decoders in a manner similar to multi-instance expansion tracing (see Theorem 2.5). That is, we construct decoders D_j for $j \in [\ell]$ by revoking all partitions $\mathsf{Ord}_i[< pos_j]$, where pos_j is the position such that $j = \mathsf{Ord}_i[pos_j]$. Informally, to revoke a partition $q \in \mathsf{Ord}_i[< pos_j]$, we replace the u_q element of the ciphertext, which is supposed to mask the partition P_q's share of the KEM key k, with a uniform random value. Thus, decrypting in partition P_q can no longer contribute to decrypting the overall ciphertext.

Due to threshold secret sharing the KEM key k, the PLBE argument guarantees the existence of a good decoder among the first $0.2\ell + 1$ partitions, since decryption advantage drops to 0 already after revoking $0.2\ell + 1$ partitions. Thus, we will find 1 traitor from the first $0.2\ell + 1$ partitions, i.e., $\mathsf{Found}_i \in \mathsf{Ord}_i[\leq 0.2\ell + 1]$. In the tracing algorithm, we denote this cut-off position as $\mathsf{Cut} \leftarrow 0.2\ell + 1$. To ensure that we find a traitor *from a different partition* in step $i + 1$, we let Ord_{i+1} be the order after swapping the partition Found_i where a traitor was found to the trailing $0.8\ell - 1$ segment of the ordering. This procedure is repeated for a total of 0.8ℓ steps, such that we find a traitor from the first $0.2\ell + 1$ partitions in every step. By the end, all partitions where a traitor was found will occupy the trailing 0.8ℓ positions in the order of partitions.

Determining Ord_1 and Dealing with Partial Insiders. Next, we aim to leverage the fact that the underlying scheme Π is already resilient against $t_1 - 1$ partial insiders. Then, partitions that are t_1-dominated by partial insiders cannot be traced in TrStep. We denote such partitions as $\overline{\mathcal{P}} = \{j \in [\ell] \colon |B_1^{(j)} \cap \mathcal{I}_{\mathsf{pd}}| \geq t_1\}$. To ensure that we never trace decoders for partitions $\overline{\mathcal{P}}$ in TrStep, we set up Ord_1 such that $\overline{\mathcal{P}}$ occupies *the trailing positions*, i.e., $\overline{\mathcal{P}}$ is a suffix of Ord_1. This is done in the *Initialization* phase of $\Pi'.\mathsf{Trace}$. As a result, tracing now proceeds in $d = 0.8\ell - |\overline{\mathcal{P}}|$ steps, as $|\overline{\mathcal{P}}|$ of the trailing positions are already occupied.

Requiring an $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ Oracle. In one part of the traceability proof for Construction 6 (in particular, Lemma 8.6), we will argue that in each step i of the tracing algorithm, the tracer finds a partition $\mathsf{Found}_i \neq \bot$ and that the traitors identified from Found_i must be real traitors. To prove this, we execute a hybrid argument over the different steps. The reduction in the hybrid

$\Pi'.\mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$

$(\mathsf{test}, \mathsf{Ord}_1, d, \mathsf{end}_1, \mathsf{Cut}) \xleftarrow{\$} \mathsf{Init}^D()$
for $i \in [d]$ **do**
  // Invariant: At the beginning of step i,
  // $\left(\cup_{j<i} \mathsf{Found}_j \cup \overline{\mathcal{P}}\right) \subseteq \mathsf{Ord}_i[\geq \mathsf{end}_i + 1]$
  $(\mathcal{T}'_i, \mathsf{Found}_i, \mathsf{pos}_i) \xleftarrow{\$} \mathsf{TrStep}^D(i)$
  $\mathsf{Ord}_{i+1} \leftarrow \mathsf{Ord}_i$
  $\mathsf{Ord}_{i+1}[\mathsf{end}_i] \leftarrow \mathsf{Found}_i$
  $\mathsf{Ord}_{i+1}[\mathsf{pos}_i] \leftarrow \mathsf{Ord}_i[\mathsf{end}_i]$
  $\mathsf{end}_{i+1} \leftarrow \mathsf{end}_i - 1$
return $\mathcal{T}' \leftarrow \bigcup_{i\in[d]} \mathcal{T}'_i$

Subroutine $\mathsf{Init}^{D(\cdot)}()$

$\mathsf{test} \xleftarrow{\$} \mathsf{TestDec}^D_{\Pi'}\left(\mathsf{pk}, \emptyset, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}, \frac{8\lambda}{\varepsilon^2}, \frac{\varepsilon}{2}\right)$
$\bar{\rho} \leftarrow \min(\rho, t_1 - 1)$
$\overline{\mathcal{P}} \leftarrow \left\{j \in [\ell] \colon \mathcal{I}_{\mathsf{pd}} \ (\bar{\rho}+1)\text{-dominates } B_1^{(j)}\right\}$
Let Ord_1 s.t. $\overline{\mathcal{P}}$ occupy the last positions
$d \leftarrow \max\{0, t_{\mathsf{parts}} - |\overline{\mathcal{P}}|\}$
$\mathsf{end}_1 \leftarrow \ell - |\overline{\mathcal{P}}|$
$\mathsf{Cut} \leftarrow \ell - t_{\mathsf{parts}} + 1$
return $(\mathsf{test}, \mathsf{Ord}_1, d, \mathsf{end}_1, \mathsf{Cut})$

Subroutine $\mathsf{TrStep}^{D(\cdot)}(i)$

$\mathsf{Found}_i \leftarrow \perp$; $\mathcal{T}'_i \leftarrow \emptyset$
for $j \in \mathsf{Ord}_i[\leq \mathsf{Cut}]$ **do**
  if $\mathsf{test} = 1$ **then** Construct $D_j^{(i)}$
  else Set $D_j^{(i)} \leftarrow \mathsf{Dummy}$
  $\mathcal{T}'_i(j) \leftarrow \Pi.\mathsf{Trace}^{D_j^{(i)}}(\Pi.\mathsf{pk}^{(j)}, \mathsf{tk}^{(j)}, \varepsilon', \mathcal{I}_{\mathsf{pd}} \cap B_1^{(j)}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{pd}}\cap B_1^{(j)}})$
  $\mathcal{T}'_i \leftarrow \mathcal{T}'_i \cup \mathcal{T}'_i(j)$
  if $\mathsf{Found}_i = \perp \wedge \mathcal{T}'_i(j) \neq \emptyset$ **then**
    $\mathsf{Found}_i \leftarrow j$
    Set pos_i s.t. $\mathsf{Found}_i = \mathsf{Ord}_i[\mathsf{pos}_i]$
return $(\mathcal{T}'_i, \mathsf{Found}_i, \mathsf{pos}_i)$

Decoder $D_j^{(i)}(k_j^\Pi, c_j^\Pi, \{d_v^{(j)}\}_{v\in\mathcal{I}_{\mathsf{pd}}\cap B_1^{(j)}})$

Let pos_j s.t. $j = \mathsf{Ord}_i[pos_j]$
$(k^0, k^1) \xleftarrow{\$} \mathcal{K}^2$
$\{s_q\}_{q\in[\ell]} \xleftarrow{\$} \mathsf{Share}(\ell, t_{\mathsf{parts}}, k^0)$
$\forall q \in [\ell] \setminus \{j\} \colon (c_q^\Pi, k_q^\Pi) \xleftarrow{\$} \Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(q)})$
$\forall q \in [\ell] \colon (c_q^{\mathsf{TE}}, k_q^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{TE.pk}^{(q)})$
  // Revoking partitions prior to j in Ord_i
$\forall q \in \mathsf{Ord}_i[< pos_j] \colon u_q \xleftarrow{\$} \mathcal{K}$
$\forall q \in \mathsf{Ord}_i[\geq pos_j] \colon u_q \leftarrow s_q \oplus k_q^\Pi \oplus k_q^{\mathsf{TE}}$
$c \leftarrow \{((c_q^\Pi, c_q^{\mathsf{TE}}), u_q)\}_{q\in[\ell]}$
$\forall v \in \mathcal{I}_{\mathsf{pd}} \colon d_v \leftarrow \mathsf{PDec}(c, sk_v)$
$b \xleftarrow{\$} \{0,1\}$
return $b \oplus D(k^b, c, \{d_v\}_{v\in\mathcal{I}_{\mathsf{pd}}})$

Figure 6: The tracing algorithm of Construction 6. In Init, we test the decoder for the same reason as in the 1-partition compiler (Section 2.3). The TestDec algorithm is defined in Figure 4.

argument must initially guess a step i^* and a partition j^* and embeds its instance of Π into partition j^*. The issue is that, in order to simulate up to step i^*, the reduction must simulate execution of all steps $i < i^*$ to come up with Ord_{i^*}. If, however, it holds that $j^* \in \mathsf{Ord}_i[\leq \mathsf{Cut}]$ for any step $i < i^*$, then simulating that step requires the reduction to run tracing on the challenge partition j^*. This is clearly problematic whenever Π is not a public tracing scheme. However, we make the following observation, which allows us to instantiate even with private tracing schemes, should they offer a certain additional property. Before presenting the property, observe that, in order to simulate up to Step i^*, the reduction does not actually need to learn $\mathcal{T}'_i(j^*)$ in full. It only needs to decide whether to shuffle partition j^* to the back in the subsequent step, i.e., whether $j^* = \mathsf{Found}_i$. To do so, the reduction does not require the full $\mathcal{T}'_i(j^*)$, but merely requires to learn the predicate, whether $\mathcal{T}'_i(j^*) = \emptyset$. This is a very relevant distinction, because in some private-tracing schemes, in particular in fingerprinting-based schemes, any party can check whether tracing a given decoder would yield an empty set or not *without access to the tracing key* tk.

We model this property by granting the adversary in Game-Tr (Figure 3) access to the $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ oracle introduced in Figure 7. To prove traceability of Construction 6, we require additionally, that Π offers $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$. We demonstrate, at the example of our scheme with

$\mathcal{O}^{\mathsf{Trace\text{-}Empty}}(\widetilde{D}, \widetilde{\mathcal{I}_{\mathsf{pd}}})$

if $\mathsf{Trace}^{\widetilde{D}}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \widetilde{\mathcal{I}_{\mathsf{pd}}}, \{\mathsf{sk}_i\}_{i \in \widetilde{\mathcal{I}_{\mathsf{pd}}}}) = \emptyset$ **then return** 1

else return 0

Figure 7: The $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ oracle that returns 1, if tracing a decoder D returns an empty set and 0 otherwise.

native partial insider resilience in Section 9, how fingerprinting-based schemes can provide such an oracle.

Remark 8.4 In the above compiler, if the decoder D has decryption advantage ε, we trace $D_j^{(i)}$ using $\varepsilon' = \varepsilon/16\ell$ in each step. Since ℓ is not a constant, this necessitates that the base scheme Π be *limitless*, meaning that it can trace decoders with arbitrarily small but noticeable decryption advantage. To ensure this property, we generalize the limit-elimination compiler of [54, 28] to the insider setting (Theorem C.2). This limit-elimination compiler should be applied to Π *prior* to our compiler. We emphasize that it incurs only a constant ciphertext overhead, independent of the number of parties n.

Proof of Traceability (Theorem 8.1). Our proof comprises 4 major parts.

Part 1: At Most 0.2ℓ Partitions are Dominated by Partial Insiders. We show that if the adversary specifies at most $\rho' = n/2 - c_2 n$ partial insiders, then at most 0.2ℓ partitions can be t_1-dominated in the first bin by partial insiders. As a result, there must be at least $d \geq 0.6\ell$ steps in tracing.

Lemma 8.5 *If $|\mathcal{I}_{\mathsf{pd}}| \leq \rho'$, then $|\overline{\mathcal{P}}| \leq 0.2\ell$ except with negligible probability.*

Proof of Lemma 8.5. During key generation, Construction 6 independently samples a collection of ℓ uniform random $(2, n)$-unbalanced partitions, such that $|B_1^{(j)}| = n^a$ for all $j \in [\ell]$. We show via a probabilistic argument that, for an appropriate choice of ℓ (specified in Theorem 8.1), the probability that there exists a set of size at most ρ' that $(0.2\ell, (t_1, 0))$-dominates $\{P_j\}_{j \in [\ell]}$ is negligible. To see this, note that the expected number of parties from an a priori fixed set of size ρ' in the first bin of a given partition is $(\frac{1}{2} - c_2)n^a$. Thus, to dominate a partition, $\mathcal{I}_{\mathsf{pd}}$ must deviate from this expectation by $\Theta(c_2 n^a)$. The probability that this deviation occurs in $\geq 0.2\ell$ partitions is $\exp(-\Theta(0.2\ell c_2 n^a))$. The argument then proceeds as in the proof of Lemma 8.3: To account for adaptive choice of $\mathcal{I}_{\mathsf{pd}}$, we apply a union bound over all possible choices of $\mathcal{I}_{\mathsf{pd}}$, and conclude that such $\mathcal{I}_{\mathsf{pd}}$ exist only with negligible probability. The full probabilistic argument is provided in Part 3 of Lemma 8.2. ∎

Part 2: Tracing Succeeds in Every Step. We show that in each step $i \in [d]$, a partition from $\mathsf{Ord}_i[\leq \mathsf{Cut}]$ is found, and that honest parties cannot be framed in any step. To formalize this intuition, we first define what it means that tracing *succeeds* in a step $i \in [d]$. Recall that $\mathcal{T}_i'(j)$ denotes the traitors found from partition j in step i. Denote with $\mathsf{GoodStep}_i$ for $i \in [d]$ the following event:

$$\mathsf{GoodStep}_i = \begin{cases} 1 & \text{if } \mathsf{Found}_i \neq \perp \text{ and } \mathcal{T}_i'(\mathsf{Found}_i) \subseteq \mathcal{T} \cap B_1^{(\mathsf{Found}_i)} \\ 0 & \text{otherwise.} \end{cases}$$

Denote with $\mathsf{GoodSteps}$ the event that $\mathsf{GoodStep}_i = 1$ for all $i \in [d]$. Also, denote with $\mathsf{BadStep}_i$ the event that any non-traitor is framed as traitor in step i, and with $\mathsf{BadSteps}$ the event that there exists an $i \in [d]$ such that $\mathsf{BadStep}_i = 1$.

Lemma 8.6 *Let* Π *be a* 1*-traceable threshold KEM against* ρ *partial insiders. For all PPT adversaries* $\mathcal{A}$ *against* Game-Tr *(Figure 3) of* Π' *there exists a negligible function* negl *such that*

$$\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda) \text{ and } \Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq \mathsf{negl}(\lambda).$$

Lemma 8.6 does the heavy-lifting in the traceability proof, so, for better clarity, we defer its proof to Section 8.2 and continue with the proof outline for now.

Part 3: τ' unique traitors are found. We show that if we find at least one traitor from the first bin of 0.6ℓ partitions, then τ' of the traitors found must be *distinct*.

Lemma 8.7 *If traitors are identified in* $d \geq 0.6\ell$ *steps, then except with negligible probability it holds that* $|\mathcal{T}'| \geq \tau' = 0.3n^{1-a}$.

Proof of Lemma 8.7. Suppose traitors are found in at least 0.6ℓ steps. Recall that after each step, the first partition in which a traitor is found is moved to the back and cannot contribute in subsequent steps. Therefore, traitors are found in the first bin of at least 0.6ℓ *distinct* partitions.

For any fixed party, the expected number of partitions in which it appears in the first bin is ℓ/n^{1-a}. By a standard Chernoff bound and a union bound over all n parties, the probability that any party appears in the first bin of more than $2\ell/n^{1-a}$ partitions is negligible in λ when $\ell = \Theta(\lambda n^{1-a})$. (See Part 4 of Lemma 8.2).

Consequently, among the traitors identified from distinct partitions, at least $\frac{0.6\ell}{2\ell/n^{1-a}} = 0.3n^{1-a}$ must be distinct. ∎

Part 4: Putting it all together. Equipped with Lemma 8.6, we can easily conclude unframability of Π', i.e., $\Pr[\mathsf{BadTr}_{\mathcal{A}}] = \Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq \mathsf{negl}(\lambda)$.

Next, we look at GoodTr. Following Lemma 8.5, it must hold that $|\overline{\mathcal{P}}| \leq 0.2\ell$, and thus tracing carries out at least $d \geq t_{\mathsf{parts}} - 0.2\ell = 0.6\ell$ steps with overwhelming probability. Now suppose that $\mathsf{GoodSteps}$ happens, which implies that we found traitors in all $d \geq 0.6\ell$ steps. We apply Lemma 8.7 to argue that in this case we must have found $|\mathcal{T}'| \geq \tau' = 0.3n^{1-a}$ *unique* real traitors, which implies $\mathsf{GoodTr} = 1$. Finally, we apply Lemma 8.6 and get that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodSteps}_{\mathcal{A}}] - \mathsf{negl}(\lambda) \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda) - \mathsf{negl}'(\lambda),$$

for negligible functions negl and negl'. This concludes the proof of Theorem 8.1. ∎

8.2 A Proof of Lemma 8.6

Proof of Lemma 8.6. Let $\mathcal{I}_{\mathsf{pd}}$ be the set of partial insiders output by $\mathcal{A}$ and $\overline{\mathcal{P}}$ defined as in the Trace algorithm. Assume w.l.o.g. that $|\overline{\mathcal{P}}| < t_{\mathsf{parts}}$, since otherwise the lemma holds trivially as tracing would carry out $d = 0$ steps.

Proof Outline. Our proof roughly proceeds as follows: First, we construct a reduction $\mathcal{B}$ to 1-traceability against $(t_1 - 1)$ partial insiders of Π, and use $\mathcal{B}$ to upper bound $\Pr[\mathsf{BadSteps}_{\mathcal{A}}]$.

Then, we show that $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$ through a hybrid argument over the maximum number of steps t_{parts} executed during tracing. In the hybrid $\mathcal{H}_i$ for $i \in [t_{\mathsf{parts}}]$, we essentially set $\mathsf{GoodSteps} \leftarrow \mathsf{test}$, if there exists a step $u \leq i$ such that $\mathsf{GoodStep}_u = 0$.

We prove $\Pr[\mathsf{GoodSteps}] \geq \Pr[\mathsf{test}]$ to hold in the final hybrid $\mathcal{H}_{t_{\mathsf{parts}}}$. A probabilistic argument shows that test is implied by $\mathsf{GoodDec}$ except with negligible probability, and thus $\Pr[\mathsf{GoodSteps}] \geq \Pr[\mathsf{GoodDec}] - \mathsf{negl}(\lambda)$ holds in $\mathcal{H}_{t_{\mathsf{parts}}}$.

To close the proof, it remains to show that we increased $\Pr[\mathsf{GoodSteps}]$ at most by a negligible amount between $\mathcal{H}_0$ and $\mathcal{H}_{t_{\mathsf{parts}}}$. For this, we show that in *subsequent* hybrids, the increase in $\Pr[\mathsf{GoodSteps}]$ can be bound by the advantage of a reduction $\mathcal{B}$ against traceability of Π. To do so, we first show by a PLBE-style argument that if D is good, then in every step i, at least

one of the decoders $D_j^{(i)}$ must be good, where $j \in \mathsf{Ord}_i[\leq \mathsf{Cut}]$. Similar to the reduction for the $\mathsf{BadSteps}$ part, $\mathcal{B}$ initially guesses a step i^* and a partition j^* into which it embeds its challenge instance of Π. We modify $\mathcal{B}$ to only ever output a good decoder whenever $D_{j^*}^{(i^*)}$ is the *first* good decoder in $\mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$. This allows us to prove that $\Pr[\mathsf{GoodDec}_{\mathcal{B}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}]/\ell$. Recall that dummy decoders cannot be traced except with negligible probability. $\mathcal{B}$ outputs a dummy decoder whenever it guesses j^* incorrectly (i.e., not the first good decoder in step i^*), and therefore $\mathsf{GoodTr}_{\mathcal{B}}$ can only ever happen if $\mathcal{B}$ guesses j^* correctly. In this case, it holds that $\mathsf{GoodTr}_{\mathcal{B}}$ implies $\mathsf{GoodStep}_i$, and thus $\Pr[\mathsf{GoodTr}_{\mathcal{B}}] \leq \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}]/\ell$. We can conclude that

$$\Pr\Big[\neg\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}\Big] \leq \ell \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \Pr[\mathsf{GoodTr}_{\mathcal{B}}]),$$

conditioned on the event that $\mathcal{B}$ guesses $i^* = i$. This inequality bounds the increase in $\Pr[\mathsf{GoodSteps}]$ between $\mathcal{H}_{i-1}$ and $\mathcal{H}_i$ by $\mathcal{B}$'s advantage against traceability of Π.

Proving $\Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq \mathsf{negl}(\lambda)$. We will start our proof with the latter part of the lemma and show that for any PPT adversary $\mathcal{A}$, there exists a PPT reduction $\mathcal{B}$ to 1-traceability against ρ partial insiders of Π such that $\Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq \ell \cdot t_{\mathsf{parts}} \cdot \Pr[\mathsf{BadTr}_{\mathcal{B}}]$.

Reduction $\mathcal{B}$ to traceability of Π. The reduction $\mathcal{B}$ initially samples a random guess for a step $i^* \xleftarrow{\$} [t_{\mathsf{parts}}]$ and a partition $j^* \xleftarrow{\$} [\ell]$. The reduction runs against $\mathsf{Game\text{-}Tr}$ of Π and simulates $\mathsf{Game\text{-}Tr}$ of Π' to $\mathcal{A}$ by embedding the challenge public key into the first bin of partition j^*, simulating all remaining partitions according to protocol. Denote with $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$ the random variables corresponding to the corrupted parties of $\mathcal{A}$ and $\mathcal{B}$. Note that this means that $\mathcal{B}$ knows all secret keys and tracing keys except for partition j^*. In partition j^*, $\mathcal{B}$ knows only the TE secret keys of parties in B_2. When $\mathcal{A}$ issues a corruption query for party p and $p \in B_1^{(j^*)}$, then $\mathcal{B}$ queries its own corruption oracle on p to learn $\mathsf{sk}_p^{(j^*)}$ of Π. All remaining secret keys of party p (i.e., those of the threshold encryption scheme TE and the Π keys of party p in other partitions) are known to $\mathcal{B}$ from the simulation. In particular, this means that $\mathcal{T}_{\mathcal{A}} \cap B_1^{(j^*)} = \mathcal{T}_{\mathcal{B}}$. To simulate $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}})$, $\mathcal{B}$ simply queries $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ of its own challenger, whenever $i \in B_1^{(j^*)}$.

When $\mathcal{A}$ outputs a decoder D and a set of partial insiders $\mathcal{I}_{\mathsf{pd}}$, $\mathcal{B}$ needs to construct decoder $D_{j^*}^{(i^*)}$ (Fig. 6). Since the construction of this decoder relies on the ordering Ord_{i^*}, $\mathcal{B}$ simulates the tracing procedure up to step i^*. To do so, $\mathcal{B}$ must first simulate all decoders $D_j^{(i)}$ in each step $i \leq i^*$ and then also trace them to obtain the ordering Ord_{i+1} for the subsequent step. Denote for all $j \in [\ell]$ with $\mathcal{I}_{\mathsf{pd}}{}^{(j)} = \mathcal{I}_{\mathsf{pd}} \cap B_1^{(j)}$ the set of partial insiders in the first bin of partition j. To simulate decoders as constructed by $\Pi'.\mathsf{Trace}$ for $j \neq j^*$, $\mathcal{B}$ must provide partial decryptions on behalf of $\mathcal{I}_{\mathsf{pd}}{}^{(j^*)}$ for honestly generated ciphertexts, which it can obtain through its challenger's $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle. More formally, to simulate $D_j^{(i)}$ for $j \neq j^*$ given Ord_i, $\mathcal{B}$ does the following: To simulate $D_j^{(i)}\Big(k_j^{\Pi}, c_j^{\Pi}, \{d_i^{(j)}\}_{i \in \mathcal{I}_{\mathsf{pd}}{}^{(j)}}\Big)$,

- Generate $(c_q^{\Pi}, k_q^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(q)})$ for all $q \in [\ell] \backslash \{j, j^*\}$. Compute $d_i^{(q)} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(q)}, c_q^{\Pi})$ for all $q \in [\ell] \setminus \{j, j^*\}$ and $i \in \mathcal{I}_{\mathsf{pd}} \cap B_1^{(q)}$.
- Sample $r_{\mathsf{Enc}}^{(j^*)} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$ and compute $c_{j^*}^{\Pi} \leftarrow \Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(j^*)}; r_{\mathsf{Enc}}^{(j^*)})$. For all $i \in \mathcal{I}_{\mathsf{pd}}{}^{(j^*)}$ obtain partial decryptions $d_i \leftarrow \mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}}^{(j^*)})$ making use of the challenger's oracle.
- Generate $(c_q^{\mathsf{TE}}, k_q^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{TE.pk}^{(q)})$ for all $q \in [\ell]$. Compute $d_i^{(q)} \leftarrow \mathsf{TE.PDec}(\mathsf{sk}_i^{(q)}, c_q^{\mathsf{TE}})$ for all $q \in [\ell]$ and $i \in \mathcal{I}_{\mathsf{pd}} \cap B_2^{(q)}$.
- Sample $(k^0, k^1) \xleftarrow{\$} \mathcal{K}^2$ as well as a random bit $b \xleftarrow{\$} \{0,1\}$. Compute $\{s_q\}_{q \in [\ell]} \xleftarrow{\$} \mathsf{Share}(\ell, t_{\mathsf{parts}}, k^0)$.

- Let pos_j such that $j = \mathsf{Ord}_i[pos_j]$. Then, for $q \in \mathsf{Ord}_i[< pos_j]$, sample $u_q \xleftarrow{\$} \mathcal{K}$, for $q \in \mathsf{Ord}_i[\geq pos_j]$ (in particular, for $q = j$), set $u_q \leftarrow s_q \oplus k_q^{\Pi} \oplus k_q^{\mathsf{TE}}$.
- Assemble $c = \{((c_q^{\Pi}, c_q^{\mathsf{TE}}), u_q)\}_{q\in[\ell]}$ as well as $d_i \leftarrow \{d_i^{(q)}\}_{q\in[\ell]}$ for all $i \in \mathcal{I}_{\mathsf{pd}}$. Derive the output of $D_j^{(i)}$ as $b \oplus D(k^b, c, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$.

Next, consider the case that partition j^* is already in $\mathsf{Ord}_i[\leq \mathsf{Cut}]$ for some $i < i^*$. In this case, $\mathcal{B}$ cannot simulate tracing the decoder $D_{j^*}^{(i)}$, since $\mathcal{B}$ does not know tk_{j^*}. However, $\mathcal{B}$ can query its challenger's oracle $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}(D_{j^*}^{(i)}, \mathcal{I}_{\mathsf{pd}}{}^{(j^*)})$ to learn whether tracing $D_{j^*}^{(i)}$ would return an empty set or not, and thus update the ordering in step i accordingly. Having obtained Ord_{i^*}, $\mathcal{B}$ constructs $D_{j^*}^{(i^*)}$ and returns it to its challenger alongside $\mathcal{I}_{\mathsf{pd}}{}^{(j^*)}$. If $i^* > d$ or $j^* \notin \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$, $\mathcal{B}$ just outputs a dummy decoder.

Analysis. Whenever $\mathcal{A}$ triggers $\mathsf{BadSteps}$, there exists $i \in [t_{\mathsf{parts}}]$ and $j \in [\ell]$ such that tracing $D_i^{(j)}$ outputs at least one non-traitor. Since $\mathcal{A}$'s probability of triggering $\mathsf{BadSteps}$ is independent of $\mathcal{B}$'s choice of j^* and i^*, $\mathcal{B}$ guesses i and j correctly with probability at least $1/\ell t_{\mathsf{parts}}$. If $\mathcal{B}$ guesses correctly, then $\mathcal{T}'_{\mathcal{B}} = \mathcal{T}'_{i^*}(j^*) \not\subseteq \mathcal{T}(j^*) = \mathcal{T}_{\mathcal{B}}$, and thus $\Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq \ell \cdot t_{\mathsf{parts}} \cdot \Pr[\mathsf{BadTr}_{\mathcal{B}}]$.

Proving $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$. Next, we show that for any PPT adversary $\mathcal{A}$ it holds that $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$. We define a series of hybrids $\mathcal{H}_i$ for $i \in [0, t_{\mathsf{parts}}]$ such that $\mathcal{H}_0$ is identical to the real game. In hybrid $\mathcal{H}_i$, however, we modify the definition of events as follows: If for any $u \leq i$ it holds that $\mathsf{GoodStep}_u = 0$, then we overwrite $\mathsf{GoodSteps} \leftarrow \mathsf{test}$, i.e., if the adversary manages that in any step $u \leq i$ either no partition is found or non-traitors are framed from the found partition, then we just set $\mathsf{GoodSteps}$ to 1, iff the decoder D tests good ($\mathsf{test} = 1$).

Denote with $\mathsf{GoodSteps}^{(i)}$ a random variable representing the event $\mathsf{GoodSteps}$ in hybrid $\mathcal{H}_i$. First, observe that in the last hybrid $\mathcal{H}_{t_{\mathsf{parts}}}$ it trivially holds that for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr\left[\mathsf{GoodSteps}_{\mathcal{A}}^{(t_{\mathsf{parts}})}\right] \geq \Pr[\mathsf{test}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda),$$

since, by a simple Hoeffding inequality, if a decoder is good, it will only test bad with negligible probability.[14]

It remains to show that for all PPT adversaries there exists a negligible function negl such that

$$\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}] \geq \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t_{\mathsf{parts}})}] - \mathsf{negl}(\lambda). \tag{1}$$

Before we prove Equation 1, we must first argue that whenever D tests good ($\mathsf{test} = 1$), then in every step $i \in [d]$, there exists a decoder $D_j^{(i)}$ with $j \in \mathsf{Ord}_i[\leq \mathsf{Cut}]$ that is at least $\varepsilon/8\ell$-good against the instance of Π in partition j.

Claim 8.8 If $\mathsf{test} = 1$, then for all $i \in [d]$ there must exist a $j \in \mathsf{Ord}_i[\leq \mathsf{Cut}]$ such that

$$P_{\Pi_j}(D_j^{(i)}) \geq 1/2 + \varepsilon/8\ell,$$

except with negligible probability, where P_{Π_j} is with respect to instance Π_j and partial insiders $\mathcal{I}_{\mathsf{pd}}{}^{(j)}$.

Proof of Claim 8.8. Suppose test happens. By a simple Hoeffding inequality, we can show that if test happens, then D is at least $\varepsilon/4$-good with overwhelming probability.[15] We next apply a thresholdized PLBE-style argument.

[14] We proved an identical statement in Claim D.2 using Hoeffding's inequality.
[15] An identical statement was shown to hold in the proof of Claim D.3.

Suppose that the decoder D is $\varepsilon/4$-good. Then by definition, it holds that

$$\Pr\left[\begin{array}{c} D\Big(\{(c_j,u_j)\}_{[\ell]},k^b,\{d_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}},j\in[\ell]}\Big) \\ =b \end{array} : \begin{array}{c} b\xleftarrow{\$}\{0,1\},(k^0,k^1)\xleftarrow{\$}\mathcal{K}^2 \\ (s_1,\ldots,s_\ell)\xleftarrow{\$}\mathsf{Share}(k_0,\ell,t_{\mathsf{parts}}) \\ \text{For all } j\in[\ell]\text{:} \\ (c_j^{\Pi},k_j^{\Pi})\xleftarrow{\$}\Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(j)}) \\ \Big\{d_i^{(j)}\leftarrow\Pi.\mathsf{PDec}(\mathsf{sk}_i^{(j)},c_j^{\Pi})\Big\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j)}} \\ (c_j^{\mathsf{TE}},k_j^{\mathsf{TE}})\xleftarrow{\$}\Pi.\mathsf{Enc}(\mathsf{TE}.\mathsf{pk}^{(j)}) \\ \Big\{d_i^{(j)}\leftarrow\mathsf{TE}.\mathsf{PDec}(\mathsf{sk}_i^{(j)},c_j^{\mathsf{TE}})\Big\}_{i\in\mathcal{I}_{\mathsf{pd}}\cap B_2^{(j)}} \\ u_j\leftarrow s_j\oplus k_j^{\Pi}\oplus k_j^{\mathsf{TE}} \end{array}\right]\geq\frac{1}{2}+\frac{\varepsilon}{4}. \tag{2}$$

Let $X_h^{(i)}(b)$ for $h\in[0,\mathsf{Cut}]$ be a series of distributions on $\Big(\{(c_q,u_q)\}_{q\in[\ell]},k^b,\{d_i^{(q)}\}_{i\in\mathcal{I}_{\mathsf{pd}},q\in[\ell]}\Big)$, where $X_0^{(i)}(b)$ is as in the experiment in Equation 2, but fixing the bit b. For all $h\in[\mathsf{Cut}]$, we define $X_h^{(i)}(b)$ to be the same distribution, except that for all $q\in\mathsf{Ord}_i[\leq h]$, u_q is replaced by a uniformly sampled element from $\mathcal{K}$, respectively. We define the probability

$$p_h^{(i)}\coloneqq\Pr\left[D\left(X_h^{(i)}(b)\right)=b\colon b\xleftarrow{\$}\{0,1\}\right].$$

Then, by definition, it holds that $p_0^{(i)}\geq 1/2+\varepsilon/4$. Now observe that in $X_{\mathsf{Cut}}^{(i)}(b)$, a total of Cut elements u_j are uniformly random from $\mathcal{K}$, and thus $X_{\mathsf{Cut}}^{(i)}(b)$ only depends on $t_{\mathsf{parts}}-1$ shares of k^0, which is unconditionally independent of k^0. As a result $X_{\mathsf{Cut}}^{(i)}(0)\equiv X_{\mathsf{Cut}}^{(i)}(1)$ and thus $p_{\mathsf{Cut}}^{(i)}=1/2$. As a consequence, there must be at least one $h\in[\mathsf{Cut}]$ such that $p_h^{(i)}\leq p_{h-1}^{(i)}-\varepsilon/(4\cdot\mathsf{Cut})$.

Let $j^*=\mathsf{Ord}_i[h^*]$ for such h^* in step i and consider the decoder $D_{j^*}^{(i)}$ on input $\Big(c_{j^*}^{\Pi},k_{j^*}^{\Pi},\{d_i^{(j^*)}\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j^*)}}\Big)$. Suppose $k_{j^*}^{\Pi}$ is the correct key, i.e., $(c_{j^*}^{\Pi},k_{j^*}^{\Pi})$ is sampled from $\Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(j^*)})$. In this case $D_{j^*}^{(i)}$ samples a random bit b and outputs $b\oplus D(X_{h^*-1}^{(i)}(b))$, and thus, under the condition that $k_{j^*}^{\Pi}$ is correct, $D_{j^*}^{(i)}$ outputs 0 whenever D correctly guesses b, which happens with probability $p_{h^*-1}^{(i)}$. If, however, $k_{j^*}^{\Pi}$ is invalid, then $D_{j^*}^{(i)}$ runs D on distribution $X_{h^*}^{(i)}(b)$ for a random bit b and outputs 1, whenever D *incorrectly* guesses b. Hence $D_{j^*}^{(i)}$ *correctly* guesses that $k_{j^*}^{\Pi}$ is invalid with probability $1-p_{h^*}^{(i)}$. Resolving the conditional probabilities, we get that $D_{j^*}^{(i)}$ correctly *predicts* the validity of $k_{j^*}^{\Pi}$ with the following probability:

$$\frac{1}{2}\left(p_{h^*-1}^{(i)}+1-p_{h^*}^{(i)}\right)=\frac{1}{2}+\frac{p_{h^*-1}^{(i)}-p_{h^*}^{(i)}}{2}\geq\frac{1}{2}+\frac{\varepsilon}{8\ell}$$

■

To prove Equation 1, we will show that for all $i\in[t_{\mathsf{parts}}]$, $\Pr[\mathsf{GoodSteps}^{(i)}]-\Pr[\mathsf{GoodSteps}^{(i-1)}]$ is negligible. To do so, as explained in the outline, we must modify the reduction $\mathcal{B}$ as to only produce a good decoder whenever it guesses $j^*\in\mathsf{Ord}_{i^*}[\leq\mathsf{Cut}]$ to be the *first* partition where $D_{j^*}^{(i^*)}$ is good. For this purpose, we will let $\mathcal{B}$ *test* all decoders in step i^*.

Modifying the reduction $\mathcal{B}$. Consider the reduction $\mathcal{B}$ we constructed for the $\mathsf{BadSteps}$ case. Recall that the reduction initially samples random $i^*\xleftarrow{\$}[t_{\mathsf{parts}}]$ and $j^*\xleftarrow{\$}[\ell]$. We modify $\mathcal{B}$ to carry out an additional test on each decoder $D_j^{(i^*)}$ for $j\in\mathsf{Ord}_{i^*}[\leq\mathsf{Cut}]$. For this, $\mathcal{B}$ uses the simulated decoders as described in the $\mathsf{BadSteps}$ proof above. Let $Q'\coloneqq 512\ell^2\lambda/\varepsilon^2$, and simulate for all $j\in\mathsf{Ord}_{i^*}[\leq\mathsf{Cut}]$: $\mathsf{test}_j\xleftarrow{\$}\mathsf{TestDec}^{D_j^{(i^*)}}\Big(\Pi.\mathsf{pk}^{(j)},\emptyset,\mathcal{I}_{\mathsf{pd}}{}^{(j)},\{\mathsf{sk}_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j)}},Q',\frac{3\varepsilon}{32\ell}\Big)$. Observe that for $j=j^*$, the reduction does not know $\{\mathsf{sk}_i^{(j^*)}\}_{i\in\mathcal{I}_{\mathsf{pd}}{}^{(j^*)}}$. However, these secret keys are

only used during testing for generating partial decryptions for honestly generated ciphertexts, and thus $\mathcal{B}$ can instead simulate these using its $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle. More formally, $\mathcal{B}$ can compute test_{j^*} as follows:

- Sample $r_{\mathsf{Enc}}^{(j^*)} \xleftarrow{\$} \Pi_{j^*}.\mathcal{R}_{\mathsf{Enc}}$, compute $(c, k^0) \leftarrow \Pi.\mathsf{Enc}(\Pi.\mathsf{pk}^{(j^*)}; r_{\mathsf{Enc}}^{(j^*)})$, and obtain partial decryptions by querying the oracle $d_i \leftarrow \mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}}^{(j^*)})$ for all parties $i \in \mathcal{I}_{\mathsf{pd}}{}^{(j^*)}$.
- Sample $k^1 \xleftarrow{\$} \mathcal{K}$, and $b \xleftarrow{\$} \{0,1\}$.
- Run $b' \xleftarrow{\$} D_j^{(i^*)}(c, k^b, \{d_i^{(j)}\}_{i \in \mathcal{I}_{\mathsf{pd}}{}^{(j)}})$. If $b' = b$, increment ctr.
- If $ctr \geq Q' \cdot (1/2 + 3\varepsilon/32\ell)$, set the flag $\mathsf{test}_{j^*} \leftarrow 1$. Otherwise, set $\mathsf{test}_{j^*} \leftarrow 0$.

Next, we make the following modification to the decoder that is returned by $\mathcal{B}$: If $\mathsf{test}_{j^*} = 1$ and $\mathsf{test}_j = 0$ for all $j \in \mathsf{Ord}_{i^*}[< pos_{j^*}]$ s.t. $j^* = \mathsf{Ord}_{i^*}[pos_{j^*}]$ (i.e., if $D_{j^*}^{(i^*)}$ is the *first* decoder that tested good) then $\mathcal{B}$ outputs $D_{j^*}^{(i^*)}$ and the set $\mathcal{I}_{\mathsf{pd}}{}^{(j^*)}$ to the challenger as before. Otherwise, $\mathcal{B}$ outputs a Dummy decoder that always returns 0 instead.

Analysing $\Pr[\mathsf{GoodDec}_{\mathcal{B}}]$. Given the modified reduction $\mathcal{B}$, we show that whenever $\mathsf{test}_{\mathcal{A}}$ happens, then $\mathcal{B}$ produces an $\varepsilon/16\ell$-good decoder against Π_j with probability almost $1/\ell$. More formally, we show the following claim.

Claim 8.9 For all PPT adversaries $\mathcal{A}$, it holds that

$$\Pr[\mathsf{test}_{\mathcal{A}}] \leq \ell \cdot \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \mathsf{negl}(\lambda).$$

Proof of Claim 8.9. We want to show that, if $\mathsf{test}_{\mathcal{A}}$ happens, then $\mathcal{B}$ produces a good decoder with probability almost $1/\ell$ First, we apply Claim 8.8 to argue that, if $\mathsf{test}_{\mathcal{A}}$ happens, then it holds that $P_{\Pi_j}(D_j^{(i^*)}) \geq 1/2 + \varepsilon/8\ell$ for at least one $j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$, except with negligible probability. Since $\mathcal{B}$ tests all decoders $D_j^{(i^*)}$, by a Hoeffding bound, the probability that there exists any j that is $\varepsilon/8\ell$-good, but fails test_j is negligible. To see this, recall that $\mathcal{B}$ tests $D_j^{(i^*)}$ for all $j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$ across Q' repetitions, and sets $\mathsf{test}_j = 1$, if the empirical advantage is at least $3\varepsilon/32\ell$. We can apply a Hoeffding bound to show that for $Q' \geq 512\ell^2\lambda/\varepsilon^2$, a decoder that is $\varepsilon/8\ell$ good, will only ever test bad with probability $\exp(-\lambda)$. A union bound over all Cut decoders that are test_j shows that if test happens, then test_j happens for at least one $j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$, except with negligible probability.

Additionally, we must rule out that $\mathsf{test}_{j^*} = 1$, but $D_{j^*}^{(i^*)}$ is not actually $\varepsilon/16\ell$-good against Π_{j^*}, or in other words, that test_{j^*} is a false positive. An analogous Hoeffding bound shows that if test_{j^*} happens, then $D_{j^*}^{(i^*)}$ must be $\varepsilon/16\ell$-good against Π_{j^*}, except with negligible probability.

Let κ_j be the probability that test_j happens but test_q does not happen for any $q \in \mathsf{Ord}_{i^*}[< pos_j]$ s.t. $j = \mathsf{Ord}_{i^*}[pos_j]$. The above implies that

$$\sum_{j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]} \kappa_j \geq \Pr[\mathsf{test}_{\mathcal{A}}] - \mathsf{negl}(\lambda),$$

and that

$$\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] \geq (1/\ell) \cdot \sum_{j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]} \kappa_j - \mathsf{negl}(\lambda).$$

Combining both, we get that

$$\Pr[\mathsf{test}_{\mathcal{A}}] \leq \ell \cdot \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] + \mathsf{negl}(\lambda).$$

■

Bounding $\Pr[\mathsf{GoodSteps}^{(i)}] - \Pr[\mathsf{GoodSteps}^{(i-1)}]$. Finally, in order to get the desired bound of Equation 1, we want to bound $\Pr[\mathsf{GoodSteps}^{(i)}_{\mathcal{A}}] - \Pr[\mathsf{GoodSteps}^{(i-1)}_{\mathcal{A}}]$ by $\mathcal{B}$'s advantage against traceability of Π. In other words, we aim to show that in subsequent hybrids, $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}]$ can increase by at most a negligible amount. To do so, we first show the following claim.

Claim 8.10 For all PPT adversaries $\mathcal{A}$ and all $i \in [t_{\mathsf{parts}}]$ there exists a negligible function such that $\Pr[\mathsf{GoodSteps}^{(i)}_{\mathcal{A}}] - \Pr[\mathsf{GoodSteps}^{(i-1)}_{\mathcal{A}}] \leq \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \mathsf{negl}(\lambda)$.

Proof of Claim 8.10. First, observe that the only way for $\mathcal{A}$ to cause $\mathsf{GoodSteps}^{(i)}_{\mathcal{A}} = 1$, but $\mathsf{GoodSteps}^{(i-1)}_{\mathcal{A}} = 0$ is to set $\mathsf{GoodStep}_i = 0$, and $\mathsf{test} = 1$. Hence, we get that

$$\Pr[\mathsf{GoodSteps}^{(i)}_{\mathcal{A}}] - \Pr[\mathsf{GoodSteps}^{(i-1)}_{\mathcal{A}}] \leq \Pr[\neg\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}}].$$

We can apply the law of total probability (twice) to get

$$\begin{aligned}\Pr[\neg\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}}] &= \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}}] \\ &= \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}} = 0]\end{aligned}$$

W.l.o.g., we can assume that $\Pr[\mathsf{test}_{\mathcal{A}} = 0] > 0$, and thus

$$\begin{aligned}&\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}} = 0] \\ \leq &\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \mid \mathsf{test}_{\mathcal{A}} = 0].\end{aligned}$$

Observe that $\mathsf{GoodStep}_i$, can only happen if tracing in step i identifies at least one non-empty set. If $\mathsf{test} = 0$, however, the tracing algorithm will trace only dummy decoders in every step, and thus we get

$$\begin{aligned}&\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \mid \mathsf{test}_{\mathcal{A}} = 0] \\ \leq &\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\Pi.\mathsf{Trace}^{\mathsf{Dummy}(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon') \neq \emptyset].\end{aligned}$$

Finally, due to traceability of Π, tracing a dummy decoder can only return a non-empty set with negligible probability, and we get the claim:

$$\Pr[\mathsf{GoodSteps}^{(i)}_{\mathcal{A}}] - \Pr[\mathsf{GoodSteps}^{(i-1)}_{\mathcal{A}}] \leq \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \mathsf{negl}(\lambda).$$

■

Next, we need to show that

$$\begin{aligned}&\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] \\ \leq &\ell \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i]) + \mathsf{negl}(\lambda).\end{aligned} \tag{3}$$

In Claim 8.9, we have already shown that $\Pr[\mathsf{test}_{\mathcal{A}}] \leq \ell \cdot \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] + \mathsf{negl}(\lambda)$. Hence, to prove Equation 3, it remains to show the following claim.

Claim 8.11 For all PPT adversaries $\mathcal{A}$, it holds that

$$\Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] \geq \ell \cdot \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \mathsf{negl}(\lambda).$$

Proof of Claim 8.11. Observe that the respective events that test_j happen are independent of $\mathcal{B}$'s choice of j^*. Hence, given that $\mathsf{test}_j = 1$ for at least one $j \in \mathsf{Ord}_{i^*}[\leq \mathsf{Cut}]$, $\mathcal{B}$ will pick this instance with probability at least $1/\ell$. If $\mathcal{B}$ guesses incorrectly, then $\mathcal{B}$ will output a dummy decoder, in which case $\mathsf{GoodTr}_{\mathcal{B}}$ can only happen with negligible probability. If, however, $\mathcal{B}$ guesses correctly, then $\mathsf{GoodTr}_{\mathcal{B}}$ implies $\mathsf{GoodStep}_i$, because $\mathcal{T}_{\mathcal{B}} = \mathcal{T}_{\mathcal{A}}(j^*)$, and thus

$$\Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] \leq \frac{\ell - 1}{\ell} \cdot \mathsf{negl}(\lambda) + \frac{\Pr[\mathsf{GoodStep}_{i,\mathcal{A}}]}{\ell}.$$

■

Together, Claims 8.9, 8.10, and 8.11 imply that

$$\begin{aligned}&\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i)}] - \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i-1)}]\\ &\leq \ell \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i]) + \mathsf{negl}(\lambda).\end{aligned}$$

Equipped with this analysis, we get that

$$\begin{aligned}&\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t_{\mathsf{parts}})}] - \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}]\\ &= \sum_{i=1}^{t_{\mathsf{parts}}} \left(\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i)}] - \Pr[\mathsf{GoodSteps}\mathcal{A}^{(i-1)}]\right)\\ &\leq \sum_{i=1}^{t_{\mathsf{parts}}} \left(\ell \left(\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i]\right) + \mathsf{negl}'(\lambda)\right)\\ &= \sum_{i=1}^{t_{\mathsf{parts}}} \frac{\ell \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i])}{\Pr[\mathcal{B}.i^* = i]} + t_{\mathsf{parts}} \cdot \mathsf{negl}'(\lambda)\\ &= \ell \cdot t_{\mathsf{parts}} \cdot \sum_{i=1}^{t_{\mathsf{parts}}} (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i]) + t_{\mathsf{parts}} \cdot \mathsf{negl}'(\lambda)\\ &= \ell \cdot t_{\mathsf{parts}} \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \Pr[\mathsf{GoodTr}_{\mathcal{B}}]) + t_{\mathsf{parts}} \cdot \mathsf{negl}'(\lambda)\\ &\leq \ell \cdot t_{\mathsf{parts}} \cdot \mathsf{negl}(\lambda) + t_{\mathsf{parts}} \cdot \mathsf{negl}'(\lambda).\end{aligned}$$

Putting it all together, assuming traceability of Π, and simplifying negligibilities, we get that

$$\begin{aligned}\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] &= \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}]\\ &\geq \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t_{\mathsf{parts}})}] - \mathsf{negl}(\lambda) \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}'(\lambda).\end{aligned}$$

This concludes the proof for Lemma 8.6. ∎

Remark 8.12 Part 2 (Lemma 8.6) was proved for arbitrary t_{parts} and ρ, and essentially shows that in every step $i \in [d]$, a traitor is found from at least one unique partition. Hence, one could redo the analysis in parts 1, 3, and 4 for different t_{parts} values and assumed partial-insider resilience ρ of Π, to get different tradeoffs between partial-insider resilience, numbers of traitors found, and robustness.

9 Achieving Maximal Partial Insider Resilience

To construct a traceable threshold KEM that is natively resilient against partial insiders, we follow the fingerprinting approach and introduce a modified version of a bipartite threshold KEM that allows the tracer to simulate up to $t-1$ partial decryptions for both valid and malformed ciphertexts.

For this, we first define the building block PD-BT-KEM, as well as the requirements for one-sided security in the presence of simulated partial decryptions, which will be essential for tracing. Then, we show that we can instantiate our new building block with a variant of the pairing-based BT-KEM scheme introduced in [11] with minimal modifications.

Finally, we show how to generically transform any PD-BT-KEM scheme to a 1-traceable TT-KEM that is natively resilient against up to $t-1$ partial insiders (i.e., maximally partial-insider resilient), with the restriction that our transformation only achieves traceability against decrypting decoders instead of distinguishing decoders. We leave it as an interesting problem for future work to achieve native partial insider resilience for distinguishing decoders.

9.1 Bipartite Threshold Encryption with Simulated Partial Decryptions

We adjust the BT-KEM definition due to [11] by adding a SimPD algorithm to the syntax.

Definition 9.1 (PD-BT-KEM). A PD-BT-KEM scheme with keyspace $\mathcal{K}_\lambda$ is a tuple of algorithms $\Pi = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{PDec}, \mathsf{SimPD}, \mathsf{Combine})$ with the following syntax.

- $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$. Given the security parameter λ, the Setup algorithm outputs public parameters pp that are implicit input to all subsequent algorithms.
- $\left(\mathsf{pk}, \left\{\left(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)}\right)\right\}_{i\in[n],j\in[\ell]}\right) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \ell)$. KGen receives the security parameter λ, the number of parties $n \in \mathbb{N}$, the threshold $0 < t \leq n$, and the number of positions $\ell \in \mathbb{N}$. It outputs a public key pk, and for every party $i \in [n]$ and every position $j \in [\ell]$ a left secret key share $\mathsf{sk}_{i,0}^{(j)}$ and a right secret key share $\mathsf{sk}_{i,1}^{(j)}$. As shorthand, we denote the set of all secret key shares sk, and the set of all secret key shares for party i as sk_i.
- $(k, c := (c_0, c_1)) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j)$. Given the public key pk and a position $j \in [\ell]$, the encryption algorithm outputs a KEM key $k \in \mathcal{K}_\lambda$ and a ciphertext c that consists of a left component c_0 and a right component c_1
- $d_i \leftarrow \mathsf{PDec}(\mathsf{sk}_{i,b}^{(j)}, (c_0, c_1), j))$. Given the secret key share $\mathsf{sk}_{i,b}^{(j)}$ of party i at position j for side $b \in \{0, 1\}$, a ciphertext (c_0, c_1), and the position j, this PDec algorithm outputs a partial decryption d_i.
- $d_i \xleftarrow{\$} \mathsf{SimPD}(\mathsf{pk}, i, (c_0, c_1), j, r_{\mathsf{Enc}})$. Given a participant index i, a ciphertext (c_0, c_1), the ciphertext position $j \in [\ell]$, and randomness used during encryption r_{Enc}, the SimPD algorithm outputs a simulated partial decryption d_i on behalf of party i.
- $k/\perp \leftarrow \mathsf{Combine}(\mathsf{pk}, S, \{d_i\}_{i\in S}, (c_0, c_1), j)$. Given the public key pk, the set of partial decryptions $\{d_i\}_{i\in S}$ from parties in $S \subseteq [n]$ with $|S| \geq t$, the ciphertext (c_0, c_1), and the position j, output the KEM key k or $\perp$ to indicate failure.

A PD-BT-KEM must additionally achieve the following properties. Like [11], we require that one can mix left and right shares to decrypt for a given position j.

Definition 9.2 (Two-Sided Correctness). A PD-BT-KEM scheme is two-sided correct if for all $n, t, \ell \in \mathsf{poly}(\lambda)$ the following holds: For all $\lambda \in \mathbb{N}$, $j \in [\ell]$, $S \subseteq [n]$ with $|S| \geq t$, all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, all $\left(\mathsf{pk}, \{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n],j\in[\ell]}\right) \in \mathsf{KGen}(1^\lambda, n, t, \ell)$, $(k, c) \in \mathsf{Enc}(\mathsf{pk}, j)$, and all n-bit string $s \in \{0, 1\}^n$ it holds that

$$\mathsf{Combine}\left(\mathsf{pk}, S, \left\{\mathsf{PDec}\left(\mathsf{sk}_{i,s_i}^{(j)}, c, j\right)\right\}_{i\in S}, c, j\right) = k.$$

We introduce an additional property of simulability of partial decryptions, which states that for honestly generated ciphertexts, the partial decryptions output by SimPD must be identical to partial decryptions generated with the left and right key. In particular, this implies that for honestly generated ciphertexts, partial decryptions with the left and right key must be equal. We summarize both properties in a single definition.

Definition 9.3 (Perfect Partial Decryption Simulation for PD-BT-KEM). We say that a PD-BT-KEM scheme achieves simulability of partial decryptions if for all $\lambda \in \mathbb{N}$, $0 < t \leq n \in \mathbb{N}$, and for all $i \in [n]$ it holds that

$$\mathsf{SimPD}\left(\mathsf{pk}, i, (c_0, c_1), j, r_{\mathsf{Enc}}\right) = \mathsf{PDec}\left(\mathsf{sk}_{i,0}^{(j)}, (c_0, c_1), j\right) = \mathsf{PDec}\left(\mathsf{sk}_{i,1}^{(j)}, (c_0, c_1), j\right),$$

for all $\mathsf{pp} \in \mathsf{Setup}(1^\lambda)$, $\left(\mathsf{pk}, \{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n],j\in[\ell]}\right) \in \mathsf{KGen}(1^\lambda, n, t, \ell)$, all $r_{\mathsf{Enc}} \in \mathcal{R}_{\mathsf{Enc}}$ for $(k, (c_0, c_1)) = \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$.

$\mathsf{Game\text{-}OSS\text{-}Left}_{\mathcal{A}}(1^\lambda)$	$\mathsf{Game\text{-}OSS\text{-}Right}_{\mathcal{A}}(1^\lambda)$
$(\ell, u, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda)$	$(\ell, u, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_1(1^\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \ell)$	$(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \ell)$
// Only left keys for position u.	// Only right keys for position u.
$\mathsf{shares} \leftarrow \left(\{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n], j\in[\ell]\setminus\{u\}}, \{\mathsf{sk}_{i,0}^{(u)}\}_{i\in[n]}\right)$	$\mathsf{shares} \leftarrow \left(\{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n], j\in[\ell]\setminus\{u\}}, \{\mathsf{sk}_{i,1}^{(u)}\}_{i\in[n]}\right)$
$r^*_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$	$r^*_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$
$(k^{(0)}, c^{(0)} := (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, u; r^*_{\mathsf{Enc}})$	$(k^{(0)}, c^{(0)} := (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, u; r^*_{\mathsf{Enc}})$
$(k^{(1)}, c^{(1)} := (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, u)$	$(k^{(1)}, c^{(1)} := (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, u)$
$b \xleftarrow{\$} \{0,1\}$	$c^* \leftarrow (c_0^{(0)}, c_1^{(1)})$
$c^* \leftarrow (c_0^{(0)}, c_1^{(b)})$	$(\mathcal{I}_{\mathsf{pd}}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_2(\mathsf{st}, \mathsf{pk}, \mathsf{shares}, c^*)$
$(\mathcal{I}_{\mathsf{pd}}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}_2(\mathsf{st}, \mathsf{pk}, \mathsf{shares}, c^*, k^{(0)})$	**if** $\mathcal{I}_{\mathsf{pd}} \not\subseteq [n] \vee \lvert\mathcal{I}_{\mathsf{pd}}\rvert \geq t$ **then return** 0
if $\mathcal{I}_{\mathsf{pd}} \not\subseteq [n] \vee \lvert\mathcal{I}_{\mathsf{pd}}\rvert \geq t$ **then return** 0	**for** $i \in \mathcal{I}_{\mathsf{pd}}$ **do**
for $i \in \mathcal{I}_{\mathsf{pd}}$ **do**	$d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, u, r^*_{\mathsf{Enc}})$
$d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, u, r^*_{\mathsf{Enc}})$	$k' \xleftarrow{\$} \mathcal{A}_3(\mathsf{st}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$
$b' \xleftarrow{\$} \mathcal{A}_3(\mathsf{st}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$	**if** $k' = k^{(0)}$ **then return** 1
return $b \stackrel{?}{=} b'$	**return** 0

Figure 8: Game-OSS-Left and Game-OSS-Right for PD-BT-KEM.

Diverging further from [11], we define our own version one-sided security in the presence of up to $t-1$ simulated partial decryptions. Crucially, we pose different requirements for left and right one-sided security.

Left-Sided Security with Simulated Partial Decryptions (LSS-SPD). At a high level, LSS-SPD states that an adversary who has only left keys (i.e., no right keys) for a chosen position $u \in [\ell]$ cannot distinguish whether or not the right side c_1^* of a given ciphertext $c^* = (c_0, c_1^*)$ fits with the left side and the key k, even if he obtains up to $t-1$ simulated partial decryptions with the randomness fitting the left side and the key. More formally, suppose that $r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$, $(k^{(0)}, c_0^{(0)}, c_1^{(0)}) \leftarrow \mathsf{Enc}(\mathsf{pk}, u; r_{\mathsf{Enc}})$, and $(k^{(1)}, c_0^{(1)}, c_1^{(1)}) \leftarrow \mathsf{Enc}(\mathsf{pk}, u)$, then an adversary who for position u only has access to left keys, cannot significantly distinguish bewteen

$$\left(k^{(0)}, (c_0^{(0)}, c_1^{(0)}), \left\{\mathsf{SimPD}\left(\mathsf{pk}, i, (c_0^{(0)}, c_1^{(0)}), u, r_{\mathsf{Enc}}\right)\right\}_{i\in\mathcal{I}_{\mathsf{pd}}}\right)$$

and

$$\left(k^{(0)}, (c_0^{(0)}, c_1^{(1)}), \left\{\mathsf{SimPD}\left(\mathsf{pk}, i, (c_0^{(0)}, c_1^{(1)}), u, r_{\mathsf{Enc}}\right)\right\}_{i\in\mathcal{I}_{\mathsf{pd}}}\right),$$

for any chosen $\mathcal{I}_{\mathsf{pd}} \subseteq [n]$ with $\lvert\mathcal{I}_{\mathsf{pd}}\rvert < t$.

Right-Sided Security with Simulated Partial Decryptions (RSS-SPD). For RSS-SPD, we instead argue that an adversary who for index u has only right keys is unable to compute the key $k^{(0)}$ fitting the left part of the ciphertext $c_0^{(0)}$ given malformed ciphertext $(c_0^{(0)}, c_1^{(1)})$, even if it obtains less than t simulated partial decryptions using the randomness r_{Enc} belonging to the left ciphertxt and key. More formally, suppose that $r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$, $(k^{(0)}, c_0^{(0)}, c_1^{(0)}) \leftarrow \mathsf{Enc}(\mathsf{pk}, u; r_{\mathsf{Enc}})$, and $(k^{(1)}, c_0^{(1)}, c_1^{(1)}) \leftarrow \mathsf{Enc}(\mathsf{pk}, u)$, then an adversary that does not have any left keys for position u cannot compute $k^{(0)}$ given

$$\left((c_0^{(0)}, c_1^{(1)}), \left\{\mathsf{SimPD}\left(\mathsf{pk}, i, (c_0^{(0)}, c_1^{(1)}), u, r_{\mathsf{Enc}}\right)\right\}_{i\in\mathcal{I}_{\mathsf{pd}}}\right),$$

except with negligible probability.

In both games, we require that the set $\mathcal{I}_{\mathsf{pd}}$ can be chosen by the adversary *adaptively*, i.e., even after seeing the challenge ciphertext. This allows us to prove an even stronger variant of traceability against *adaptive* partial insiders, i.e., where the decoder can select a set of partial insiders after seeing the ciphertext.

Definition 9.4 (One-Sided Security against Simulated Partial Decryptions (OSS-SPD)). A PD-BT-KEM scheme is said to achieve one-sided security against simulated partial decryptions if for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$

$$\Pr[\mathsf{Game\text{-}OSS\text{-}Left}_{\mathcal{A}}(1^\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \text{ and } \Pr[\mathsf{Game\text{-}OSS\text{-}Right}_{\mathcal{A}}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda),$$

where Game-OSS-Left and Game-OSS-Right are defined in Figure 8. We say that a PD-BT-KEM scheme is Left-SPD secure and Right-SPD secure when referring to security with respect to the two games, respectively.

$\mathsf{Game\text{-}IND\text{-}BT\text{-}CPA}_{\mathcal{A}}(1^\lambda)$

$b \xleftarrow{\$} \{0,1\}$
$(\ell, \mathsf{Cor}, \mathsf{st}_1) \xleftarrow{\$} \mathcal{A}_1(1^\lambda)$
if $\mathsf{Cor} \not\subseteq [n] \vee |\mathsf{Cor}| \geq t$ **then return** 0
$(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, n, t, \ell)$
$j^* \xleftarrow{\$} [\ell];\ k_0 \xleftarrow{\$} \mathcal{K}(\lambda)$
$(k_1, c^*) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j^*)$
$b' \xleftarrow{\$} \mathcal{A}_2\left(\mathsf{st}_2, c^*, k_b, \mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}}\right)$
return $b \stackrel{?}{=} b'$

Figure 9: The semantic security game of PD-BT-KEM.

Definition 9.5 (Semantic Security of PD-BT-KEM). A PD-BT-KEM scheme is said to achieve semantic security if for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$

$$\Pr[\mathsf{Game\text{-}IND\text{-}BT\text{-}CPA}_{\mathcal{A}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

where Game-IND-BT-CPA is defined in Figure 9.

9.2 A PD-BT-KEM from Pairings

We next show a construction from pairings, by slightly modifying the construction in [11, §5.2].

The key challenge in lifting BT-KEM to PD-BT-KEM is that left and right keys are correlated, so partial decryptions of one side may leak information about the other. To give intuition, we start by describing why the construction DDH-based construction in [11, §5.1] is not PD-BT-KEM.

An Attack On [11, §5.1]. We next show why the construction in [11, §5.1] does not satisfy our RSS-SPD security notion. To see this, let us consider the construction. For simplicity, we assume that $\ell = 1$. The public key is $Y = g^y, Z = g^z, X = g^\alpha$ for random $\alpha, y, z \in \mathbb{Z}_p$. The left key for the i-th party is s_i/y and the right key is s_i/z, where $s_1, \ldots, s_n$ are t-out-of-n secret shares of α. To encrypt, sample a random r and let $c_0 = Y^r, c_1 = Z^r$. The encapsulated KEM key is $k = X^r = g^{\alpha r}$.

To see why RSS-SPD security is broken, consider an *invalid ciphertext*, i.e., $c_0 = Y^r$, $c_1 = Z^{r'}$ for independent random $r, r' \in \mathbb{F}_p$, and suppose the decoder holds all right keys but no left

keys. Further, suppose that the decoder receives a simulated partial decryption $d_i = g^{s_i r}$ of the left encapsulated key on behalf of a party i, whose *right key* is s_i/z. It can then compute $d_i^{z/s_i} = g^{zr} = Z^r$ using the right key $z/s_i = (s_i/z)^{-1}$. As Z^r is an encapsulation of the KEM key $g^{\alpha r}$ in a *right ciphertext*, the adversary can then recover $g^{\alpha r}$ using only the right keys, breaking Right-SPD security.

Fortunately, this issue does not appear in the construction in [11, §5.2], which uses pairings: secret keys live in the source group while partial decryptions are target group elements, which prevents the adversary from performing the nonlinear operations on the secret keys required by the above attack. We next present our construction.

We next present our PD-BT-KEM, which is a variant of [11, §5.2] with minor modifications. In particular, we add an "individual public key" for each party to the public key and instantiate a SimPD algorithm. Modifications over the BT-KEM in [11, §5.2] are highlighted in gray.

Construction 7 (A PD-BT-KEM $\Sigma_{\mathsf{PD\text{-}BT}}$)**.** Let the ensemble $E = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, p) \xleftarrow{\$} \mathcal{G}(1^\lambda)$ be public parameters.

$\Sigma_{\mathsf{PD\text{-}BT}}.\mathsf{KGen}(1^\lambda, n, t, \ell)$ Sample $\alpha, y, z \xleftarrow{\$} \mathbb{Z}_p$ as well as a Shamir t-out-of-n sharing $s_1, \ldots, s_n$ of α and compute $X \leftarrow g_1^{\alpha yz}$, $\{X_i \leftarrow e(g_1, g_2)^{s_i yz}\}_{i\in[n]}$, $Y \leftarrow g_1^y$, and $Z \leftarrow g_1^z$. Set the public key to $\mathsf{pk} := (X, \{X_i\}_{i\in[n]}, Y, Z)$.[16] For all $i \in [n]$ and $j \in [\ell]$, do the following

- Compute $k_0^{(0)} \leftarrow H(j)^{zs_i}$ and $k_1^{(0)} \leftarrow g_2^{zs_i}$. Assign $\mathsf{sk}_{i,0}^{(j)} \leftarrow (k_0^{(0)}, k_1^{(0)}) \in \mathbb{G}_1 \times \mathbb{G}_2$.
- Compute $k_0^{(1)} \leftarrow H(j)^{ys_i}$ and $k_1^{(1)} \leftarrow g_2^{ys_i}$. Assign $\mathsf{sk}_{i,1}^{(j)} \leftarrow (k_0^{(1)}, k_1^{(1)}) \in \mathbb{G}_1 \times \mathbb{G}_2$.

Output $(\mathsf{pk}, \{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n],j\in[\ell]})$.

$\Sigma_{\mathsf{PD\text{-}BT}}.\mathsf{Enc}(\mathsf{pk}, j)$**:** Parse pk as $(X, *, Y, Z)$ and do the following:

- Sample $r, t_0, t_1 \xleftarrow{\$} \mathbb{Z}_p$.
- Set $k \leftarrow e(X, g_2)^r$ $[= e(g_1, g_2)^{\alpha yzr}]$.
- Compute $c_0 \leftarrow (g_2^{t_0}, Y^r \cdot H(j)^{t_0})$ and $c_1 \leftarrow (g_2^{t_1}, Z^r \cdot H(j)^{t_1})$.
- Output $(k, (c_0, c_1))$.

$\Sigma_{\mathsf{PD\text{-}BT}}.\mathsf{PDec}(\mathsf{sk}_{i,b}^{(j)}, (c_0, c_1), j)$**:** Parse $\mathsf{sk}_{i,b}^{(j)}$ as (k_0, k_1) and c_b as (u, v). Then, compute and return

$$d_i \leftarrow \frac{e(v, k_1)}{e(k_0, u)} \; [= e(g_1, g_2)^{s_i yzr}].$$

$\Sigma_{\mathsf{PD\text{-}BT}}.\mathsf{Combine}(j, c, S, \{d_i\}_{i\in[S]})$**:** Compute the Lagrange coefficients $\{\lambda_{i,S}^{(0)}\}_{i\in S}$ for interpolating among set S at position 0. Then, compute and output $k \leftarrow \prod_{i\in S} d_i^{\lambda_{i,S}^{(0)}}$.

$\Sigma_{\mathsf{PD\text{-}BT}}.\mathsf{SimPD}(\mathsf{pk}, i, (c_0, c_1), j, r_{\mathsf{Enc}})$**:** Parse X_i $[= e(g_1, g_2)^{s_i yz}]$ from the public key and r_{Enc} as (r, t_0, t_1). Then, compute and output $d_i \leftarrow X_i^r$ $[= e(g_1, g_2)^{s_i yzr}]$.

Theorem 9.6 *The construction $\Sigma_{\mathsf{PD\text{-}BT}}$ is a secure* PD-BT-KEM *scheme under the two variants of the ABDDH assumption ([11, Definition 8], and Assumption 2), and the t-ABCDH assumption (Assumption 1).*

We present a proof of Theorem 9.6 in Appendix F. In our proof, we introduce the relevant assumptions. In particular, we show that the t-ABCDH assumption holds in the GGM.

[16] In [11, Section 5.2] the public key does not contain the X_i elements. Note that we only use these elements for SimPD, hence encrypting users do not need to store them.

9.3 From PD-BT-KEM to Traceability with Maximal Partial Insider Resilience

In this section, we present a transformation from PD-BT-KEM to a TT-KEM that is *natively* resilient against $t-1$ partial insiders. There is, however, one caveat: We can only guarantee partial insider resilience against *decrypting decoders*, i.e., decoders that do not just distinguish whether a given ciphertext-key pair (k, c) is correct, but instead, given a ciphertext c, output the corresponding KEM-key k. To this end, we adjust the traceability definition (Definition 4.1) to instead use the following version of the event GoodDec within this section:

Definition 9.7 (GoodDec for Decrypting Decoders (GoodDDec)). For $\varepsilon \in (0,1]$, a TT-KEM Π, a public key pk, secret keys $\{\mathsf{sk}_i\}_{i\in[n]}$, and a set of partial insiders $\mathcal{I}_{\mathsf{pd}} \subseteq [n]$, we say that a decrypting decoder D is good (i.e., $\mathsf{GoodDDec} = 1$), exactly if $P^{\mathsf{Dec}}(D) \geq \varepsilon$, where

$$P^{\mathsf{Dec}}(D) \coloneqq \Pr\left[D(c, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}) = k \colon \begin{array}{c}(k,c) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}) \\ \forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{PDec}(\mathsf{sk}_i, c)\end{array}\right].$$

Similar to [11], our transformation requires a collusion-resistant fingerprinting code.

Definition 9.8 (Noisy Words and Feasibility). For $\ell \in \mathbb{N}$, a word of length ℓ is a binary string $w \in \{0,1\}^\ell$. For $j \in [\ell]$, we denote with w_j the j'th bit of a word w. A *noisy* word is a ternary string $w \in \{0, 1, \text{'?'}\}$. For a set of words $W = \{w^{(1)}, \dots w^{(f)}\}$, we say that a noisy word w is feasible for W, if for all $j \in [\ell]$ either $w_j =$ '?' or there exists some $i \in [f]$ such that $w_j = w_j^{(i)}$. Denote with $F(W)$ the set of words that is feasible for W. For $0 \leq \delta \leq 1$, we say that a word is δ-noisy or at most δ-noisy, if it contains '?' in at most $\delta \cdot \ell$ positions. We denote with $F_\delta(W) = \{w \in F(W) \colon w \text{ is } \delta\text{-noisy}\}$ the set of words that are at most δ-noisy and feasible for W.

$\mathsf{Game\text{-}FC}_{\mathcal{A}}(1^\lambda, n, \delta)$	Oracle $\mathcal{O}^{\mathsf{codeword}}(i)$
$\mathsf{Cor} \leftarrow \emptyset$	**if** $i \notin [n]$ **then return** $\perp$
$\left(\Gamma = \{\mathsf{cw}^{(i)}\}_{i\in[n]}, \ell, \mathsf{tk}\right) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, n, \delta)$	$\mathsf{Cor} \leftarrow \mathsf{Cor} \cup \{i\}$
$w \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{codeword}}}(1^\lambda, n, \delta)$	**return** $\mathsf{cw}^{(i)}$
if $w \notin F_\delta(\{\mathsf{cw}^{(i)}\}_{i\in\mathsf{Cor}})$ **then return** 0	
$S \leftarrow \mathsf{Trace}(\mathsf{tk}, w)$	
if $S = \emptyset \vee S \not\subseteq \mathsf{Cor}$ **then return** 1	
return 0	

Figure 10: Full Collusion-Resistance experiment for fingerprinting codes

Definition 9.9 (Collusion-Resistant Fingerprinting Codes). A fingerprinting code $\mathcal{F}$ is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Trace})$ with the following syntax:

- $(\Gamma \coloneqq \{\mathsf{cw}^{(i)}\}_{i\in[n]}, \ell, \mathsf{tk}) \xleftarrow{\$} \mathsf{Gen}(1^\lambda, n, \delta)$. Gen is a code generation algorithm that receives the security parameter λ, the number of words $n > 0$ and a noise bound δ. It outputs a set of n words $\Gamma = \{\mathsf{cw}^{(i)}\}_{i\in[n]}$, the codeword-length ℓ, and a tracing key tk, such that $\mathsf{cw}^{(i)} \in \{0,1\}^\ell$, where $\ell(\lambda, n, \delta)$ is a function of λ, δ, and n.
- $S \leftarrow \mathsf{Trace}(\mathsf{tk}, w)$. Given the tracing key tk and a noisy word w, the tracing algorithm outputs a subset $S \subseteq [n]$.

We say that a fingerprinting code $\mathcal{F}$ is fully collusion-resistant if there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all PPT adversaries $\mathcal{A}$, all $\lambda, n \in \mathbb{N}$, and all $\delta \in [0,1]$ it holds that

$$\Pr\left[\mathsf{Game\text{-}FC}^{\mathcal{F}}_{\mathcal{A}}(1^\lambda, n, \delta) = 1\right] \leq \mathsf{negl}(\lambda),$$

where Game-FC is defined in Figure 10.

Construction 8. Let $\mathcal{F}$ be a fingerprinting code and $\Pi_{\mathsf{PD\text{-}BT}}$ be a PD-BT-KEM scheme. The resulting TT-KEM scheme Π is constructed as follows.

$\Pi.\mathsf{KGen}(1^\lambda, n, t, \varepsilon \geq 2/\sqrt{\lambda})$: Let $\delta \leftarrow (1-\varepsilon)/(1-\frac{2}{\sqrt{\lambda}})$ and sample a fingerprinting code $(\Gamma, \ell, \mathsf{tk}) \xleftarrow{\$} \mathcal{F}.\mathsf{Gen}(1^\lambda, n, \delta)$. Generate PD-BT-KEM keys as

$$\left(\mathsf{pk}, \{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n], j\in[\ell]}\right) \xleftarrow{\$} \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{KGen}(1^\lambda, n, t, \ell)$$

and assign secret keys according to parties'codewords in Γ: For all $i \in [n]$ assign $\mathsf{sk}_i \leftarrow \left\{\mathsf{sk}_{i,\mathsf{cw}_j^{(i)}}^{(j)}\right\}_{j\in[\ell]}$. Output $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk})$.

$\Pi.\mathsf{Enc}(\mathsf{pk})$**:** Sample $j \xleftarrow{\$} [\ell]$ and encrypt $(k, (c_0, c_1)) \xleftarrow{\$} \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{Enc}(\mathsf{pk}, j)$. Output $(k, ((c_0, c_1), j)$.

$\Pi.\mathsf{PDec}(\mathsf{sk}_i, ((c_0, c_1), j))$**:** Output $d_i \leftarrow \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{PDec}\left(\mathsf{sk}_{i,\mathsf{cw}_j^{(i)}}^{(j)}, (c_0, c_1), j\right)$.

$\Pi.\mathsf{Combine}(S, \{d_i\}_{i\in S}, ((c_0, c_1), j))$**:** Compute and output

$$k \leftarrow \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{Combine}(S, \{d_i\}_{i\in S}, ((c_0, c_1), j)).$$

$\mathsf{Tr}^D(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, b)$**:** We define the subroutine Tr used in the tracing algorithm. Initialize a counter $ctr \leftarrow 0$ and repeat Q times:

- Sample $r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$.
- Compute $(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$.
- Sample $(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{Enc}(\mathsf{pk}, j)$.
- Assemble $c^* \leftarrow (c_0^{(0)}, c_1^{(b)})$.
- Compute $\{d_i \xleftarrow{\$} \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r_{\mathsf{Enc}})\}_{i\in\mathcal{I}_{\mathsf{pd}}}$.
- If $D(c^*, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}) = k^{(0)}$, increment $ctr \leftarrow ctr + 1$.

After Q iterations, output ctr.

$\Pi.\mathsf{Trace}^D(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{pd}})$**:** If $\mathcal{I}_{\mathsf{pd}} \not\subseteq [n]$ or $|\mathcal{I}_{\mathsf{pd}}| \geq t$, abort and output $\emptyset$. Otherwise, continue as follows: Set $Q \leftarrow 8\lambda^2$. Initialize an ℓ-length word w. For all $j \in [\ell]$, do:

- Compute $p_j \xleftarrow{\$} \mathsf{Tr}^D(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, 0)$.
- Compute $q_j \xleftarrow{\$} \mathsf{Tr}^D(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, 1)$.
- Assign the j'th position of the word w as

$$w_j \leftarrow \begin{cases} 0 & \text{if } q_j \neq 0, \\ 1 & \text{if } q_j = 0 \text{ and } p_j/Q \geq 1/\sqrt{\lambda}, \\ ? & \text{else.} \end{cases}$$

Finally, if w is not δ-noisy output $\mathcal{T}' \leftarrow \emptyset$. Else, compute $\mathcal{T}' \xleftarrow{\$} \mathcal{F}.\mathsf{Trace}(\mathsf{tk}, w)$. If $\mathcal{T}' \neq \emptyset$, output $\mathcal{T}'$. Else, output $[n]$.

Checking for Empty Traces. Observe that the Trace algorithm of Construction 8 only ever outputs $\mathcal{T}' = \emptyset$ if the word w produced during tracing is not δ-noisy. Hence, checking whether tracing a given decoder D returns an empty set, and thus simulating $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ only requires one to compute w and check whether it is δ-noisy or not. Crucially, this can be done *without* access to the tracing key tk.

Theorem 9.10 *If the underlying* PD-BT-KEM *is two-sided correct, has simulable partial decryptions, is OSS-SPD secure, and semantically secure, then Construction 8 is a perfect* 1*-traceable* TT-KEM *against* $t-1$ *partial insiders with respect to decrypting decoders with limit* $\varepsilon > 2/\sqrt{\lambda}$*. Traceability holds in the presence of an* $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$ *and an* $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ *oracle.*

Remark 9.11 (Adaptive Partial Insider Resilience). Interestingly, we can even prove resilience against *adaptive* partial insiders: I.e., we can prove partial insider resilience in a model, where the adversary is not required to fix $\mathcal{I}_{\mathsf{pd}}$ once initially, but where the decoder can adaptively ask for a set $\mathcal{I}_{\mathsf{pd}}$ of its choice in every query after seeing the queried ciphertext c. This is possible because the left and right OSS-SPD security games allow our reduction to query adaptively for a set $\mathcal{I}_{\mathsf{pd}}$, on behalf of which to obtain simulated partial decryptions, *after seeing the challenge ciphertext*. We believe that properly formalizing and exploring adaptive partial insider resilience is a very interesting direction for future work, especially when motivated by the context-dependent scenario [7], where some honest parties might regularly release partial decryptions.

Since we only modified the Trace algorithm, and did not modify correctness or semantic security requirements of PD-BT-KEM compared to standard BT-KEM, we refer to [11] for proofs of correctness, robustness, and semantic security. Next, we prove 1-traceability against $t-1$ partial insiders.

9.4 Traceability Proof for Construction 8

Proof of Theorem 9.10. To prove 1-traceability against $t-1$ partial insiders of Construction 8, we proceed in several steps, which we introduce next at a high level.

First, we will modify the experiment underpinning the event GoodDDec, replacing honestly computed partial decryptions with partial decryptions generated by the SimPD algorithm of the underlying PD-BT-KEM, which, for honestly generated ciphertexts, are identical to normal partial decryptions (Definition 9.3). Hence, in the following steps, we can assume that the decoder is good with respect to *simulated* partial decryptions.

Then we will argue that any decoder that is at least ε-good for $\varepsilon \geq 2/\sqrt{\lambda}$, must also be at least $2/\sqrt{\lambda}$-good for at least a fraction $(1-\delta)$ of positions $j \in [\ell]$, where δ is chosen as in Construction 8. This allows us to argue that whenever the adversary $\mathcal{A}$ produces a good decoder, the word w produced by the Trace algorithm can have a '?' in at most a fraction δ of positions (i.e., the produced word is at most δ-noisy). This holds because the Trace algorithm only ever sets $w_j \leftarrow ?$ if the decoder fails to correctly decrypt at least a fraction of $1/\sqrt{\lambda}$ of honestly generated ciphertexts ($(p/Q) \geq 1/\sqrt{\lambda}$), and any decoder that is good for a given position j can only fail to do so with negligible probability. This essentially demonstrates that the word w produced by Trace is sufficiently noiseless for tracing using the underlying fingerprinting code.

Finally, we must show that the word w is *feasible* for the set of traitors corrupted by the adversary $\mathcal{A}$. We do so in two steps: First, we show by reduction to Right-SPD security of the underlying PD-BT-KEM that whenever the traitors have *only right keys* for a position $j \in [\ell]$, they cannot trick the Trace algorithm to set $w_j \leftarrow 0$. Then, by reduction to Left-SPD security of the PD-BT-KEM, we show that, vice-versa, if the traitors have *only left keys* for a position $j \in [\ell]$ they cannot trick the Trace algorithm to set $w_j \leftarrow 1$. Note that in the reduction to both properties, the reduction receives simulated partial decryptions for the challenge ciphertext on behalf of the set $\mathcal{I}_{\mathsf{pd}}$, which allows it to simulate tracing as required by the Tr subroutine.

By the above steps, it must hold that whenever the adversary produces a good decoder, the word produced by Trace is (a) at most δ-noisy, and (b) must be feasible for the set of traitors

corrupted by the adversary, and thus the tracing procedure of the fingerprinting code will identify at least one traitor. Crucially, this holds despite the fact that the adversary obtains access to the $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle prior to tracing, and that the decoder receives partial decryptions for up to $t-1$ partial insiders. In the following, we formalize the approach outlined above.

Similar to the event $\mathsf{GoodDDec}$, for a given tuple $(\mathsf{pk}, D, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in[\mathcal{I}_{\mathsf{pd}}]})$, define the event $\mathsf{GoodDecSim}$ to be 1, exactly if $P^{\mathsf{Sim}}(D) \geq \varepsilon$, where $P^{\mathsf{Sim}}(D)$ is defined as

$$\Pr\left[D(((c_0,c_1),j),\{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}) = k\colon \begin{array}{c} j \xleftarrow{\$} [\ell];\ r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}} \\ (k,(c_0,c_1)) \leftarrow \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}}) \\ \forall i \in \mathcal{I}_{\mathsf{pd}}\colon \\ d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, (c_0,c_1), j, r_{\mathsf{Enc}}) \end{array}\right],$$

and Enc and SimPD refer to the algorithms of $\Pi_{\mathsf{PD\text{-}BT}}$. Note that due to the perfect partial decryption simulability of $\mathsf{PD\text{-}BT\text{-}KEM}$ (Definition 9.3), it holds that the events $\mathsf{GoodDDec}$ and $\mathsf{GoodDecSim}$ are *identical*. Henceforth, we will consider traceability with respect to the event $\mathsf{GoodDecSim}$ instead of $\mathsf{GoodDDec}$.

Next, we argue that tracing any ε-good decoder will produce a word w that is at most δ-noisy (i.e., has $w_j = ?$ in at most $\delta\ell$ positions $j \in [\ell]$). Denote with $w(\mathcal{A})$ the random variable for the word produced in the $\mathsf{Game\text{-}Tr}$ experiment instantiated with Construction 8 against a given adversary $\mathcal{A}$. For a word w, denote with $\delta\text{-}\mathsf{noisy}(w)$ the event that w is at most δ-noisy.

Lemma 9.12 *For all PPT adversaries $\mathcal{A}$, it holds that*

$$\Pr[\neg\delta\text{-}\mathsf{noisy}(w(\mathcal{A}))|\mathsf{GoodDecSim}_{\mathcal{A}}]] \leq \ell \cdot \exp(-\lambda)$$

Proof of Lemma 9.12. First, we argue that any ε-good decoder must also be "at least $(2/\sqrt{\lambda})$-good for at least a fraction $(1-\delta)$ of positions $j \in [\ell]$". Fix any $\varepsilon \geq 2/\sqrt{\lambda}$ and let $\delta = (1-\varepsilon)/(1-\frac{2}{\sqrt{\lambda}})$. We say that a decoder D is good for a position j, if $P_j^{\mathsf{Sim}}(D) \geq 2/\sqrt{\lambda}$, where P_j^{Sim} is the same as P^{Sim} except that we fix j.

Claim 9.13 For all decoders D with $P^{\mathsf{Sim}}(D) \geq \varepsilon$, there must exist at least $(1-\delta)\cdot\ell$ positions $j \in [\ell]$ such that $P_j^{\mathsf{Sim}} \geq 2/\sqrt{\lambda}$.

Proof of Lemma 9.13. Assume towards contradiction that $P_j^{\mathsf{Sim}}(D) < 2/\sqrt{\lambda}$ for a fraction of more than δ positions $j \in [\ell]$. In this case, we can bound

$$\begin{aligned} P^{\mathsf{Sim}}(D) &< \delta \cdot \frac{2}{\sqrt{n}} + (1-\delta)\cdot 1 \\ &= 1 + \delta \cdot \left(\frac{2}{\sqrt{\lambda}} - 1\right) \\ &= 1 + \left(\frac{\varepsilon - 1}{\frac{2}{\sqrt{\lambda}} - 1}\right) \cdot \left(\frac{2}{\sqrt{\lambda}} - 1\right) \\ &= 1 + \varepsilon - 1 \\ &= \varepsilon, \end{aligned}$$

which contradicts $P^{\mathsf{Sim}}(D) \geq \varepsilon$. ∎

Equipped with Claim 9.13, we inspect how w is computed in the tracing procedure. In particular, observe that w_j is only ever set to '?' whenever $p_j/Q < 1/\sqrt{\lambda}$, where p_j is the result of the p-experiment in the j-th iteration of Trace $(p_j \leftarrow \mathsf{Tr}^D(\mathsf{pk}, j, Q, 0))$, which is essentially counting the number of successful decryptions by the decoder, when run in an experiment identical to that of P_j^{Sim} across Q repetitions. Next, we show that for any index $j \in [\ell]$ where the decoder is good (i.e., $P_j^{\mathsf{Sim}}(D) \geq 2/\sqrt{\lambda}$), we can bound the probability of $p_j/Q < 1/\sqrt{\lambda}$ occurring to be negligible by an appropriate choice of Q.

Claim 9.14 For all $j \in [\ell]$ and all decoders D with $P_j^{\mathsf{Sim}}(D) \geq 2/\sqrt{\lambda}$ it holds that

$$\Pr[p_j/Q < 1/\sqrt{\lambda}] \leq \exp(-Q/4\sqrt{\lambda}).$$

Proof of Claim 9.14. The proof follows a simple Chernoff bound. ∎

To summarize, we know that all ε-good decoders must be good for at least $(1-\delta)$ positions, and we will only set $w_j = ?$ for a good position with probability at most $\exp(-Q/4\sqrt{\lambda})$. We apply a union bound over the maximum number of good positions ℓ and conclude that

$$\Pr\left[\neg\delta\text{-}\mathsf{noisy}(w(\mathcal{A}))|\mathsf{GoodDecSim}\right] \leq \ell \cdot \exp\left(\frac{-Q}{4\sqrt{\lambda}}\right) \overset{Q \geq 4\lambda^{3/2}}{\leq} \ell \cdot \exp(-\lambda).$$

∎

Next, we use game hops to make two modifications to the Tr subroutine executed during tracing in $\mathsf{Game\text{-}Tr}$. We call the modified Tr subroutines Tr' and Tr'' respectively. Similar to $w(\mathcal{A})$, we denote with $w'(\mathcal{A})$ and $w''(\mathcal{A})$ the random varable for the word produced when tracing $\mathcal{A}$ making use of Tr' and Tr'' respectively. Further, denote with the random variable $F(\mathcal{T}_\mathcal{A})$ the set of words that are feasible for the set of traitors $\mathcal{T}_\mathcal{A}$ corrupted by $\mathcal{A}$ with respect to their codewords $\{w^{(i)}\}_{i\in\mathcal{T}_\mathcal{A}}$. We will show that our modifications do not noticeably decrease the probability that $\mathcal{A}$ manages to cause the tracing algorithm to produce a codeword that is not feasible for $\mathcal{T}_\mathcal{A}$. More formally, we will show that by reduction to SPD-security of $\Pi_{\mathsf{PD\text{-}BT}}$ that

$$\Pr[w(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] - \Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] \leq \mathsf{negl}(\lambda)$$

as well as

$$\Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] - \Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] \leq \mathsf{negl}(\lambda).$$

Before we introduce our modifications in more detail, recall that for all positions j, the tracing algorithm runs the Tr subroutine *twice*. Once with input $b = 0$, testing how many *valid* ciphertexts the decoder decrypts correctly, and assigning the result to p_j, and once with input $b = 1$, testing how many *invalid* ciphertexts of the form $(c_0^{(0)}, c_1^{(1)})$ the decoder decrypts to $k^{(0)}$, assigning the result to q_j. In both experiments, the decoder receives simulated partial decryptions for its respective ciphertexts.

Subroutine Tr': In this step, we modify the subroutine Tr when executed in $\mathsf{Game\text{-}Tr}$ for $b = 1$: Upon input (pk, j, Q, b), if $b = 1$ and all traitors $\mathcal{T}_\mathcal{A}$ have a right key for position j (i.e., $\mathsf{cw}_j^{(i)} = 1$ for all $i \in \mathcal{T}_\mathcal{A}$), set $ctr = q_j = 0$. This change essentially enforces $q_j = 0$ for all positions j where all traitors have right keys, which prevents us from ever setting $w_j = 0$ (indicating that traitors would have (some) left keys in position j) whenever traitors have only right keys in position j. The changes are laid out in detail in Figure 11.

Claim 9.15 For all PPT adversaries $\mathcal{A}$ there exists a PPT reduction $\mathcal{B}$ such that

$$\Pr[w(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] - \Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] \leq \ell \cdot Q \cdot \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Right}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda).$$

Proof of Claim 9.15. Fix an arbitrary PPT adversary $\mathcal{A}$ against $\mathsf{Game\text{-}Tr}$. We construct a reduction $\mathcal{B}$ that uses $\mathcal{A}$ to win against $\mathsf{Game\text{-}OSS\text{-}Right}$ instantiated with $\Pi_{\mathsf{PD\text{-}BT}}$. Initially, $\mathcal{B}$ chooses a random $u \in [\ell]$ and outputs (ℓ, u) to its challenger. As a result, $\mathcal{B}$ receives a public key pk as well as *all* secret keys except for the left keys in position u:

$$\mathsf{shares} = \left(\{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n], j\in[\ell]\setminus\{u\}}, \{\mathsf{sk}_{i,1}^{(u)}\}_{i\in[n]}\right)$$

$\mathcal{B}$ generates the code $\Gamma = \{\mathsf{cw}^{(i)}\}_{i\in[n]}$ and runs $\mathcal{A}(\mathsf{pp}, \mathsf{pk})^{\mathcal{O}^{\mathsf{corrupt}}, \mathcal{O}^{\mathsf{PDec\text{-}Hon}}, \mathcal{O}^{\mathsf{Trace\text{-}Empty}}}$, simulating the oracles as follows:

Subroutine $\mathsf{Tr}(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, \mathbf{b} = \mathbf{1})$	Subroutine $\mathsf{Tr}'(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, \mathbf{b} = \mathbf{1})$
$q_j \leftarrow 0$	$q_j \leftarrow 0$
repeat Q **times**	**repeat** Q **times**
$r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$	$r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$
$(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$	$(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$
$(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j)$	$(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j)$
$c^* \leftarrow (c_0^{(0)}, c_1^{(1)})$	$c^* \leftarrow (c_0^{(0)}, c_1^{(1)})$
$\forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r_{\mathsf{Enc}})$	$\forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r_{\mathsf{Enc}})$
if $D(c^*, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}) = k^{(0)}$ **then**	**if** $D(c^*, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}) = k^{(0)}$ **then**
$q_j \leftarrow q_j + 1$	$q_j \leftarrow q_j + 1$
	if $\forall i \in \mathcal{T}_{\mathcal{A}} : \mathsf{cw}_j^{(i)} = 1$ **then**
	$q_j \leftarrow 0$
return q_j	**return** q_j

Figure 11: The Tr and Tr' subroutine for $b = 1$. The algorithms Enc and SimPD are invoked with respect to $\Pi_{\mathsf{PD\text{-}BT}}$. Modifications are highlighted in gray.

Simulating $\mathcal{O}^{\mathsf{corrupt}}(i)$: $\mathcal{B}$ checks whether party i is supposed to have a left key in position u according to Γ (i.e, whether $\mathsf{cw}_u^{(i)} = 0$). If this is the case, $\mathcal{B}$ aborts and returns a random group element to its challenger. Otherwise, $\mathcal{B}$ knows all secret keys for i and forwards $\left\{\mathsf{sk}^{(j)}_{i,\mathsf{cw}_j^{(i)}}\right\}_{j \in [\ell]}$ to $\mathcal{A}$.

Simulating $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}})$: $\mathcal{B}$ parses the encryption randomness r_{Enc} as $(j, r_{\mathsf{Enc}}^{\Pi_{\mathsf{PD\text{-}BT}}})$ and computes $(k, (c_0, c_1)) \leftarrow \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$. Then it computes and returns a partial decryption $d_i \leftarrow \Pi_{\mathsf{PD\text{-}BT}}.\mathsf{PDec}(\mathsf{sk}_{i,1}^{(j)}, (c_0, c_1), j)$. Note that $\mathcal{B}$ can compute this, because it knows *all* right keys $\mathsf{sk}_{i,1}^{(j)}$ for all $i \in [n]$ and $j \in [\ell]$. Further, note that $\mathcal{B}$ always uses a right key for simulating partial decryption, even if party i is supposed to have a left key for index j. Nevertheless, $\mathcal{B}$'s simulation is perfect, because of the equality-of-partial-decryptions property of $\Pi_{\mathsf{PD\text{-}BT}}$ (Definition 9.3), which states that left and right partial decryptions are equal for honestly generated ciphertexts.

Simulating $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}(\widetilde{D}, \widetilde{\mathcal{I}_{\mathsf{pd}}})$: To simulate $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}$, $\mathcal{B}$ runs $\mathsf{Trace}^{\widetilde{D}}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \widetilde{\mathcal{I}_{\mathsf{pd}}})$ up to learning the word $\tilde{w}$. If $\tilde{w}$ is not δ-noisy, $\mathcal{B}$ outputs 1, otherwise, $\mathcal{B}$ outputs 0.

As a result, $\mathcal{B}$ receives a decoder D and a set of insiders $\mathcal{I}_{\mathsf{pd}}$. W.l.o.g., assume that $\mathcal{I}_{\mathsf{pd}} \subseteq [n]$ and $|\mathcal{I}_{\mathsf{pd}}| < t$. $\mathcal{B}$ forwards $\mathcal{I}_{\mathsf{pd}}$ to its challenger and, as a result, receives a challenge ciphertext $c^* = (c_0^{(0)}, c_1^{(1)})$ as well as a set of simulated partial decryptions $\{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}$. Finally, $\mathcal{B}$ queries $D((c^*, u), \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}})$ to receive a key k, which $\mathcal{B}$ returns to its challenger.

Analysis: Observe that $w(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})$ but $w'(\mathcal{A}) \in F(\mathcal{T}_{\mathcal{A}})$ can only ever happen, if the adversary $\mathcal{A}$ triggers the modified condition, i.e. if there exists an index $j \in [\ell]$ such that $\mathsf{cw}_j^{(i)} = 1$ for all $i \in \mathcal{T}_{\mathcal{A}}$, for which q_j is not already 0 with respect to Tr. That means, if $\mathcal{A}$ distinguishes, it must happen that in at least one out of the Q iterations of Tr that

$$D\left((c_0^{(0)}, c_1^{(1)}, \left\{d_i \xleftarrow{\$} \mathsf{SimPD}(\mathsf{pk}, i, (c_0^{(0)}, c_1^{(1)}), j, r_{\mathsf{Enc}})\right\}_{i \in \mathcal{I}_{\mathsf{pd}}}\right) = k^{(0)}, \tag{4}$$

where $r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$, $(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{PD\text{-}BT\text{-}KEM.Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$, as well as $(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{PD\text{-}BT\text{-}KEM.Enc}(\mathsf{pk}, j)$. In this case, $\mathcal{B}$ will guess u to be this j with probability at least $1/\ell$.

Further, $\mathcal{B}$ will not abort (due to inability to simulate $\mathcal{O}^{\mathsf{corrupt}}$), since, by definition, all traitors in $\mathcal{T}_{\mathcal{A}}$ must have right keys. Finally, when $\mathcal{B}$ queries the decoder D on its challenge ciphertext and the corresponding simulated partial decryptions obtained by its Game-OSS-Right challenger, he simulates exactly the experiment in the above Equation 4 and wins, whenever the decoder succeeds in its experiment. As a result, we can bound

$$\Pr[w(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] - \Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] \leq \ell \cdot Q \cdot \Pr\left[\mathsf{Game\text{-}OSS\text{-}Right}_{\mathcal{B}}^{\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) = 1\right].$$

■

Subroutine Tr'': In this step, we add an additional modification to the Tr' subroutine. Upon input (pk, j, Q, b), if $b = 0$ and all traitors $\mathcal{T}_{\mathcal{A}}$ have a left key for position j (i.e., $\mathsf{cw}_j^{(i)} = 0$ for all $i \in \mathcal{T}_{\mathcal{A}}$), we change the experiment to be equivalent to that of $b = 1$. In particular, if the above condition is met, we query the adversary with *invalid* ciphertexts instead of valid ones to determine p_j. Intuitively, this prevents us from setting $w_j \leftarrow 1$ for some j, where all traitors have left keys. To see this, observe that we only ever set $w_j \leftarrow 1$, whenever $q_j = 0$ and $p_j/Q \geq 1/\sqrt{\lambda}$. In Tr'', however, q_j and p_j are derived from an identical experiment, whenever all traitors have left keys. Consequently, we will argue that the results of the experiments can only differ by that much with negligible probability for an appropriate choice of the number of repetitions Q. A detailed description of our modifications is shown in Figure 12.

Subroutine $\mathsf{Tr}'(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, \mathbf{b} = \mathbf{0})$	Subroutine $\mathsf{Tr}''(\mathsf{pk}, j, Q, \mathcal{I}_{\mathsf{pd}}, \mathbf{b} = \mathbf{0})$
$p_j \leftarrow 0$	$p_j \leftarrow 0$
repeat Q **times**	**repeat** Q **times**
$r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$	$r_{\mathsf{Enc}} \xleftarrow{\$} \mathcal{R}_{\mathsf{Enc}}$
$(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$	$(k^{(0)}, (c_0^{(0)}, c_1^{(0)})) \leftarrow \mathsf{Enc}(\mathsf{pk}, j; r_{\mathsf{Enc}})$
$(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j)$	$(k^{(1)}, (c_0^{(1)}, c_1^{(1)})) \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, j)$
	if $\forall i \in \mathcal{T}_{\mathcal{A}} : \mathsf{cw}_j^{(i)} = 0$ **then**
	$c^* \leftarrow (c_0^{(0)}, c_1^{(1)})$
	else
$c^* \leftarrow (c_0^{(0)}, c_1^{(0)})$	$c^* \leftarrow (c_0^{(0)}, c_1^{(0)})$
$\forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r_{\mathsf{Enc}})$	$\forall i \in \mathcal{I}_{\mathsf{pd}} \colon d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r_{\mathsf{Enc}})$
if $D(c^*, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}) = k^{(0)}$ **then**	**if** $D(c^*, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}}) = k^{(0)}$ **then**
$p_j \leftarrow p_j + 1$	$p_j \leftarrow p_j + 1$
return p_j	**return** p_j

Figure 12: The Tr' and Tr'' subroutine for $b = 0$. The algorithms Enc and SimPD are invoked with respect to $\Pi_{\mathsf{PD\text{-}BT}}$. Modifications are highlighted in gray.

Claim 9.16 For all PPT adversaries $\mathcal{A}$ there exists a PPT reduction $\mathcal{B}$ such that

$$\Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] - \Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] \leq 2\ell \cdot Q \cdot \mathsf{Adv}_{\mathcal{B}, \Pi_{\mathsf{PD\text{-}BT}}}^{\mathsf{Game\text{-}OSS\text{-}Left}}(1^\lambda) + \mathsf{negl}(\lambda).$$

Intuitively, the claim follows from Left-SPD security of $\Pi_{\mathsf{PD\text{-}BT}}$, because an adversary with only left keys should not be able to distinguish valid ciphertexts from ciphertexts with a malformed right component, even in the presence of simulated partial decryptions. A formal proof, however, is significantly more involved.

Proof of Claim 9.16. First, we introduce a variant of Game-OSS-Left that will simplify the upcoming analysis. Denote with Game-OSS-Left-Q an experiment that is identical to Game-OSS-Left,

except that it returns Q challenge ciphertexts instead of one. A simple hybrid argument shows that for all PPT adversaries $\mathcal{A}$ there exists a PPT reduction $\mathcal{B}$ such that

$$\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left\text{-}}Q}_{\mathcal{A},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) \leq Q \cdot \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{A},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda).$$

In the following, we will reduce Claim 9.16 to $\mathsf{Game\text{-}OSS\text{-}Left}$-$Q$-security of $\Pi_{\mathsf{PD\text{-}BT}}$.

Let $\mathcal{A}$ be an arbitrary PPT adversary against $\mathsf{Game\text{-}Tr}$. Denote with $p'_j(\mathcal{A})$ the random variable representing p_j as it is computed, when $\mathcal{A}$ is run in $\mathsf{Game\text{-}Tr}$ with respect to Tr'. Similarly, denote with $p''_j(\mathcal{A})$ the random variable representing p_j as it is computed, when $\mathcal{A}$ is run in $\mathsf{Game\text{-}Tr}$ with respect to Tr''. Henceforth, we say that a position $j \in [\ell]$ is *relevant*, if the adversary hits the modified condition in Tr'', i.e., all traitors have left keys ($\mathsf{cw}^{(i)}_j = 0$ for all $i \in \mathcal{T}_\mathcal{A}$) at position j. Now, observe that the event that $w'(\mathcal{A}) \notin F(\mathcal{T}(\mathcal{A}))$, but $w''(\mathcal{A}) \in F(\mathcal{T}(\mathcal{A}))$ can only ever happen if there exists a *relevant* position $j \in [\ell]$ such that all of the following conditions are met.

- $q_j(\mathcal{A}) = 0$, since otherwise w_j would be set to 0 in both Tr' and Tr''. Recall that q_j counts successful decryptions when queried with invalid ciphertexts for index j.
- $p'_j(\mathcal{A})/Q \geq 1/\sqrt{\lambda}$, since otherwise $w'_j(\mathcal{A})$ would be set to '?', which implies that position j is irrelevant for the infeasibility of $w'(\mathcal{A})$. Recall that p'_j counts successful decryptions when queried with valid ciphertexts for index j.
- $p''_j(\mathcal{A})/Q < 1/\sqrt{\lambda}$, since otherwise $w''_j(\mathcal{A})$ would be set to 1, which implies that $w''(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})$. Recall that p''_j counts successful decryptions when queried with invalid ciphertexts for index j.

Note that the first case implies the third case, except with negligible probability. I.e., $q_j(\mathcal{A}) = 0$ but $p''_j(\mathcal{A})/Q \geq 1/\sqrt{\lambda}$ can only happen with negligible probability for an appropriate choice of Q since the experiments q_j and p''_j are identical for relevant positions.

Denote with Hit the event that a relevant j exists that meets all three of the conditions above. By the above argument, it holds that

$$\Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] - \Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_\mathcal{A})] \leq \Pr[\mathsf{Hit}_\mathcal{A}].$$

With this in mind, we construct a PPT reduction $\mathcal{B}$ that internally runs $\mathcal{A}$ to win $\mathsf{Game\text{-}OSS\text{-}Left}$-$Q$ of $\Pi_{\mathsf{PD\text{-}BT}}$. The reduction samples a random index u, generates a code Γ, and embeds the public key pk it receives from its challenger. It simulates the corruption, partial decryption, and empty-trace oracles to $\mathcal{A}$ similarly to the reduction in the previous proof, except that when simulating the $\mathcal{O}^{\mathsf{corrupt}}(i)$ oracle, $\mathcal{B}$ now aborts and outputs a random bit β' to its challenger, whenever $\mathcal{A}$ queries a party that has a right key at position u ($\mathsf{cw}^{(i)}_u = 1$). When $\mathcal{A}$ outputs a decoder D alongside a set $\mathcal{I}_\mathsf{pd} \subseteq [n]$ with $|\mathcal{I}_\mathsf{pd}| < t$, $\mathcal{B}$ forwards $\mathcal{I}_\mathsf{pd}$ to its challenger and receives a set of Q challenges $\{(k^{(0)}_v, c^*_v, \{d_{i,v}\}_{i\in\mathcal{I}_\mathsf{pd}}\}_{v\in[Q]}$. $\mathcal{B}$ is tasked to guess the internal bit β of $\mathsf{Game\text{-}OSS\text{-}Left}$-$Q$, i.e., whether the challenge ciphertexts are either all valid ($\beta = 0$) or all invalid ($\beta = 1$). Observe that if $\beta = 0$, the challege is distributed identical to the experiment $p'_j(\mathcal{A})$ (i.e., valid ciphertexts), and if $\beta = 1$ the challenge is distributed identical to the experiments $q_j(\mathcal{A})$ and $p''_j(\mathcal{A})$ (i.e., invalid ciphertexts).

First, $\mathcal{B}$ computes $q_j(\mathcal{A})$ as well as $p'_j(\mathcal{A})$, by running the respective experiments on D. If either $q_j(\mathcal{A}) \neq 0$ or $p'_j(\mathcal{A})/Q < 1/\sqrt{\lambda}$, the reduction aborts and outputs a random bit β'. As a result, if $\mathcal{B}$ does not abort, then $\mathsf{Hit}_\mathcal{A}$ happens.

In this case, $\mathcal{B}$ queries the decoder D on its Q challenge tuples and counts for how many $v \in [Q]$ it holds that

$$D\left(c^*_v, \{d_{i,v}\}_{i\in\mathcal{I}_\mathsf{pd}}\right) = k^{(0)}_v.$$

Let $p^*(\mathcal{A})$ denote that count.

Finally, $\mathcal{B}$ returns $\beta' = 1$, if $p^*(\mathcal{A})/Q < 1/2\sqrt{\lambda}$, i.e., if the challenge is closer to the experiment result $q_j(\mathcal{A})$ with *invalid* ciphertexts, and $\beta' = 0$ otherwise, i.e. if $p^*(\mathcal{A})/Q \geq 1/2\sqrt{\lambda}$, which indicates that the challenge is closer to the experiment result $p'_j(\mathcal{A})$ with *valid* ciphertexts.

Claim 9.17 If $\beta = 1$, it can only happen with negligible probability that $q_j(\mathcal{A}) = 0$, and $p^*(\mathcal{A})/Q \geq 1/2\sqrt{\lambda}$.

Proof of Claim 9.17. Suppose that $\beta = 1$. In this case, q_j and p^* are identically distributed experiments. For $v \in [Q]$, denote with the random variables X_v and Y_v the results in the v'th iteration of the experiment p^* and q_j, respectively (i.e., $X_v = 1$, if the decoder decrypts successfully in the v'th iteration of p^* and $X_v = 0$ otherwise). Also, let $Z_v = Y_v - X_v$. Observe that $Z_v \in \{-1, 0, 1\}$, all X_v and Z_v are i.i.d, and $Z = \sum_{v=1}^{Q} Z_v = \sum_{v=1}^{Q}(Y_v - X_v) = q_j - p^*$, and thus $\mathbb{E}[Z] = 0$. We are interested in the case $q_j = 0$ and $p^* \geq Q/2\sqrt{\lambda}$, which implies that $|Z| \geq Q/2\sqrt{\lambda}$. We can use Hoeffding's inequality to bound Z as the sum of bounded, independent random variables Z_v:

$$\begin{aligned}
&\Pr\left[q_j(\mathcal{A}) = 0 \ \wedge \ p^*(\mathcal{A}) \geq \frac{Q}{2\sqrt{\lambda}} \mid \beta = 1\right] \\
&\leq \Pr\left[Z \geq \frac{Q}{2\sqrt{\lambda}}\right] = \Pr\left[\sum_{v=1}^{Q} Z_v \geq \frac{Q}{2\sqrt{\lambda}}\right] \\
&\leq \exp\left(\frac{-Q^2}{8\lambda \cdot Q}\right) \\
&\overset{Q \geq 8\lambda^2}{\leq} \exp(-\lambda)
\end{aligned}$$

■

A similar argument shows that, if $\beta = 0$, it can only happen with negligible probability that $p^*(\mathcal{A}) < Q/2\sqrt{\lambda}$ and $p'_j(\mathcal{A}) \geq Q/\sqrt{\lambda}$. Putting it all together, we get that if $\mathsf{Hit}_{\mathcal{A}}$ happens, $\mathcal{B}$'s guess u is a *relevant* index that meets the three conditions with probability at least $1/\ell$. If $\mathsf{Hit}_{\mathcal{A}}$ does not happen, or $\mathcal{B}$'s guess is not relevant, $\mathcal{B}$ outputs a random bit and guesses β with probability 1/2. Finally, if $\mathsf{Hit}_{\mathcal{A}}$ happens and $\mathcal{B}$'s guess is relevant, $\mathcal{B}$ guesses β except with probability bound by the claims above. Hence, we get that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}\text{-}Q}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) &= \Pr[\beta' = \beta] - \frac{1}{2} \\
&\geq (1 - \mathsf{negl}(\lambda)) \cdot \frac{\Pr[\mathsf{Hit}_{\mathcal{A}}]}{\ell} + \frac{1}{2} \cdot \left(1 - \frac{\Pr[\mathsf{Hit}_{\mathcal{A}}]}{\ell}\right) - \frac{1}{2} \\
&\geq \frac{(1 - \mathsf{negl}(\lambda))}{2\ell} \cdot \Pr[\mathsf{Hit}_{\mathcal{A}}] \\
&\geq \frac{1}{2\ell} \cdot (\Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] - \Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})]) - \mathsf{negl}'(\lambda),
\end{aligned}$$

for appropriate negligible functions $\mathsf{negl}(\lambda)$ and negl'. Together with

$$\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}\text{-}Q}_{\mathcal{A},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) \leq Q \cdot \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{A},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda),$$

we conclude that

$$\Pr[w'(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] - \Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] \leq \ell \cdot Q \cdot \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) + \mathsf{negl}(\lambda),$$

for a negligible function negl. ■

Taking into account our modifications to Tr, we next prove that $w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})$ can only happen with negligible probability.

Claim 9.18 For all adversaries $\mathcal{A}$ it holds that

$$\Pr[w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] \leq \ell \cdot \exp(-\lambda).$$

Proof of Claim 9.18. By definition, $w''(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})$ happens exactly if there exists a position $j \in [\ell]$ where (a) all traitors hold left keys and $w''_j(\mathcal{A}) = 1$, or (b) all traitors hold right keys and $w''_j(\mathcal{A}) = 0$. Case (b) can clearly never happen since we always return $q_j = 0$ in Tr'', for positions where all traitors hold right keys. Case (a), on the other hand, can only happen if $q_j = 0$ and $p''_j \geq Q/\sqrt{\lambda}$. Note that by the nature of our modifications q_j and p''_j are identical experiments in case (a), hence we can make a very similar probabilistic argument as in the proof of Claim 9.17 to show that, for any given j, if all adversaries hold left keys

$$\Pr\left[q_j = 0 \ \wedge \ p''_j \geq Q/\sqrt{\lambda}\right] \leq \exp\left(-\frac{Q}{2\lambda}\right) \overset{Q \geq 2\lambda^2}{\leq} \exp(-\lambda).$$

A union bound over all ℓ positions completes the proof. ∎

Applying Claims 9.15, 9.16, and 9.18 and simplifying negligibilities, we get that

$$\Pr[w(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] \leq \ell \cdot Q \cdot \left(\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Right}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) + \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda)\right) + \mathsf{negl}(\lambda).$$

To conclude the proof, we apply full collusion resilience of the fingerprinting code. Full collusion resilience tells us that whenever tracing is run on a word that is feasible and at most δ-noisy for a set S, the resulting set of traitors will be non-empty and a subset of S except with probability $2^{-\lambda}$. Note that, importantly, the reduction to the fingerprinting code needs to simulate the partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ on behalf of non-traitors, for which it does not know the codeword (i.e., it does not know whether that party is supposed to have a left or a right key in a given position j, and hence it does not know what key to use). Due to the equality of partial decryptions property of the underlying PD-BT-KEM (Definition 9.3), however, the reduction can just issue a partial decryption using any key (i.e., honest partial decryptions do not leak the side of the key that was used). The $\mathcal{O}^{\mathsf{Trace\text{-}Empty}}(\widetilde{D}, \widetilde{\mathcal{I}_{\mathsf{pd}}})$ oracle can be simulated simply by computing the word $\widetilde{w}$ for decoder $\widetilde{D}$ and partial insiders $\widetilde{\mathcal{I}_{\mathsf{pd}}}$, and outputting 1 if $\widetilde{w}$ is not δ-noisy.

Putting it all together, we can bound the advantage of any PPT adversary $\mathcal{A}$ as follows

$$\begin{aligned}\Pr[\mathsf{BadTr}_{\mathcal{A}}] &\leq \Pr[w(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] + 2^{-\lambda} \\ &\leq \ell \cdot Q \cdot \left(\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Right}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) + \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda)\right) + \mathsf{negl}(\lambda),\end{aligned}$$

which is negligible by assumption, as well as

$$\begin{aligned}&\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \\ &\geq \Pr[\mathsf{GoodDecSim}_{\mathcal{A}}] - \Pr[\neg\delta\text{-}\mathsf{noisy}(w(\mathcal{A}))|\mathsf{GoodDecSim}_{\mathcal{A}}] - \Pr[w(\mathcal{A}) \notin F(\mathcal{T}_{\mathcal{A}})] - 2^{\lambda} \\ &\geq \Pr[\mathsf{GoodDDec}_{\mathcal{A}}] - \ell \cdot \exp(-\lambda) - \ell \cdot Q \cdot \left(\mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Right}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda) + \mathsf{Adv}^{\mathsf{Game\text{-}OSS\text{-}Left}}_{\mathcal{B},\Pi_{\mathsf{PD\text{-}BT}}}(1^\lambda)\right) \\ &\quad - \mathsf{negl}(\lambda) - 2^{-\lambda} \\ &\geq \Pr[\mathsf{GoodDDec}_{\mathcal{A}}] - \mathsf{negl}'(\lambda),\end{aligned}$$

which holds by assumption for an appropriate negligible function negl'. ∎

10 Generic Threshold Traitor Tracing and Gap Elimination Compiler

In this section, we formalize our generic compiler to thresholdize traitor tracing schemes. Moreover, we show that the compiler can be used to eliminate the gap between the security threshold t and

the reconstruction threshold r of any ramp traceable threshold KEM, without compromising traceability guarantees. This compiler is useful for the schemes presented in Sections 6, 7, and 8, which all introduce a gap due to the assignment of parties into bins and the requirement that all bins must be decrypted individually.

To eliminate the gap, the compiler takes as input (1) a ramp $\mathsf{TT\text{-}KEM}$ that is τ-traceable against (σ, ρ)-insiders for reconstruction thresholds $r \in \mathcal{S}_n$ with security threshold $t(r) < r$, and (2) any standard (non-traceable) threshold KEM TE without ramp. It constructs a scheme that offers the same traceability guarantees as Π, and has security threshold $t'(r) = r$. The compiler works by setting up both schemes for the desired reconstruction threshold r, but sets up TE with $t = r$. To encrypt, one performs encryption in *both* schemes $(k^{\Pi}, c^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^{\Pi})$ and $(k^{\mathsf{TE}}, c^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}})$, and outputs $\Big(k := k^{\Pi} \oplus k^{\mathsf{TE}}, c := (c^{\Pi}, c^{\mathsf{TE}})\Big)$.

Construction 9 (Gap Elimination Compiler)**.** Let Π be a $\mathsf{TT\text{-}KEM}$ that supports reconstruction thresholds in $\mathcal{S}_n \subseteq [n]$ and security thresholds $t(r)$. Let TE be a *perfect* threshold KEM (Definition 3.11). Let the keyspaces $\Pi.\mathcal{K} = \mathsf{TE}.\mathcal{K}$ of both schemes be equal. Then, the $\mathsf{TT\text{-}KEM}$ Π' for $\mathcal{S}_n$ and security threshold $t'(r) = r$ is constructed as follows:

$\Pi'.\mathsf{KGen}(1^{\lambda}, n, r, t', \varepsilon)$**:** Generate keys in both schemes as follows

$$\Big(\mathsf{pk}^{\Pi}, \{\mathsf{sk}_i^{\Pi}\}_{i\in[n]}, \mathsf{tk}\Big) \xleftarrow{\$} \Pi.\mathsf{KGen}(1^{\lambda}, n, r, t(r), \varepsilon)$$

$$\Big(\mathsf{pk}^{\mathsf{TE}}, \{\mathsf{sk}_i^{\mathsf{TE}}\}_{i\in[n]}\Big) \xleftarrow{\$} \mathsf{TE.KGen}(1^{\lambda}, n, r, t').$$

Set $\mathsf{pk} \leftarrow \Big(\mathsf{pk}^{\Pi}, \mathsf{pk}^{\mathsf{TE}}\Big)$ and $\mathsf{sk}_i \leftarrow (\mathsf{sk}_i^{\Pi}, \mathsf{sk}_i^{\mathsf{TE}})$ for all $i \in [n]$. Output $\Big(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}\Big)$.

$\Pi'.\mathsf{Enc}(\mathsf{pk} := (\mathsf{pk}^{\Pi}, \mathsf{pk}^{\mathsf{TE}})$**:** Generate two ciphertexts

$$(k^{\Pi}, c^{\Pi}) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^{\Pi}) \text{ and } (k^{\mathsf{TE}}, c^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}}).$$

Set $k \leftarrow k^{\Pi} \oplus k^{\mathsf{TE}}$ and $c \leftarrow (c^{\Pi}, c^{\mathsf{TE}})$. Output (k, c).

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i := (\mathsf{sk}_i^{\Pi}, \mathsf{sk}_i^{\mathsf{TE}}), c := (c^{\Pi}, c^{\mathsf{TE}}))$**:** Compute

$$d_i^{\Pi} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{\Pi}, c^{\Pi}) \text{ and } d_i^{\mathsf{TE}} \leftarrow \mathsf{TE.PDec}(\mathsf{sk}_i^{\mathsf{TE}}, c^{\mathsf{TE}}).$$

Output $d_i \leftarrow (d_i^{\Pi}, d_i^{\mathsf{TE}})$.

$\Pi'.\mathsf{Combine}(S, d_i := \{(d_i^{\Pi}, d_i^{\mathsf{TE}})\}_{i\in S}, c := (c^{\Pi}, c^{\mathsf{TE}})$**:** If $|S| < r$, abort. Else, compute

$$k^{\Pi} \leftarrow \Pi.\mathsf{Combine}(S, \{d_i^{\Pi}\}_{i\in S}, c^{\Pi}) \text{ and } k^{\mathsf{TE}} \leftarrow \mathsf{TE.Combine}(S, \{d_i^{\mathsf{TE}}\}_{i\in S}, c^{\mathsf{TE}}).$$

Output $k \leftarrow k^{\Pi} \oplus k^{\mathsf{TE}}$.

$\Pi'.\mathsf{Trace}^{D'(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$**:** First, compute a decoder D making use of D' as follows. On input $D\Big(k^{\Pi}, c^{\Pi}, \{\mathsf{sk}_i^{\Pi}\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_i^{\Pi}\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)$

- Generate $(k^{\mathsf{TE}}, c^{\mathsf{TE}}) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}})$ and compute $d_i^{\mathsf{TE}} \leftarrow \mathsf{TE.PDec}(\mathsf{sk}_i^{\mathsf{TE}}, c^{\mathsf{TE}})$ for all $i \in \mathcal{I}_{\mathsf{pd}}$.
- Output $D'\Big(k^{\Pi} \oplus k^{\mathsf{TE}}, (c^{\Pi}, c^{\mathsf{TE}}), \{(\mathsf{sk}_i^{\Pi}, \mathsf{sk}_i^{\mathsf{TE}})\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{(d_i^{\Pi}, d_i^{\mathsf{TE}})\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)$.

Then, output $\mathcal{T}' \xleftarrow{\$} \Pi.\mathsf{Trace}^{D}(\mathsf{pk}^{\Pi}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i^{\Pi}\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$.

Note that the public key, secret key, partial decryption, and ciphertext size are the sum of those of Π and TE, respectively.

Theorem 10.1 *Let* Π *be a secure* TT-KEM *that is* τ*-traceable against* (σ, ρ)*-insiders,* c*-robust, and supports reconstruction thresholds* $\mathcal{S}_n \subseteq [n]$ *with security threshold* $t(r) \leq r$*. Let* TE *be a perfect* T-KEM *(Definition 3.11). Then, the* TT-KEM Π' *(Construction 9) is also semantically secure,* τ*-traceable against* (σ, ρ)*-insiders, and* c*-robust. It supports all reconstruction thresholds* $\mathcal{S}_n$ *with security threshold* $t'(r) = r$*.*

Observe that a *traceable broadcast encryption scheme* can essentially be seen as a TT-KEM with $\mathcal{S}_n = [n]$, $t = 1$, and 1-traceability against $(0, 0)$-insiders. Hence, Theorem 10.1 also formalizes the result of the generic compiler introduced in Section 2.1 from traceable broadcast encryption to traceable threshold encryption, and we get the following Corollary.

Corollary 10.2 *Let* Π *be a perfect traceable broadcast encryption scheme and* TE *be a perfect* T-KEM*. Then* Π' *(Construction 9) is* 1*-traceable against* $(0, 0)$*-insiders, perfectly correct, perfectly robust, and semantically secure with* $\mathcal{S}_n = [n]$ *and* $t'(r) = r$*.*

10.1 A Proof of Theorem 10.1

Lemma 10.3 *If* TE *is semantically secure for* $t(r) = r$*, then* Π' *is also semantically secure for* $t(r) = r$

Proof of Lemma 10.3. Lemma 10.3 follows a straightforward reduction to the semantic security of Π. Fix any $n \in \mathbb{N}$, $r \in \mathcal{S}_n$, $0 < t \leq r$, $\varepsilon \in (0, 1/2]$, and let $\mathcal{A}$ be a PPT adversary against the semantic security of Π'. We construct a PPT reduction $\mathcal{B}$ against the semantic security of TE. When run on input $(1^\lambda, n, r, t)$, $\mathcal{B}$ first runs $\mathcal{A}(1^\lambda, n, r, t, \varepsilon)$ to receive a set of corrupted parties Cor. W.l.o.g., assume that $\mathsf{Cor} \subseteq [n]$ and $|\mathsf{Cor}| < t$. Then, $\mathcal{B}$ runs $\Pi.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$ to obtain $(\mathsf{pk}^\Pi, \mathsf{sk}^\Pi, \mathsf{tk})$. It forwards Cor to its challenger to obtain $(\mathsf{pk}^{\mathsf{TE}}, \{\mathsf{sk}_i^{\mathsf{TE}}\}_{i \in \mathsf{Cor}})$ as well as the challenge pair $(k^{\mathsf{TE}}, c^{\mathsf{TE}})$. $\mathcal{B}$ runs $(k^\Pi, c^\Pi) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^\Pi)$ and assembles $k^* \leftarrow k^\Pi \oplus k^{\mathsf{TE}}$ and $c^* \leftarrow (c^\Pi, c^{\mathsf{TE}})$ and obtains

$$b' \xleftarrow{\$} \mathcal{A}\left((\mathsf{pk}^\Pi, \mathsf{pk}^{\mathsf{TE}}), \mathsf{tk}, \{(\mathsf{sk}_i^\Pi, \mathsf{sk}_i^{\mathsf{TE}})\}_{i \in \mathsf{Cor}}, k^*, c^*\right).$$

Finally, $\mathcal{B}$ outputs the same bit b' to its challenger.

Analysis. Denote with β the internal bit of $\mathcal{B}$'s challenger. Now observe that, if $\beta = 0$, then $\mathcal{B}$ simulates $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}(1^\lambda, n, r, t, \varepsilon)$ to $\mathcal{A}$ for $b = 0$. If, however, $\beta = 1$, then k^{TE} is uniformly random, and thus $k = k^\Pi \oplus k^{\mathsf{TE}}$ is also uniformly random, hence $\mathcal{B}$ simulates $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}(1^\lambda, n, r, t, \varepsilon)$ to $\mathcal{A}$ for $b = 1$. Thus, $\mathcal{B}$ simulates $\mathsf{Game\text{-}IND\text{-}CPA}$ perfectly to $\mathcal{A}$ and since it returns the same bit as $\mathcal{A}$, $\mathcal{B}$ wins whenever $\mathcal{A}$ wins. We get that

$$\Pr\left[\mathsf{Game\text{-}IND\text{-}CPA}_{\mathcal{A}}^{\Pi'}(1^\lambda, n, f, t, \varepsilon) = 1\right] = \Pr\left[\mathsf{Game\text{-}IND\text{-}CPA}_{\mathcal{B}}^{\mathsf{TE}}(1^\lambda, n, r, t) = 1\right].$$

■

Lemma 10.4 *If* Π *is* c*-robust and* TE *is perfectly robust, then* Π' *is also* c*-robust.*

Proof of Lemma 10.4. A straightforward reduction shows that for all PPT adversaries $\mathcal{A}$ there exists a PPT reduction $\mathcal{B}$ such that

$$\mathsf{Adv}_{\mathcal{A},\Pi'}^{\text{c-rob.}}(1^\lambda, n, r) \leq \mathsf{Adv}_{\mathcal{B},\Pi}^{\text{c-rob.}}(1^\lambda, n, r) + \mathsf{Adv}_{\mathcal{B},\mathsf{TE}}^{(n-r)\text{-rob.}}(1^\lambda, n, r).$$

■

Lemma 10.5 *If* Π *is* τ*-traceable against* (σ, ρ)*-insiders, then so is* Π'*. If* Π *is limitless, then so is* Π'*.*

Proof of Lemma 10.5. Let Π be τ-traceable against (σ,ρ)-insiders. Let $\mathcal{A}$ be a PPT adversary against $\mathsf{Game\text{-}Tr}$ of Π' and ε be an inverse polynomial. Fix appropriate n, r, and t'.

We first show that, given a decoder D', $\Pi.\mathsf{Trace}$ produces a decoder D with the same decryption advantage.

Claim 10.6 For all $\left(\mathsf{pk}^{\Pi},\{\mathsf{sk}_i^{\Pi}\}_{i\in[n]},\mathsf{tk}\right)\in\Pi.\mathsf{KGen}(1^\lambda,n,r,t(r),\varepsilon)$ as well as $\left(\mathsf{pk}^{\mathsf{TE}},\{\mathsf{sk}_i^{\mathsf{TE}}\}_{i\in[n]}\right)\in\mathsf{TE.KGen}(1^\lambda,n,r,t')$, $\mathsf{pk}:=(\mathsf{pk}^{\Pi},\mathsf{pk}^{\mathsf{TE}})$, and $\mathsf{sk}_i:=(\mathsf{sk}_i^{\Pi},\mathsf{sk}_i^{\mathsf{TE}})$ for all $i\in[n]$, all $\mathcal{I}_{\mathsf{sk}}\subseteq[n]$ and $\mathcal{I}_{\mathsf{pd}}\subseteq[n]$ with $|\mathcal{I}_{\mathsf{sk}}|\leq\sigma$, $|\mathcal{I}_{\mathsf{pd}}|\leq\rho$, and $\mathcal{I}_{\mathsf{sk}}\cap\mathcal{I}_{\mathsf{pd}}=\emptyset$, and all decoders D', let D be the decoder produced and subsequently traced by $\Pi'.\mathsf{Trace}^{D}(\mathsf{pk},\mathsf{tk},\mathcal{I}_{\mathsf{sk}},\mathcal{I}_{\mathsf{pd}},\{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$. Then, it holds that

$$P_{\Pi'}(D')=P_{\Pi}(D).$$

Proof of Claim 10.6. Fix all keys and parameters as indicated in the claim. It holds that

$$P_{\Pi}(D)=\Pr\left[D\Big(c^{\Pi},k_b^{\Pi},\{\mathsf{sk}_i^{\Pi}\}_{i\in\mathcal{I}_{\mathsf{sk}}},\{d_i^{\Pi}\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)=b\colon\begin{array}{c}(c^{\Pi},k_0^{\Pi})\xleftarrow{\$}\Pi.\mathsf{Enc}(\mathsf{pk}^{\Pi})\\ k_1^{\Pi}\xleftarrow{\$}\mathcal{K},b\xleftarrow{\$}\{0,1\}\\ \{d_i^{\Pi}\leftarrow\Pi.\mathsf{PDec}(sk_i^{\Pi},c^{\Pi})\}_{i\in\mathcal{I}_{\mathsf{pd}}}\end{array}\right]$$

$$=\Pr\left[D'\Big((c^{\Pi},c^{\mathsf{TE}}),k_b^{\Pi}\oplus k^{\mathsf{TE}},\{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}},\{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)=b\colon\begin{array}{c}(c^{\Pi},k_0^{\Pi})\xleftarrow{\$}\Pi.\mathsf{Enc}(\mathsf{pk}^{\Pi})\\ (c^{\mathsf{TE}},k^{\mathsf{TE}})\xleftarrow{\$}\mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}})\\ k_1^{\Pi}\xleftarrow{\$}\mathcal{K},b\xleftarrow{\$}\{0,1\}\\ \{d_i\leftarrow\Pi'.\mathsf{PDec}(\mathsf{sk}_i,(c^{\Pi},c^{\mathsf{TE}})\}_{i\in\mathcal{I}_{\mathsf{pd}}}\end{array}\right]$$

$$=\Pr\left[D'\Big((c^{\Pi},c^{\mathsf{TE}}),k_b,\{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}},\{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)=b\colon\begin{array}{c}(c^{\Pi},k_0^{\Pi})\xleftarrow{\$}\Pi.\mathsf{Enc}(\mathsf{pk}^{\Pi})\\ (c^{\mathsf{TE}},k^{\mathsf{TE}})\xleftarrow{\$}\mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}})\\ k_0\leftarrow k_0^{\Pi}\oplus k^{\mathsf{TE}}\\ k_1\xleftarrow{\$}\mathcal{K},b\xleftarrow{\$}\{0,1\}\\ \{d_i\leftarrow\Pi'.\mathsf{PDec}(\mathsf{sk}_i,(c^{\Pi},c^{\mathsf{TE}})\}_{i\in\mathcal{I}_{\mathsf{pd}}}\end{array}\right]$$

$$=\Pr\left[D'\Big(c,k_b,\{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}},\{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big)=b\colon\begin{array}{c}(c,k_0)\xleftarrow{\$}\Pi'.\mathsf{Enc}(\mathsf{pk})\\ k_1\xleftarrow{\$}\mathcal{K},b\xleftarrow{\$}\{0,1\}\\ \forall i\in\mathcal{I}_{\mathsf{pd}}\colon d_i\leftarrow\Pi'.\mathsf{PDec}(sk_i,c)\end{array}\right]=P_{\Pi'}(D').$$

In the first step, we replaced D by its call to D'. In the second step, we use the perfect secrecy of the one-time pad, i.e., if k_1^{Π} is uniformly random, then so is $k_1^{\Pi}\oplus k^{\mathsf{TE}}$. The result is then identical to the experiment for $P'_{\Pi}(D')$. ∎

Equipped with Claim 10.6, we first show that $\Pr[\mathsf{BadTr}_{\mathcal{A}}]\leq\mathsf{negl}(\lambda)$. Consider a PPT reduction $\mathcal{B}$ running in $\mathsf{Game\text{-}Tr}^{\Pi'}_{\mathcal{B},\varepsilon,\sigma,\rho}(1^\lambda,n,r,t(r))$. $\mathcal{B}$ simulates $\mathsf{Game\text{-}Tr}^{\Pi}_{\mathcal{A},\varepsilon,\sigma,\rho}(1^\lambda,n,r,t')$ to $\mathcal{A}$ as follows. Denote with $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$ the set of real traitors corrupted by $\mathcal{A}$ and the reduction $\mathcal{B}$, respectively. Denote with $\mathcal{T}'_{\mathcal{A}}$ and $\mathcal{T}'_{\mathcal{B}}$, the set of traitors found by the trace algorithm in the $\mathsf{Game\text{-}Tr}$ experiments when running $\mathcal{A}$ against Π' and $\mathcal{B}$ against Π, respectively.

When $\mathcal{B}$ receives the security parameter λ and the challenge public key pk^{Π}, it runs $\mathsf{TE.KGen}(1^\lambda,n,r,t')$ to generate the instance of TE. It embeds the challenge public key by setting $\mathsf{pk}:=(\mathsf{pk}^{\Pi},\mathsf{pk}^{\mathsf{TE}})$ and runs $\mathcal{A}(1^\lambda,\mathsf{pk})$, simulating the $\mathcal{O}^{\mathsf{corrupt}}$ and $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracles, by forwarding queries to its challenger and embedding the TE parts.

Simulating $\mathcal{O}^{\mathsf{corrupt}}(i)$. On input i, $\mathcal{B}$ queries $\mathcal{O}^{\mathsf{corrupt}}(i)$ from its challenger to obtain sk_i^{Π} and forwards $(\mathsf{sk}_i^{\Pi},\mathsf{sk}_i^{\mathsf{TE}})$ to $\mathcal{A}$. As a result, $\mathcal{B}$ corrupts the same parties as $\mathcal{A}$, i.e., $\mathcal{T}_{\mathcal{A}}=\mathcal{T}_{\mathcal{B}}$.

Simulating $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i,r_{\mathsf{Enc}})$. $\mathcal{B}$ parses r_{Enc} as $(r^{\Pi}_{\mathsf{Enc}},r^{\mathsf{TE}}_{\mathsf{Enc}})$ and obtains d_i^{Π} by querying the oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i,r^{\Pi}_{\mathsf{Enc}})$ from its own challenger. It computes $(k^{\mathsf{TE}},c^{\mathsf{TE}})\leftarrow\mathsf{TE.Enc}(\mathsf{pk}^{\mathsf{TE}};r^{\mathsf{TE}}_{\mathsf{Enc}})$ and $d_i^{\mathsf{TE}}\leftarrow\mathsf{TE.PDec}(\mathsf{sk}_i^{\mathsf{TE}},c^{\mathsf{TE}})$. $\mathcal{B}$ returns $d_i=(d_i^{\Pi},d_i^{\mathsf{TE}})$ to $\mathcal{A}$.

When $\mathcal{A}$ outputs $(D', \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}})$, $\mathcal{B}$ constructs D as indicated by $\Pi'.\mathsf{Trace}$, which is possible, because $\mathcal{B}$ knows $\mathsf{sk}_i^{\mathsf{TE}}$ for all $i \in [n]$. $\mathcal{B}$ outputs $(D, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}})$ to its challenger.

Analysis. We start with BadTr: By construction it holds that $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}}$, and since $\mathcal{B}$ corrupts the same parties as $\mathcal{A}$ through its simulation of $\mathcal{O}^{\mathsf{corrupt}}$, it also holds that $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$. Hence, we get that

$$\Pr[\mathsf{BadTr}_{\mathcal{A}}] = \Pr[\mathcal{T}'_{\mathcal{A}} \not\subseteq \mathcal{T}_{\mathcal{A}}] = \Pr[\mathcal{T}'_{\mathcal{B}} \not\subseteq \mathcal{T}_{\mathcal{B}}] = \Pr[\mathsf{BadTr}_{\mathcal{B}}],$$

which is assumed to be negligible.

To bound $\Pr[\mathsf{GoodTr}_{\mathcal{A}}]$, consider Claim 10.6, which implies that $\Pr[\mathsf{GoodDec}_{\mathcal{A}}] = \Pr[\mathsf{GoodDec}_{\mathcal{B}}]$. Hence, we get that

$$\begin{aligned}
\Pr[\mathsf{GoodTr}_{\mathcal{A}}] &= \Pr[\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}} \wedge |\mathcal{T}'_{\mathcal{A}}| \geq \tau] \\
&= \Pr[\mathcal{T}'_{\mathcal{B}} \subseteq \mathcal{T}_{\mathcal{B}} \wedge |\mathcal{T}'_{\mathcal{B}}| \geq \tau] \\
&= \Pr[\mathsf{GoodTr}_{\mathcal{B}}] \\
&\geq \Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \mathsf{negl}(\lambda) = \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda),
\end{aligned}$$

for a negligible function $\mathsf{negl}(\lambda)$ by assumption. ■

References

[1] Agarwal, A., Fernando, R., Pinkas, B.: Efficiently-thresholdizable batched identity based encryption, with applications. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part III. LNCS, vol. 16002, pp. 69–100. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01881-6_3 (Cited on page 4, 69.)

[2] Bebel, J., Ojha, D.: Ferveo: Threshold decryption for mempool privacy in BFT networks. Cryptology ePrint Archive, Report 2022/898 (2022), https://eprint.iacr.org/2022/898 (Cited on page 4, 69.)

[3] Bhattacharyya, R., Bormet, J., Faust, S., Mukherjee, P., Othman, H.: CCA-secure traceable threshold (ID-based) encryption and application. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 2324–2338. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3744876 (Cited on page 4, 8, 22, 102.)

[4] Billet, O., Phan, D.H.: Efficient traitor tracing from collusion secure codes. In: Safavi-Naini, R. (ed.) ICITS 08. LNCS, vol. 5155, pp. 171–182. Springer, Berlin, Heidelberg (Aug 2008). https://doi.org/10.1007/978-3-540-85093-9_17 (Cited on page 15.)

[5] Blakley, G.R., Meadows, C.: Security of ramp schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO'84. LNCS, vol. 196, pp. 242–268. Springer, Berlin, Heidelberg (Aug 1984). https://doi.org/10.1007/3-540-39568-7_20 (Cited on page 9.)

[6] Boneh, D., Boyen, X., Goh, E.J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Berlin, Heidelberg (May 2005). https://doi.org/10.1007/11426639_26 (Cited on page 98.)

[7] Boneh, D., Bünz, B., Nayak, K., Rotem, L., Shoup, V.: Context-dependent threshold decryption and its applications. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VI. LNCS, vol. 16250, pp. 506–538. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_16 (Cited on page 4, 53, 69, 70.)

[8] Boneh, D., Laufer, E., Tas, E.N.: Batch decryption without epochs and its application to encrypted mempools. Cryptology ePrint Archive, Report 2025/1254 (2025), https://eprint.iacr.org/2025/1254 (Cited on page 4, 69.)

[9] Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM CCS 2008. pp. 501–510. ACM Press (Oct 2008). https://doi.org/10.1145/1455770.1455834 (Cited on page 15, 16.)

[10] Boneh, D., Partap, A., Rotem, L.: Accountability for misbehavior in threshold decryption via threshold traitor tracing. Cryptology ePrint Archive, Report 2023/1724 (2023), https://eprint.iacr.org/2023/1724 (Cited on page 97.)

[11] Boneh, D., Partap, A., Rotem, L.: Accountability for misbehavior in threshold decryption via threshold traitor tracing. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VII. LNCS, vol. 14926, pp. 317–351. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68394-7_11 (Cited on page 4, 5, 6, 7, 8, 9, 15, 20, 22, 46, 47, 48, 49, 50, 51, 53, 69, 94, 102.)

[12] Boneh, D., Partap, A., Rotem, L.: Traceable secret sharing: Strong security and efficient constructions. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part V. LNCS, vol. 14924, pp. 221–256. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68388-6_9 (Cited on page 16.)

[13] Boneh, D., Partap, A., Rotem, L.: Traceable verifiable random functions. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part II. LNCS, vol. 16001, pp. 203–234. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01878-6_7 (Cited on page 16.)

[14] Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data (extended abstract). In: Coppersmith, D. (ed.) CRYPTO'95. LNCS, vol. 963, pp. 452–465. Springer, Berlin, Heidelberg (Aug 1995). https://doi.org/10.1007/3-540-44750-4_36 (Cited on page 15.)

[15] Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: Juels, A., Wright, R.N., De Capitani di Vimercati, S. (eds.) ACM CCS 2006. pp. 211–220. ACM Press (Oct / Nov 2006). https://doi.org/10.1145/1180405.1180432 (Cited on page 12.)

[16] Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 480–499. Springer, Berlin, Heidelberg (Aug 2014). https://doi.org/10.1007/978-3-662-44371-2_27 (Cited on page 16.)

[17] Bormet, J., Choudhuri, A.R., Faust, S., Garg, S., Othman, H., Policharla, G.V., Qu, Z., Wang, M.: BEAST-MEV: Batched threshold encryption with silent setup for MEV prevention. Cryptology ePrint Archive, Report 2025/1419 (2025), https://eprint.iacr.org/2025/1419 (Cited on page 4, 69.)

[18] Bormet, J., Dziembowski, S., Faust, S., Lizurej, T., Mielniczuk, M.: Strong secret sharing with snitching. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part IV. LNCS, vol. 16003, pp. 295–327. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01884-7_10 (Cited on page 16.)

[19] Bormet, J., Faust, S., Othman, H., Qu, Z.: BEAT-MEV: Epochless approach to batched threshold encryption for MEV prevention. In: Bauer, L., Pellegrino, G. (eds.) USENIX Security 2025. pp. 3457–3476. USENIX Association (Aug 2025), https://www.usenix.org/conference/usenixsecurity25/presentation/bormet (Cited on page 4, 69.)

[20] Bormet, J., Hofmann, J., Othman, H.: Traceable threshold encryption without a trusted dealer. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VI. LNCS, vol. 16250, pp. 471–505. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_15 (Cited on page 4, 8, 16, 22, 92, 101, 102.)

[21] Boyen, X.: The uber-assumption family (invited talk). In: Galbraith, S.D., Paterson, K.G. (eds.) PAIRING 2008. LNCS, vol. 5209, pp. 39–56. Springer, Berlin, Heidelberg (Sep 2008). https://doi.org/10.1007/978-3-540-85538-5_3 (Cited on page 98.)

[22] Canard, S., Papon, N., Phan, D.H.: Public traceability in threshold decryption. IACR Communications in Cryptology **2**(2) (2025). https://doi.org/10.62056/akjb0lmol (Cited on page 4, 22, 102.)

[23] Chakrabarti, A., Maitra, M., Mondal, A., Sehgal, K.: Silent threshold traitor tracing & enhancing mempool privacy. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 1769–1783. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3765099 (Cited on page 4, 8, 22, 101.)

[24] Chen, Y., Vaikuntanathan, V., Waters, B., Wee, H., Wichs, D.: Traitor-tracing from LWE made simple and attribute-based. In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 341–369. Springer, Cham (Nov 2018). https://doi.org/10.1007/978-3-030-03810-6_13 (Cited on page 16.)

[25] Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 257–270. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5_25 (Cited on page 4, 16.)

[26] Choudhuri, A.R., Garg, S., Piet, J., Policharla, G.V.: Mempool privacy via batched threshold encryption: Attacks and defenses. In: Balzarotti, D., Xu, W. (eds.) USENIX Security 2024. USENIX Association (Aug 2024), https://www.usenix.org/conference/usenixsecurity24/presentation/choudhuri (Cited on page 4, 69.)

[27] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., Juels, A.: Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges (2019), https://arxiv.org/abs/1904.05234 (Cited on page 69.)

[28] Das, S., Datta, P., Partap, A., Sasmal, S., Zhandry, M.: Optimal threshold traitor tracing. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 486–516. Springer Nature Switzerland, Cham (2026) (Cited on page 4, 8, 9, 13, 22, 39, 75, 76, 77, 101, 102.)

[29] Desmedt, Y.: Society and group oriented cryptography: A new concept. In: Pomerance, C. (ed.) CRYPTO'87. LNCS, vol. 293, pp. 120–127. Springer, Berlin, Heidelberg (Aug 1988). https://doi.org/10.1007/3-540-48184-2_8 (Cited on page 4.)

[30] Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 307–315. Springer, New York (Aug 1990). https://doi.org/10.1007/0-387-34805-0_28 (Cited on page 4.)

[31] Dubhashi, D., Priebe, V., Ranjan, D.: Negative dependence through the fkg inequality. Technical Report RS-96-27, BRICS Report Series (1996) (Cited on page 70.)

[32] Dziembowski, S., Faust, S., Lizurej, T., Mielniczuk, M.: Secret sharing with snitching. In: Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D. (eds.) ACM CCS 2024. pp. 840–853. ACM Press (Oct 2024). https://doi.org/10.1145/3658644.3690296 (Cited on page 16.)

[33] Dziembowski, S., Faust, S., Luhn, J.: Shutter network: Private transactions from threshold cryptography. Cryptology ePrint Archive, Report 2024/1981 (2024), https://eprint.iacr.org/2024/1981 (Cited on page 4, 69.)

[34] Elowsson, A.: Lucid: Encrypted mempool with distributed payload propagation. https://ethresear.ch/t/lucid-encrypted-mempool-with-distributed-payload-propagation/24042 (Feb 2026), ethereum Research forum post (Cited on page 69.)

[35] Eskandari, S., Moosavi, S., Clark, J.: Sok: Transparent dishonesty: Front-running attacks on blockchain. In: Bracciali, A., Clark, J., Pintore, F., Rønne, P.B., Sala, M. (eds.) Financial Cryptography and Data Security. pp. 170–189. Springer International Publishing, Cham (2020) (Cited on page 69.)

[36] Farràs, O., Guiot, M.: Traceable secret sharing schemes for general access structures. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 3–32. Springer Nature Switzerland, Cham (2026) (Cited on page 16.)

[37] Fernando, R., Policharla, G.V., Tonkikh, A., Xiang, Z.: TrX: Encrypted mempools in high performance BFT protocols. Cryptology ePrint Archive, Report 2025/2032 (2025), https://eprint.iacr.org/2025/2032 (Cited on page 4.)

[38] Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_2 (Cited on page 97.)

[39] Garg, S., Kolonelos, D., Policharla, G.V., Wang, M.: Threshold encryption with silent setup. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VII. LNCS, vol. 14926, pp. 352–386. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68394-7_12 (Cited on page 4, 69.)

[40] Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC. pp. 25–32. ACM Press (May 1989). https://doi.org/10.1145/73007.73010 (Cited on page 75.)

[41] Goyal, R., Koppula, V., Waters, B.: Collusion resistant traitor tracing from learning with errors. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th ACM STOC. pp. 660–670. ACM Press (Jun 2018). https://doi.org/10.1145/3188745.3188844 (Cited on page 16.)

[42] Goyal, R., Vusirikala, S., Waters, B.: Collusion resistant broadcast and trace from positional witness encryption. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 3–33. Springer, Cham (Apr 2019). https://doi.org/10.1007/978-3-030-17259-6_1 (Cited on page 16.)

[43] Goyal, V., Jain, A., Partap, A.: Traceable secret sharing revisited. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 33–63. Springer Nature Switzerland, Cham (2026) (Cited on page 5, 9, 23.)

[44] Goyal, V., Song, Y., Srinivasan, A.: Traceable secret sharing and applications. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part III. LNCS, vol. 12827, pp. 718–747. Springer, Cham, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84252-9_24 (Cited on page 16.)

[45] Hall-Andersen, M., Simkin, M., Wagner, B.: Silent threshold encryption with one-shot adaptive security. Cryptology ePrint Archive, Report 2025/1384 (2025), https://eprint.iacr.org/2025/1384 (Cited on page 4, 16, 69.)

[46] Joag-Dev, K., Proschan, F.: Negative association of random variables with applications. The Annals of Statistics **11**(1), 286–295 (1983) (Cited on page 70, 71.)

[47] Kelkar, M., Ganesh, A., Partap, A., Bonneau, J., Weinberg, S.M.: Breaking Omertà: On threshold cryptography, smart collusion, and whistleblowing. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 3505–3519. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3765087 (Cited on page 4.)

[48] Maurer, U.M.: Abstract models of computation in cryptography (invited paper). In: Smart, N.P. (ed.) 10th IMA International Conference on Cryptography and Coding. LNCS, vol. 3796, pp. 1–12. Springer, Berlin, Heidelberg (Dec 2005). https://doi.org/10.1007/11586821_1 (Cited on page 15, 97.)

[49] McDiarmid, C.: On the method of bounded differences, p. 148–188. London Mathematical Society Lecture Note Series, Cambridge University Press (1989) (Cited on page 70.)

[50] Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press (2005) (Cited on page 18.)

[51] Robbins, H.: A remark on stirling's formula. The American Mathematical Monthly **62**(1), 26–29 (1955) (Cited on page 73.)

[52] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Berlin, Heidelberg (May 1997). https://doi.org/10.1007/3-540-69053-0_18 (Cited on page 15, 97.)

[53] Waters, B., Wu, D.J.: Silent threshold cryptography from pairings: Expressive policies in the plain model. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 517–547. Springer Nature Switzerland, Cham (2026) (Cited on page 4, 69.)

[54] Zhandry, M.: New techniques for traitor tracing: Size $N^{1/3}$ and more from pairings. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 652–682. Springer, Cham (Aug 2020). https://doi.org/10.1007/978-3-030-56784-2_22 (Cited on page 12, 13, 14, 22, 31, 32, 39, 75, 76, 77.)

[55] Zhandry, M.: Optimal traitor tracing from pairings. In: Fehr, S., Fouque, P.A. (eds.) EUROCRYPT 2025, Part III. LNCS, vol. 15603, pp. 305–335. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91131-6_11 (Cited on page 16.)

[56] Zhou, L., Qin, K., Cully, A., Livshits, B., Gervais, A.: On the just-in-time discovery of profit-generating transactions in DeFi protocols. In: 2021 IEEE Symposium on Security and Privacy. pp. 919–936. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00113 (Cited on page 69.)

[57] Zhou, L., Qin, K., Ferreira Torres, C., Le, D.V., Gervais, A.: High-frequency trading on decentralized on-chain exchanges. In: 2021 IEEE Symposium on Security and Privacy. pp. 428–445. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00027 (Cited on page 69.)

Appendices

A Relevance of Insider Resilience in the Blockchain Setting

We argue that both full and partial insider attacks are highly relevant in blockchain systems, particularly in the context of encrypted mempools for MEV prevention.

Miner/Maximal Extractable Value (MEV) [27] refers to the amount of profit that block proposers can extract by inserting, reordering, or omitting transactions from the mempool when publishing a block. MEV is vastly exploited in practice, and encompasses attacks such as frontrunning and variants thereof [35, 56, 57].

A prominent approach to mitigating MEV is the use of *encrypted mempools* [2, 33]. Encrypted mempools fundamentally prevent MEV attacks by keeping the content of pending transactions private until the block is finalized. This is typically realized using *threshold encryption*: Users encrypt their transactions to the public key of a committee, publish the ciphertext to the mempool, and once a block of encrypted transactions is finalized, the decryption committee jointly decrypts all transactions in the block. Encrypted mempools have recently received significant attention in both the blockchain and cryptographic communities. In Ethereum, for instance, proposals such as EIP-8105[17] and LUCID [34] explore encrypted mempool designs. At the same time, numerous advances in threshold encryption [39, 53, 45, 26, 19, 17, 1, 8, 7] have introduced concepts such as silent setup, batching, or context dependency that are largely motivated by this application. We next outline how collusion can arise in encrypted mempools and why addressing both full and partial insiders is essential.

Full Insiders: MEV-Driven Committee Members. Threshold traitor tracing was originally motivated in the encrypted mempool setting [11]. In this setting, an *external party* may pay at least t committee members to collude and construct a decryption box. This box allows the buyer to decrypt pending transactions in the mempool and extract MEV from them.

The attack is mitigated by allowing the buyer to forward the decoder to a *tracer*, who can query it to identify at least one colluding committee member, who is then penalized. Collusion is therefore discouraged by the risk of detection and punishment.

However, the existing model does not consider that a buyer might not be *external*, but itself be a committee member seeking to extract MEV. As already discussed above, a decoder that is sold to an insider and asks for the insider's secret key as additional input may only be traced back to the insider himself, breaking down the incentive structure that initially deterred collusion. To see this, consider the perspective of the insider: Once sold a decoder, the insider could (a) use it to extract MEV from future blocks, or (b) forward it to the tracer. If tracing, however, does not identify anyone but the insider, then the tracer can punish no one. Colluders stay unpunished, and the insider does not get a reward, hence a rational insider will choose option (a), and collusion remains risk-free.

Partial Insiders: Concurrent Block Attestation and Decryption Leaks Partial Decryptions. Since proof-of-stake-based blockchain systems already use a committee of validators to vote on which block to include in the chain, a natural approach for encrypted mempools is to reuse the same committee for decrypting finalized transactions. Ideally, when voting for a candidate block, committee members would publish their vote[18] together with a partial decryption of the transactions in that block. The fundamental advantage of sending partial decryptions concurrently with votes is that it significantly reduces latency compared to first finalizing an encrypted block and then performing a separate decryption round.

[17] https://eips.ethereum.org/EIPS/eip-8105

[18] Typically, such a vote consists of a signature on the candidate block.

However, this naive approach introduces a problem. Consider two candidate blocks competing for the same slot. Suppose the majority of committee members vote for the first block, thereby finalizing it and decrypting its transactions. Some honest validators, however, might have voted for the second block and thus revealed partial decryptions of transactions that appear only in that block. Hence, the system leaks partial decryptions of pending transactions. As soon as t or more partial decryptions for a given transaction are leaked (possibly across multiple slots), the transaction can be decrypted and is susceptible to MEV. Recently, Boneh et al. [7] introduced *context-dependent* threshold encryption, solving the problem outlined above: Partial decryptions are tied to a specific *decryption context*, say, the block hash, and can only be combined with other partial decryptions *for the same context*. Thus, transactions remain private until they are voted for by at least t parties *as part of the same block*. Setting t to the threshold of votes required to finalize a block keeps transactions encrypted until finalization.

From the perspective of collusion resilience, however, this introduces a new challenge: If honest parties regularly issue partial decryptions on transactions that are supposed to stay encrypted, then these honest parties effectively act as *partial insiders*. Colluders could design a decoder that decrypts a given ciphertext only when provided partial decryptions from a number of honest parties as additional input. Such a decoder can then even be sent to external buyers, who observe block votes, input partial decryptions into the decoder, and thus can decrypt even blocks that received only a minority of votes. Since such blocks may contain unfinalized transactions, this again enables MEV extraction by the buyer of the decoder. If tracing such a decoder blames the honest validators providing the partial decryptions rather than the colluders who constructed the decoder, the colluders again escape punishment.

B Proofs of Partition Lemmas

B.1 Some Probabilistic Facts

We first present some probabilistic facts that will be essential to prove our probabilistic lemmas on partitions of parties.

Definition B.1 (Negative association [46]). Let $X := (X_1, \ldots, X_\ell)$ be a vector of random variables. The variables X are *negatively associated* if, for every two disjoint index sets $I, J \subseteq [\ell]$,

$$\mathbb{E}\Big[f(X_i, i \in I) \cdot g(X_j, j \in J)\Big] \leq \mathbb{E}\Big[f(X_i, i \in I)\Big] \cdot \mathbb{E}\Big[g(X_j, j \in J)\Big],$$

for all functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both non-decreasing or both non-increasing.

We are interested in this definition because the Chernoff-Hoeffding bounds are applicable to sums of variables that satisfy the negative association condition [49].

Fact B.2 ([31]) Let $X := (X_1, \ldots, X_n)$ be random variables taking values in $\{0, 1\}$ and distributed uniformly over all m-weight vectors, where $m \leq n$. That is, for every $x = (x_1, \ldots, x_n) \in \{0, 1\}^n$ of Hamming weight m it holds that

$$\Pr[X = x] = \binom{n}{m}^{-1},$$

and $\Pr[X = x] = 0$ for every x of Hamming weight $\mathrm{wt}(x) \neq m$. Then the random variables $(X_1, \ldots, X_n)$ are negatively associated.

Claim B.3 Given a fixed n-bit string x and a uniformly chosen partition Π of $[n]$ into k bins of size n/k each, denote by $Y_{i,j}$ the random variable that takes value 1 when the j-th bit in the i-th bin is 1. Then, for every index $i \in [k]$, the n/k variables $\{ Y_{i,j} \}_{1 \leq j \leq n/k}$ are negatively associated.

Proof. The $n/k \cdot k$ random variables $Y_{i,j}$ for $1 \le i \le k$ and $1 \le j \le n/k$ satisfy the conditions for Fact B.2 for $\ell = n$ and $m = wt(x)$, and they are therefore negatively associated. By [46], any subset of negatively associated variables are also negatively associated. Thus, the claim follows. ■

Claim B.4 Given a fixed n-bit string x and a uniformly chosen partition Π of $[n]$ into 2 bins B_1, B_2 such that $|B_1| = n^a, |B_2| = n - n^a$ for $a \in \{0, 1\}$, denote by $Y_{i,j}$, the random variable that takes value 1 when the j-th bit in the i-th bin is 1. Then, for every index $i \in \{1, 2\}$, the variables $\{ Y_{i,j} \}_{1 \le j \le |B_i|}$ are negatively associated.

Proof. The $n^a + n - n^a$ random variables $\{Y_{1,j}\}_{1 \le j \le n^a}, \{Y_{2,j}\}_{1 \le j \le n - n^a}$ satisfy the conditions for Fact B.2 for $\ell = n$ and $m = wt(x)$, and they are therefore negatively associated. By [46], any subset of negatively associated variables are also negatively associated. Thus, the claim follows. ■

B.2 Proof of Lemma 6.1

Lemma 6.1 (Restated). Let $n = \mathsf{poly}(\lambda)$ and $K = o(n^a)$ for $a < 1$. Further, let $P_j = \{B_i^{(j)}\}_{i \in [K]}$ for $j \in [\ell]$ be a collection of ℓ independently and uniformly sampled (K, n)-balanced partitions, where $|B_i^{(j)}| = n/K$ for every $i \in [K]$ and every $j \in [\ell]$. Let $t_{\mathsf{bin}} = \frac{n}{2K} - \sqrt{\frac{\lambda n}{K}}$. Then,

1. For any $\ell \ge 1$, for uniformly distributed $S \subseteq [n]$ where $|S| \ge n/2$, there exists some $j \in [\ell]$ such that for all $i \in [K]$,
$$|S \cap B_i^{(j)}| \ \ge \ t_{\mathsf{bin}},$$
except with negligible probability in λ.

2. Let $r = n/K - t_{\mathsf{bin}}$, $\mathbf{r} = r \cdot (1, \ldots, 1) \in \mathbb{N}^K$, and $c \in (0, 1/2)$. Then, for $\ell \ge 2(1-c)K/c^2$, the probability that there is a subset $S \subseteq [n]$, where $|S| \le n/2 - cn$, which $(\ell, \mathbf{r})$-blocks $\{P_j\}_{j \in [\ell]}$ is negligible in λ.

Proof. Let S be a set and P a partition. For each bin B_i in P, define $Y_{i,j}$ as an indicator random variable that equals 1 if and only if the j-th party in the i-th bin belongs to S. Equivalently, if we represent the i-th bin as a binary string of length $|B_i|$, the j-th bit of the string is 1 if the corresponding party is in S, and 0 otherwise. For each $i \in [K]$, the variables $\{Y_{i,j}\}_{j \in [|B_i|]}$ are negatively associated (see Claim B.3). Let us denote by $Y_i = \sum_j Y_{i,j}$ the random variable representing the number of parties of S that are in the i-th bin.

The first part of the lemma is proven using Chernoff's bound. First note that for $|S| \ge n/2$, we get that the expectation is $\mu = \mathbb{E}[Y_i] \ge n/(2K)$. Thus, by Chernoff bound, we get $\Pr[Y_i < t_{\mathsf{bin}}] \le \exp(-\delta^2 \mu/2) = \exp(-\lambda)$, where $\delta = 2\sqrt{\lambda K/n}$. By union bound over all bins, the probability that this holds for one of the bins is at most $K \exp(-\lambda)$, which is negligible in λ.[19]

We now turn to the second part. Fix a set $S \subseteq [n]$ with $|S| = n/2 - cn$, for some constant $c \in (0, 1/2)$. For a given i, it now holds that $\mu = \mathbb{E}(Y_i) \le n/(2K) - cn/K$. We next bound the probability that S r_i-blocks the i-th bin for a partition P, i.e., $|S \cap B_i| \ge |B_i| - t_{\mathsf{bin}} = \frac{n}{2K} + \sqrt{\lambda n/K}$. We use the Chernoff upper-tail bound:

$$\Pr\left[Y_i \ge \frac{n}{(2K)} + \sqrt{\lambda n/K}\right] = \Pr[Y_i \ge (1+\delta)\mu] \le \exp\left(\frac{-\delta^2 \mu}{(2+\delta)}\right).$$

We seek δ such that $(1+\delta)\mu = \frac{n}{2K} + \sqrt{\lambda n/K}$, where $\mu = \frac{n}{2K} - cn/K$. That is, $\delta = \frac{\sqrt{\lambda n/K} + cn/K}{n/(2K) - cn/K}$. Since the Chernoff upper-tail bound is decreasing in δ, we may lower bound

[19] One could decrease the gap by increasing ℓ. However, as we aim to get ℓ, which does not depend on λ, we keep the gap as is.

$\delta \geq \frac{cn/K}{n/(2K)-cn/K} = \frac{2c}{1-2c}$. Hence, using $\delta \geq \frac{2c}{1-2c}$ and $\mu = \frac{(1-2c)n}{2K}$, we get: $\frac{\delta^2\mu}{2+\delta} \geq \frac{c^2}{1-c} \cdot \frac{n}{K}$. Thus,

$$\Pr[Y_i \geq n/(2K) + \sqrt{\lambda n/K}] \leq \exp\left(\frac{-c^2 n}{(1-c)K}\right).$$

Taking a union bound over all K bins, the probability that the set S $\mathbf{r}$-blocks partition P is

$$p_s = K \exp\left(\frac{-c^2 n}{(1-c)K}\right) \leq \exp\left(\frac{-c^2 n}{(1-c)K} + \ln K\right).$$

Since the ℓ partitions are sampled independently, the probability that S is $(\ell, \mathbf{r})$-blocks the collection of ℓ partitions is at most $\exp\left(\frac{-\ell c^2 n}{(1-c)K} + \ell \ln K\right)$. Finally, taking a loose union bound over all subsets $S \subseteq [n]$ ($|\mathcal{P}([n])| \leq \exp(n)$), the probability that there exists such an S is at most $\exp\left(n + \frac{-\ell c^2 n}{(1-c)K} + \ell \ln K\right)$. Choosing $\ell = 2K(1-c)/c^2$, this expression can be bounded by

$$\exp\left(-n + \frac{K \ln K (1-c)}{c^2}\right).$$

Since $n \in \mathsf{poly}(\lambda)$, $K = o(n^a)$ for $a < 1$, and c is a constant, this probability is negligible in λ for sufficiently large n, which concludes the proof of the second part.

■

B.3 Proof of Lemma 8.2

Lemma 8.2 (Restated). Let $P_j = \{B_1^j, B_2^j\}$ for $j \in [\ell]$ be a collection of ℓ independently and uniformly sampled $(2,n)$-unbalanced partitions, where for $a \in (0,1)$, $|B_1^j| = n^a$ and $|B_2^j| = n - n^a$ for every $j \in [\ell]$. Let $\delta_1 = 6n^{-a/2}$, $\delta_2 = 6\frac{\sqrt{n}}{n-n^a}$, $t_1 = (1-\delta_1)\frac{n^a}{2}$, $t_2 = (1-\delta_2)\frac{(n-n^a)}{2}$. Then,

1. If $\ell \geq 6\lambda$, for uniformly distributed $S \subseteq [n]$ where $|S| \geq n/2$, it holds that S $(0.8\ell, (t_1, t_2))$-dominates the partitions $\{P_j\}_{j\in[\ell]}$ with overwhelmning probability.

2. Let $c \in (0, 0.5)$, $\ell \geq 10n^{1-a}(1-c)/(c^2)$, $\mathbf{r}_1 = (n^a - t_1, n - n^a - t_2)$ and $\beta = 0.2\ell$. Then, the probability that there exists a subset $S \subseteq [n]$, such that $|S| \leq n/2 - cn$ and $(\beta, \mathbf{r}_1)$-blocks $\{P_j\}_{j\in[\ell]}$ is negligible in λ.

3. Let $c \in (0, 0.5)$, $\ell \geq 20n^{1-a}(2-3c)/c^2$, $\mathbf{r}_2 = (t_1, 0)$, and $\beta = 0.2\ell$. Then, the probability that there exists a subset $S \subseteq [n]$, such that $|S| \leq n/2 - cn$ and $(0.2\ell, \mathbf{r}_2)$-dominates $\{P_j\}_{j\in[\ell]}$ is negligible in λ.

4. If $\ell \geq 3n^{1-a}\lambda$, then the probability that there exists a $u \in [n]$ such that $u \in B_1^j$ for more than $2\ell/n^{1-a}$ indices j is negligible.

Proof. Let S be a set and P a partition. For each bin B_i in P, define $Y_{i,j}$ as an indicator random variable that equals 1 if and only if the j-th party in the i-th bin belongs to S. Equivalently, if we represent the bin as a binary string of length $|B_i|$, the j-th bit of the i-th bin is 1 if the corresponding party is in S. For each $i \in \{1,2\}$, the variables $\{Y_{i,j}\}_{j\in[|B_i|]}$ are negatively associated (see Claim B.4). Let us denote by $Y_i = \sum_j Y_{i,j}$ the random variable representing the number of parties of S that are in the i-th bin.

The first part of the lemma is proven using the Chernoff bound as follows. By the linearity of the expectation, for $|S| \geq n/2$, $\mu_1 = \mathbb{E}[Y_1] \geq n^a/2$ and $\mu_2 = \mathbb{E}[Y_2] \geq (n - n^a)/2$. We get $\Pr[Y_1 < t_1] \leq e^{-\delta_1^2\mu_1/2} = e^{-9}$ and $\Pr[Y_2 < t_2] \leq e^{-\delta_2^2\mu_2/2} = e^{-9\cdot\frac{n}{n-n^a}} < e^{-9}$. Thus, by union bound over the two bins, the probability that S does not (t_1, t_2)-dominate a partition is at most $2e^{-9}$. Finally, if we independently sample ℓ partitions, we expect $\mu \leq 2\exp(-9)\ell$ undominated

partitions. By Chernoff bound, taking $\delta = \frac{\exp(9)}{10} - 1$, and counting in a random variable X the partitions that are not dominated by S, we get that

$$\Pr\Big[S \text{ } (0.8\ell,(t_1,t_2))\text{-dominates } \{P_j\}_{j\in[\ell]}\Big] \geq 1-\Pr[X \geq 0.2\ell] \geq 1-\exp\left(-\frac{\delta^2\mu}{2+\delta}\right) \geq 1-\exp\left(-\frac{\ell}{6}\right),$$

which is overwhelming for $\ell \geq 6\lambda$.

To prove the second part of the lemma, consider a fixed set $S \subseteq [n]$ with $|S| = n/2 - cn$, for some constant $0 < c < 1/2$. In order for S to be blocking for a partition $P = \{B_1, B_2\}$, it must hold that $|S \cap B_1| \geq |B_1| - t_1 = \frac{n^a}{2} + 3n^{a/2}$ or $|S \cap B_2| \geq |B_2| - t_2 = \frac{n-n^a}{2} + 3\sqrt{n}$. As we assume that $|S| \leq n/2 - cn$, we can ignore the second bin for sufficiently large n, since $n/2 - cn < |B_2| - t_2 = n/2 - o(n)$. We next find the probability that S is blocking for the bin B_1 of P. Let $Y_1 = |S \cap B_1|$ and $\mu = \mathbb{E}[Y_1] = \frac{n^a}{2} - cn^a$. By the Chernoff bound, the probability that S is blocking for the first bin satisfies

$$p_1 = \Pr\left[Y_1 \geq \frac{n^a}{2} + 3n^{a/2}\right] \leq \exp\left(-\frac{\alpha^2\mu}{2+\alpha}\right),$$

where α satisifies $(1+\alpha)\mu = \frac{n^a}{2} + 3n^{a/2}$. That is,

$$\alpha = \frac{3n^{a/2} + cn^a}{n^a/2 - cn^a} \geq \frac{cn^a}{n^a/2 - cn^a} \geq \frac{2c}{1-2c}.$$

Since the probability decreases as α increases, we get the following.

$$p_1 \leq \exp\left(-\frac{(2c/(1-2c))^2\mu}{2+(2c/(1-2c))}\right) \leq \exp\left(-\frac{c^2}{1-c}n^a\right).$$

Since the ℓ partitions are sampled independently, the probability that S $(\beta, \mathbf{r}_1)$-blocks $\{P_j\}_{j\in[\ell]}$, i.e., it is blocking for at least β partitions, is

$$\Pr[\text{Binom}(\ell, p_1) \geq \beta] \leq \sum_{j=\beta}^{\ell} \binom{\ell}{j} p_1^j.$$

Using the standard combinatorial bound $\binom{\ell}{j} \leq (e\ell/j)^j \leq (e\ell/\beta)^j$, which follows from Stirling's approximation [51], we can bound

$$\begin{aligned}\Pr[\text{Binom}(\ell, p_1) \geq \beta] &\leq \sum_{j=\beta}^{\infty} \left(\frac{e\ell p_1}{\beta}\right)^j \\ &\overset{\text{(geometric series)}}{=} \frac{\left(\frac{e\ell p_1}{\beta}\right)^\beta}{1-\frac{e\ell p_1}{\beta}} \overset{e\ell p_1/\beta < 1/2}{\leq} 2\left(\frac{e\ell p_1}{\beta}\right)^\beta.\end{aligned}$$

We can bound this by $\exp\left(\beta(\ln p_1 + 1 + \ln(\ell/\beta))\right)$. Hence, as $\beta = 0.2\ell$, we get that the probability is at most $\exp\left(0.2\ell(-c^2n^a/(1-c)+3)\right)$.

Finally, taking a union bound over all possible sets $S \subseteq [n]$ of size $n/2 - cn$, the probability that a $(\beta, \mathbf{r}_1)$-blocking set exists is at most

$$2^n \cdot \exp\left(0.2\ell(-c^2n^a/(1-c)+3)\right) = \exp\left(n + 0.2\ell(-c^2n^a/(1-c)+3)\right).$$

Taking

$$\ell = \frac{10n^{1-a}(1-c)}{c^2},$$

yields a probability of $\exp(-n + 6n^{1-a}(1-c)/(c^2)) < \exp(-\lambda)$ for sufficiently large n.

In order to prove the third part of the lemma, we first calculate the probability that a fixed set S $\mathbf{r}_2$-dominates a 2-unbalanced uniformly random partition $P = \{B_1, B_2\}$ as defined in the lemma. That is, fix a constant $0 < c < 1/2$, and let $|S| = n/2 - cn$. We calculate the probability that $|S \cap B_1| \geq t_1$. Let $Y_1 = |S \cap B_1|$ and $\mu = \mathbb{E}(Y_1) = \frac{n^a}{2} - cn^a$. By the Chernoff bound, the probability that $Y_1 \geq t_1$ is

$$p_2 = \Pr[Y_1 \geq \frac{n^a}{2} - 3n^{a/2}] \leq \exp\left(-\frac{\alpha^2\mu}{2+\alpha}\right),$$

where α satisifies $(1+\alpha)\mu = \frac{n^a}{2} - 3n^{a/2}$. That is,

$$\alpha = \frac{cn^a - 3n^{a/2}}{n^a/2 - cn^a} \geq \frac{cn^a - cn^a/2}{n^a/2 - cn^a} = \frac{c}{1-2c},$$

where the inequality follows since $8n^{a/2} = o(cn^a)$, thus for sufficiently large n, $3n^{a/2} \leq cn^a/2$. Hence, since the probability decreases as α increases, we get the following.

$$p_2 = \Pr[Y_1 \geq t_1] \leq \exp\left(-\frac{\alpha^2\mu}{2+\alpha}\right) = \exp\left(-\frac{c^2}{(1-2c)^2} \cdot \frac{n^a/2 - cn^a}{2 + c/(1-2c)}\right) = \exp\left(-\frac{c^2}{4-6c}n^a\right).$$

We proceed as in the second part of the lemma, replacing p_1 by p_2. The probability that there exists an S that $\mathbf{r}_2$-dominates at least $\beta = 0.2\ell$ partitions is

$$\exp\left(n + 0.2\ell\left(-\frac{c^2}{4-6c}n^a + 3\right)\right),$$

which is negligible in λ when

$$\ell = \frac{10n^{1-a}(4-6c)}{c^2}.$$

Finally, to prove the fourth part, notice that the probability that a given element $u \in [n]$ appears in the first bin of a partition P_j, i.e., $u \in B_1^j$, is $1/n^{1-a}$. Let X denote the number of partitions in which u appears in the first bin. Then

$$\mu = \mathbb{E}[X] = \frac{\ell}{n^{1-a}},$$

and setting $\delta = 1$, we have for $a \in (0,1)$:

$$\Pr\left[X \geq \frac{2\ell}{n^{1-a}}\right] = \Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\mu/3\right) = \exp\left(-\frac{\ell}{3n^{1-a}}\right).$$

Taking a union bound over all $u \in [n]$, the probability that there is a party that appears in the first bin of more than $2\ell/n^{1-a}$ partitions is at most

$$n \cdot \exp\left(-\frac{\ell}{3n^{1-a}}\right),$$

which is negligible in λ when

$$\ell = \Theta(\lambda n^{1-a}).$$

■

C Limit Elimination Compiler

Next, we extend Zhandry's limit elimination compiler [54, 28] to the multi-traitor and insider setting, resulting in the following theorem. In particular, the only difference between our construction and [54, 28] is that we need to take care of full and partial insiders. We explain how we can apply these modifications without affecting the traceability of the construction.

In order to simulate partial decryptions in the Trace algorithm, we require the underlying limited construction to support the following method.

Definition C.1 (The method SimPD). We say that a $\mathsf{TT\text{-}KEM}$ Π supports the simulation method SimPD with syntax

$$d_i \leftarrow \mathsf{SimPD}(\mathsf{pk}, i, r_{\mathsf{Enc}}),$$

if for all λ, $n \in \mathsf{poly}(\lambda)$, $r \in \mathcal{S}_n$, $0 \leq t \leq t(r)$, all $\varepsilon \in (0, 1/2)$, all $\mathsf{pp} \in \Pi.\mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}) \in \Pi.\mathsf{KGen}(1^\lambda, n, r, t, \varepsilon)$, all $i \in [n]$, and all $r_{\mathsf{Enc}} \in \mathcal{R}_\lambda$, it holds that

$$\mathsf{SimPD}(\mathsf{pk}, i, r_{\mathsf{Enc}}) = \mathsf{PDec}(\mathsf{sk}_i, c),$$

where $(k, c) \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}; r_{\mathsf{Enc}})$.

Then, the following theorem is proven by Construction 10.

Theorem C.2 (Limit Elimination Compiler). *If there exists a limited secure threshold KEM, Π_{Limited}, that is τ-traceable against (σ, ρ)-insiders, then there also exists a limitless secure threshold KEM that is τ-traceable against (σ, ρ)-insiders. The resulting scheme has the same public and secret key size, while the ciphertext size only grows by a factor of $\omega(\log \lambda)$, independent of the number of parties n. If $\rho > 0$, we require that Π_{Limited} supports the the simulation method SimPD.*

In the construction, we need the following theorem.

Theorem C.3 (Goldreich-Levin Decoding [40]). *There exists a constant Γ and an oracle algorithm $\mathsf{GL}^D(\ell, \varepsilon')$ running in time $\mathsf{poly}(\ell, \log(1/\varepsilon'))$ and making $\mathsf{poly}(\ell, \log(1/\varepsilon'))$ queries to D, such that the following holds. If there exists an $x \in \{0,1\}^\ell$ such that $\Pr[D(r) = x \cdot r \bmod 2 \colon r \xleftarrow{\$} \{0,1\}^\ell] \geq \frac{1}{2} + \varepsilon'$, then $\Pr[\mathsf{GL}^{D(\ell,\varepsilon')} = x] \geq \Gamma \cdot (\varepsilon')^2$.*

Construction 10. Let Π_{Limited} be a τ-traceable threshold KEM against (σ, ρ) insiders with limit ε. We construct Π' that is τ-traceable threshold KEM against (σ, ρ) without a limit, i.e., can trace any decoder with noticeable advantage.

$\Pi'.\mathsf{KGen}(1^\lambda, n, r, t)$**:** Sample $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk}) \xleftarrow{\$} \Pi_{\mathsf{Limited}}.\mathsf{KGen}(1^\lambda, n, r, t)$. Output $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[n]}, \mathsf{tk})$.

$\Pi'.\mathsf{Enc}(\mathsf{pk})$**:** Let $y = \omega(\log \lambda)$ and $z = z(\lambda)$ be a polynomial. Compute $(k_{u,v}, c_{u,v}) \xleftarrow{\$} \Pi_{\mathsf{Limited}}.\mathsf{Enc}(\mathsf{pk})$ for all $u \in [y], v \in [z]$. For all $v \in [z]$ let $k_v = k_{1,v} \oplus k_{2,v} \oplus \cdots \oplus k_{y,v}$. Sample $s \xleftarrow{\$} \mathcal{K}$ and let $b_v = \langle s, k_v \rangle \mod 2$ (bit-wise inner product) for all $v \in [z]$. Let $k = (b_1, \ldots, b_z) \in \{0,1\}^z$ and let $c = (s, (c_{u,v})_{u\in[y],v\in[z]})$. Output (k, c).

$\Pi'.\mathsf{PDec}(\mathsf{sk}_i, c)$**:** Parse c as $(s, (c_{u,v})_{u\in[y],v\in[z]})$. For all $u \in [y], v \in [z]$ compute partial decryptions $d_{i,u,v} \leftarrow \Pi_{\mathsf{Limited}}.\mathsf{PDec}(\mathsf{sk}_i, c_{u,v})$. Output $(d_{i,u,v})_{u\in[y],v\in[z]}$.

$\Pi'.\mathsf{Combine}(\mathsf{pk}, S, (d_i)_{i\in S}, c)$**.** Parse c as $(s, (c_{u,v})_{u\in[y],v\in[z]})$ and d_i as $(d_{i,u,v})_{u\in[y],v\in[z]}$ for all $i \in S$. Compute $\hat{k}_{u,v} = \Pi_{\mathsf{Limited}}.\mathsf{Combine}(\mathsf{pk}, S, \{d_{i,u,v}\}_{i\in S}, c_{u,v})$. For all $v \in [z]$ compute $\hat{k}_v = \hat{k}_{1,v} \oplus \cdots \oplus \hat{k}_{y,v}$ and $\hat{b}_v = \langle s, \hat{k}_v \rangle \mod 2$. Output $\hat{k} = (\hat{b}_1, \ldots, \hat{b}_z)$.

$\Pi'.\mathsf{Trace}^{D(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$**:** Construct the decoder D_3 as described below and output $\Pi_{\mathsf{Limited}}.\mathsf{Trace}^{D_3(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon, \mathcal{I}_{\mathsf{sk}}, \mathcal{I}_{\mathsf{pd}}, \{\mathsf{sk}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}\cup\mathcal{I}_{\mathsf{pd}}})$.

$D_1(s, (c_u)_{u\in[y]}, \{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_{i,u}\}_{i\in\mathcal{I}_{\mathsf{pd}},u\in[y]})$**:** The algorithm D_1 is constructed as follows:

- Sample a random $v \xleftarrow{\$} [z]$. For all $u \in [y]$, set $c_{u,v} = c_u$ and $d_{i,u,v} = d_{i,u}$ for all $i \in \mathcal{I}_{\mathsf{pd}}, u \in [y]$.
- For all $v' \in [z] \setminus \{v\}$ and all $u \in [y]$, compute $(k_{u,v'}, c_{u,v'}) \xleftarrow{\$} \Pi_{\mathsf{Limited}}.\mathsf{Enc}(\mathsf{pk}; r_{u,v'})$. Let $c = (s, (c_{u,v})_{u\in[y],v\in[z]})$.
- Compute $d_{i,u,v'} \leftarrow \Pi_{\mathsf{Limited}}.\mathsf{SimPD}(\mathsf{pk}, i, r_{u,v'})$ for all $i \in \mathcal{I}_{\mathsf{pd}}, u \in [y], v' \in [z] \setminus \{v\}$.
- For all $v' \in [z] \setminus \{v\}$ compute $k_{v'} = k_{1,v'} \oplus \cdots \oplus k_{y,v'}$.
- For all $v' \le v$ sample $b_v \xleftarrow{\$} \{0,1\}$ and for $v' > v$ let $b'_v = \langle s, k_{v'} \rangle \mod 2$. Set $k = (b_1, \ldots, b_z)$.
- Output $b_v \oplus D\Big(k, c, \{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_{i,u,v}\}_{i\in\mathcal{I}_{\mathsf{pd}},u\in[y],v\in[z]}\Big)$.

$D_2((c_u)_{u\in[y]}, \{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_{i,u}\}_{i\in\mathcal{I}_{\mathsf{pd}},u\in[y]})$**:** $\mathsf{GL}^{D_1(\cdot,(c_u)_{u\in[y]},\{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}},\{d_{i,u}\}_{i\in\mathcal{I}_{\mathsf{pd}},u\in[y]})}(\ell, \varepsilon' = \varepsilon/z)$.

$D_3(k, c, \{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$**:** Let (k, c) be the output of $\Pi_{\mathsf{Limited}}.\mathsf{Enc}$ and d_i be the (simulated) partial decryption of party i for the ciphertext c. The algorithm D_3 is constructed as follows.

- For $w \in [1, \zeta]$ where $\zeta = \left(\frac{2yz^3}{\Gamma\varepsilon^3}\right) \times \omega(\log\lambda)$:
  - Choose a random $u \in [y]$ and set $c_u = c$ and $d_{i,u} = d_i$ for all $i \in \mathcal{I}_{\mathsf{pd}}$.
  - For each $u' \in [y] \setminus [u]$, run $(k_{u'}, c_{u'}) \xleftarrow{\$} \Pi_{\mathsf{Limited}}.\mathsf{Enc}(\mathsf{pk}; r_{u'})$.
  - Compute $d_{i,u'} \leftarrow \Pi_{\mathsf{Limited}}.\mathsf{SimPD}(\mathsf{pk}, i, r_{u'})$ for all $i \in \mathcal{I}_{\mathsf{pd}}, u' \in [y] \setminus \{u\}$.
  - Run $k' \xleftarrow{\$} D_2((c_u)_{u\in[y]}, \{\overline{\mathsf{sk}}_i\}_{i\in\mathcal{I}_{\mathsf{sk}}}, \{d_{i,u'}\}_{u'\in[y],i\in\mathcal{I}_{\mathsf{pd}}})$ and set $\hat{k} = k' \oplus k_1 \oplus k_{u-1} \oplus k_{u+1} \oplus k_y$.
- If $k = \hat{k}$ for any $w \in [\zeta]$, output 0. Otherwise, output 1.

Changes Compared to [54, 28]. Our construction differs from [54, 28] in two respects. First, in the construction of the decoders D_1 and D_3, the tracer must simulate partial decryptions of parties in $\mathcal{I}_{\mathsf{pd}}$. These partial decryptions are always for ciphertexts generated by the tracer itself, and therefore can be simulated using the procedure SimPD from the underlying Π_{Limited}. Second, the decoders are forwarded the secret keys of parties in $\mathcal{I}_{\mathsf{sk}}$ as input by $\Pi_{\mathsf{Limited}}.\mathsf{Trace}$.

Correctness and Security. As discussed above, the construction is identical to [28] except for the trace algorithm. Thus, correctness and security follow directly from the construction of [28].

Lemma C.4 *Let $y = \omega(\log\lambda)$, let $\varepsilon' = \varepsilon/z$, and let $\zeta = \frac{2yz^3}{\Gamma\varepsilon^3} \cdot \omega(\log\lambda)$. Suppose $\ell = \omega(\log\lambda)$. If Π_{Limited} is a τ-threshold traitor tracing scheme against (σ, ρ) insiders with limit ε, then Π' is a limitless τ-threshold traitor tracing scheme against (σ, ρ). If $\rho > 0$, we require that Π_{Limited} supports the the simulation method SimPD.*

The above changes do not affect the decryption capability of the decoder D_3. Indeed, this follows from the fact that D is a good decoder with respect to $\mathcal{I}_{\mathsf{sk}}$ and $\mathcal{I}_{\mathsf{pd}}$, and that in the constructions of D_1, D_2, and D_3 we generate valid partial decryptions and provide valid secret keys for the parties in $\mathcal{I}_{\mathsf{sk}}$. Therefore, the traceability of the construction follows from the arguments of [54, 28]. The only additional consideration is that the reduction must simulate partial decryptions, which we sketch below.

Fix an adversary $\mathcal{A}$ for Π' and an inverse-polynomial ε. Let $\mathsf{GoodTr}_{\mathcal{A}}$, $\mathsf{BadTr}_{\mathcal{A}}$, and $\mathsf{GoodDec}_{\mathcal{A}}$ be the events defined in Definition 4.1 for the scheme Π'. Similarly, let $\mathcal{B}$ be an adversary against Π_{Limited}, and let $\mathsf{GoodTr}_{\mathcal{B}}$, $\mathsf{BadTr}_{\mathcal{B}}$, and $\mathsf{GoodDec}_{\mathcal{B}}$ be the corresponding events.

The reduction $\mathcal{B}$ simulates the tracing experiment for $\mathcal{A}$ in a straightforward manner. All public parameters and corruption queries are forwarded honestly. Eventually, $\mathcal{A}$ outputs a decoder D^*. The reduction $\mathcal{B}$ then constructs the decoder D_3^* exactly as in $\Pi'.\mathsf{Trace}$ and submits it to its challenger. This is possible since the reduction can simulate the partial decryptions using SimPD of the underlying Π_{Limited}.

Since the set of corrupted parties is identical in both experiments, if the event $\mathsf{BadTr}_{\mathcal{A}}$ occurs, then $\mathsf{BadTr}_{\mathcal{B}}$ occurs as well. Therefore, $\Pr[\mathsf{BadTr}_{\mathcal{A}}] \leq \Pr[\mathsf{BadTr}_{\mathcal{B}}]$.

The proof that $\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$ follows identically to [28, 54], with the only modification being the simulation of partial decryptions as described above.

D Deferred Proofs for our Compilers

D.1 Full Insiders to Partial Insiders Proof

Proof of Theorem 5.3. Fix arbitrary τ, σ, and ρ for a base scheme Π sucht that $0 < \tau \leq t$ and $0 < \tau + \sigma + \rho \leq t$ and assume that Π is τ-traceable against (σ, ρ) insiders. Let $\mathcal{A}$ be any PPT adversary against traceability of Π'.

In the following, we introduce a number of random variables for $\mathcal{A}$ running in $\mathsf{Game\text{-}Tr}$. Denote with $\mathcal{T}_{\mathcal{A}}$ the random variable capturing the set of traitors corrupted by $\mathcal{A}$ through its corruption oracle in $\mathsf{Game\text{-}Tr}$, and with $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ as well as $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$ the set of secret key and partial decryption insiders output by $\mathcal{A}$. Denote $\mathcal{I}'_{\mathcal{A}} = \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \cup \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$. W.l.o.g., assume that $\mathcal{I}'_{\mathcal{A}} \subseteq [n]$, $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \cap \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}} = \emptyset$, $|\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}| \leq \sigma$, and $|\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}| \leq \rho + \sigma - |\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}|$. Also, denote with D' the decoder produced by $\mathcal{A}$ and by $\mathcal{T}'_{\mathcal{A}}$ the set of traitors found when tracing D' against $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ and $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$. We aim to show that there exists a negligible function negl such that

$$\Pr[\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}} \ \wedge \ |\mathcal{T}'_{\mathcal{A}}| \geq \tau] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$$

as well as

$$\Pr[\mathsf{BadTr}_{\mathcal{A}}] \leq \mathsf{negl}(\lambda).$$

We construct a reductio $\mathcal{B}$ that tries to break τ-traceability against (σ, ρ)-insiders of Π as follows: When $\mathcal{B}$ receives public parameters and the public key from its challenger, $\mathcal{B}$ forwards those to $\mathcal{A}$. When $\mathcal{A}$ issues a corruption query, $\mathcal{B}$ forwards it to its challenger and returns the resulting secret key to $\mathcal{A}$. Similarly, $\mathcal{B}$ answers queries to $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ by querying its own challenger's partial decryption oracle. As a result, when $\mathcal{A}$ outputs $(D', \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}, \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}})$, it holds that $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$.

Given the decoder D', $\mathcal{B}$ must produce a decoder D to return to its challenger. To do so, $\mathcal{B}$ follows the procedure outlined in $\Pi'.\mathsf{Trace}$. In particular, $\mathcal{B}$ samples the sets $\mathcal{I}_{\mathsf{pd}\mathcal{B}}$ as an arbitrary subset of size $\min(\rho, |\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}|)$ of $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$. It computes $\mathcal{N}_{\mathsf{pd}} \leftarrow \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}} \setminus \mathcal{I}_{\mathsf{pd}\mathcal{B}}$ and sets $\mathcal{I}_{\mathsf{sk}\mathcal{B}} \leftarrow \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \cup \mathcal{N}_{\mathsf{pd}}$. It constructs a decoder D that, on input $(k, c, \{\overline{\mathsf{sk}_i}\}_{i \in \mathcal{I}_{\mathsf{sk}\mathcal{B}}}, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}\mathcal{B}}})$, computes $d_i \leftarrow \mathsf{PDec}(\overline{\mathsf{sk}_i}, c)$ for all $i \in \mathcal{N}_{\mathsf{pd}}$ and returns the result of the query

$$D'\left(k, c, \{\overline{\mathsf{sk}_i}\}_{i \in \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}}, \{d_i\}_{i \in \mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}}\right).$$

Note that the above is possible, because $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}} \subseteq \mathcal{I}_{\mathsf{sk}\mathcal{B}}$, $\mathcal{N}_{\mathsf{pd}} \subseteq \mathcal{I}_{\mathsf{sk}\mathcal{B}}$, i.e., D gets $\overline{\mathsf{sk}_i}$ as input for all $i \in \mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ and $i \in \mathcal{N}_{\mathsf{pd}}$. Also, observe that $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}} = \mathcal{I}_{\mathsf{pd}\mathcal{B}} \cup \mathcal{N}_{\mathsf{pd}}$, hence the decoder D can compute all necessary partial decryptions from its input. Finally, the reduction $\mathcal{B}$ outputs $(D, \mathcal{I}_{\mathsf{sk}\mathcal{B}}, \mathcal{I}_{\mathsf{pd}\mathcal{B}})$ to its challenger. Denote with $\mathcal{T}'_{\mathcal{B}}$ the set of traitors identified when D is traced by $\mathcal{B}$'s challenger. Observe that $\mathcal{B}$'s simulation is perfect, and thus $\mathcal{T}'_{\mathcal{B}} = \mathcal{T}'_{\mathcal{A}}$.

Analysis. First, observe that if D' is good for $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ and $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$, then D is also good for $\mathcal{I}_{\mathsf{sk}\mathcal{B}}$ and $\mathcal{I}_{\mathsf{pd}\mathcal{B}}$. Similarly, if D' is not good for $\mathcal{I}_{\mathsf{sk}}{}'_{\mathcal{A}}$ and $\mathcal{I}_{\mathsf{pd}}{}'_{\mathcal{A}}$, then D is also not good for $\mathcal{I}_{\mathsf{sk}\mathcal{B}}$ and $\mathcal{I}_{\mathsf{pd}\mathcal{B}}$. Hence, it holds that $\mathsf{GoodDec}_{\mathcal{A}} = \mathsf{GoodDec}_{\mathcal{B}}$ and thus $\Pr[\mathsf{GoodDec}_{\mathcal{A}}] = \Pr[\mathsf{GoodDec}_{\mathcal{B}}]$.

We start with BadTr: Clearly, if $\mathsf{BadTr}_{\mathcal{A}}$ happens, then it must hold that $\mathcal{T}'_{\mathcal{A}} \not\subseteq \mathcal{T}_{\mathcal{A}}$. Since $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}}$ and $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$, it must also holds that $\mathcal{T}'_{\mathcal{B}} \not\subseteq \mathcal{T}_{\mathcal{B}}$ and thus $\mathsf{BadTr}_{\mathcal{B}}$ happens. We get that $\Pr[\mathsf{BadTr}_{\mathcal{A}}] = \Pr[\mathsf{BadTr}_{\mathcal{B}}]$, which is negligible by assumption.

Next, we look at GoodTr: Suppose that $\mathsf{GoodTr}_{\mathcal{B}}$ happens. In this case, it holds that $\mathcal{T}'_{\mathcal{B}} \subseteq \mathcal{T}_{\mathcal{B}}$ and $|\mathcal{T}'_{\mathcal{B}}| \geq \tau$. Again, since $\mathcal{T}'_{\mathcal{A}} = \mathcal{T}'_{\mathcal{B}}$ and $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$, it follows that $\mathcal{T}'_{\mathcal{A}} \subseteq \mathcal{T}_{\mathcal{A}}$ and $|\mathcal{T}'_{\mathcal{A}}| \geq \tau$ whenever $\mathsf{GoodTr}_{\mathcal{B}}$ happens. We conclude that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodTr}_{\mathcal{B}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \mathsf{negl}(\lambda) = \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda),$$

for a negligible function $\mathsf{negl}(\lambda)$ by assumption of τ-traceability of Π against (σ, ρ) insiders. ■

D.2 Deferred Proofs of the 1-Partition Compiler

D.2.1 Traceability Proof of Construction 3

To prove Theorem 6.3, we first prove the following intuitive lemma, which argues that the decoders D_j constructed by the tracer in Construction 3 are equally good as the underlying decoder D.

Lemma D.1 *For all j it holds that $P_{\Pi'}(D) = P_{\Pi}(D_j)$, where D is the decoder that is traced in Π' and D_j is the decoder for the j-th instance of Π that is constructed by $\Pi'.\mathsf{Trace}$ (before it is potentially replaced with the dummy decoder).*

Proof of Lemma D.1. Fix an arbitrary $j \in [K]$ and let $\left(\mathsf{pk} := \{\mathsf{pk}_\ell\}_{\ell\in[K]}, \mathsf{sk}, \mathsf{tk}\right) \xleftarrow{\$} \Pi'.\mathsf{KGen}(*)$. Fix $\mathcal{I}_{\mathsf{pd}} \subseteq [n]$. Let D be an arbitrary decoder and D_j be the decoder that is constructed by $\Pi'.\mathsf{Trace}$ for instance j before it is potentially replaced with the dummy decoder. When instantiating the definition of P with Π', we get that

$$\begin{aligned}
&P_{\Pi'}(D)\\
&= \Pr\left[D\Big(k^b, (c_1,\ldots,c_K), \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big) = b\colon \begin{array}{c} \{(c_\ell, k_\ell) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_\ell)\}_{\ell\in[K]} \\ \{d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_\ell)\}_{\ell\in[K], i\in\mathcal{I}_{\mathsf{pd}}\cap B_\ell} \\ k^0 \xleftarrow{\$} \bigoplus_{\ell=1}^{K} k_\ell \\ k_1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \end{array} \right]\\
&= \Pr\left[D\Big(k, (c_1,\ldots,c_K), \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big) = b\colon \begin{array}{c} \{(c_\ell, k_\ell) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_\ell)\}_{\ell\in[K]} \\ \{d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_\ell)\}_{\ell\in[K], i\in\mathcal{I}_{\mathsf{pd}}\cap B_\ell} \\ k^0_j \leftarrow k_j, k^1_j \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \\ k \leftarrow k^b_j \oplus \bigoplus_{\ell\in[K]\setminus\{j\}} k_\ell \end{array} \right].
\end{aligned}$$

The above equality holds. To see this, observe that for $b = 0$, both experiments are identical. When $b = 1$, we have $k^b_j = k^1_1 \xleftarrow{\$} \mathcal{K}$, and thus the two ditribuisions are identical:

$$\left(k \leftarrow k^b_j \oplus \bigoplus_{\ell\in[K]\setminus\{j\}} k_\ell\right) \equiv \left(k \xleftarrow{\$} \mathcal{K}\right).$$

Next, let us instantiate $P_{\Pi}(D_j)$. We get that

$$
\begin{aligned}
&P_{\Pi}(D_j)\\
&= \Pr\left[D_j\Big(k_j^b, c_j, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}\cap B_j}\Big) = b\colon \begin{array}{c}(c_j, k_j^0) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_j)\\ \{d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_j)\}_{i\in\mathcal{I}_{\mathsf{pd}}\cap B_j}\\ k_j^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}\end{array} \right]\\
&= \Pr\left[D\Big(k, \{c_\ell\}_{\ell\in[K]}, \{d_i\}_{i\in\mathcal{I}_{\mathsf{pd}}}\Big) = b\colon \begin{array}{c}\{(c_\ell, k_\ell) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_\ell)\}_{\ell\in[K]\setminus\{j\}}\\ (c_j, k_j^0) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_j)\\ \{d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c_\ell)\}_{\ell\in[K], i\in\mathcal{I}_{\mathsf{pd}}\cap B_\ell}\\ k_j^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}\\ k \leftarrow k_j^b \oplus \bigoplus_{\ell\in[K]\setminus\{j\}} k_\ell\end{array} \right]\\
&= P_{\Pi'}(D),
\end{aligned}
$$

where in the second step, we inlined the construction of D_j with respect to D as defined by $\Pi.\mathsf{Trace}$. ∎

Equipped with Lemma D.1, we can now prove Theorem 6.3.

Proof of Theorem 6.3. Fix an arbitrary x with $0 \le x < K$ and assume that Π is τ-traceable against ρ partial insiders, where $\tau > 0$ and $\tau + \rho \le t_{\mathsf{bin}}$. Let $\tau' = (K-x)\cdot\tau$ and $\rho' = \rho + x\cdot(\rho+1)$. We aim to prove τ'-traceability against ρ' partial insiders of Π'.

Denote with $\mathcal{T}'$ the set of traitors identified as in $\mathsf{Game\text{-}Tr}$ (Figure 3). Denote for all $j \in [K]$ with $\mathcal{T}'_j = B_j \cap \mathcal{T}'$ the set of traitors identified from B_j and with $\mathcal{T}_j = \mathcal{T} \cap B_j$ the set of *real* traitors from B_j. In the following proof, assume w.l.o.g. that for all adversaries against $\mathsf{Game\text{-}Tr}$ of Π', it holds that $|\mathcal{I}_{\mathsf{pd}}| < \rho'$.

Intuitively, our proof works as follows: We will show that in all bins B_j where the adversary asked for at most ρ partial insiders, tracing will succeed, and we will find at least τ traitors due to τ-traceability against ρ partial insiders of Π. Then, a counting argument shows that if $|\mathcal{I}_{\mathsf{pd}}| \le \rho'$, there can be at most x bins B_j, where $|\mathcal{I}_{\mathsf{pd}}{}^{(j)}| > \rho$, and thus we find τ traitors in all remaining $K - x$ bins. We next formalize this intuition.

First, we define the following two events for all $j \in [K]$:

- $\mathsf{GoodTrBin}_j$: This event occurs whenever both $|\mathcal{T}'_j| \ge \tau$ and $\mathcal{T}'_j \subseteq \mathcal{T}_j$.
- $\mathsf{BadTrBin}_j$: This event occurs whenever $\mathcal{T}'_j \not\subseteq \mathcal{T}_j$.

Let $\mathcal{H}_0$ be the τ'-traceability game (Definition 4.1) against ρ' partial insiders, instantiated with Π'. For $u \in [K]$, let $\mathcal{H}_u$ be the same game instantiated except with a modified winning condition. In $\mathcal{H}_u$, the additional condition triggers if the adversary manages to avoid being traced or to frame an honest party in B_j for some $j \le u$. More formally, in $\mathcal{H}_u$, we set BadTr to 0, whenever $\mathsf{BadTrBin}_j = 1$ for some $j \le u$, and we set GoodTr to test, whenever there exists some $j \le u$ for which $\mathsf{GoodTrBin}_j = 0$ and $|\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho$, where test is the event that the decoder D passes the test described in the tracing procedure.

Now, observe that Π' is naively τ'-traceable against ρ' partial insiders with respect to $\mathcal{H}_K$. To show this, we prove that for all PPT adversaries $\mathcal{A}$ it holds that $\Pr[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}] \ge \Pr[\mathsf{GoodDec}_{\mathcal{A}}^{(K)}] - \mathsf{negl}(\lambda)$, and $\Pr[\mathsf{BadTr}_{\mathcal{A}}^{(K)}] = 0$. In particular, let $\mathcal{A}$ be any PPT adversary. If $\mathcal{A}$ hits any event $\mathsf{BadTrBin}_j$ for $j \in [K]$, then $\mathsf{BadTr}_{\mathcal{A}}^{(K)}$ is set to 0 by definition. If, however, $\mathcal{A}$ avoids triggering this condition, it must cause $\mathcal{T}'_j \subseteq \mathcal{T}_j$ for all $j \in [K]$. As a consequence, the event BadTr cannot occur, because

$$\mathcal{T}' = \bigcup_{j=1}^{K} \mathcal{T}'_j \subseteq \bigcup_{j=1}^{K} \mathcal{T}_j = \mathcal{T}.$$

A similar argument holds for $\mathsf{GoodTr}_{\mathcal{A}}^{(K)}$. Suppose $\mathcal{A}$ avoids hitting the condition, i.e. for all $j \in [K]$ with $|\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \leq \rho$ it holds that $\mathsf{GoodTrBin}_j = 1$. Then, $\mathcal{T}'_j \subseteq \mathcal{T}_j$ for all $j \in [K]$[20], and because $|\mathcal{T}'_j| \geq \tau$ for all $j \in [K]$ where $|\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \leq \rho$, it holds that

$$|\mathcal{T}'| = \sum_{j=1}^{K} |\mathcal{T}'_j| \geq \sum_{j \in [K] \text{ s.t. } |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \leq \rho} \tau \geq (K - x) \cdot \tau = \tau',$$

which causes $\mathsf{GoodTr}^{(K)}$ to be set to 1. If, however, $\mathcal{A}$ does trigger the additional condition, then $\mathsf{GoodTr}^{(K)}$ is set to test. It remains to show that we can bound the probability that test does not happen when $\mathcal{A}$ produces a good decoder with a negligible function.

Claim D.2 If D is ε-good, then $\mathsf{test} = 0$ can happen with at most negligible probability.

Proof of Claim D.2. Suppose D is ε-good. Let the random variable X count the number of successful tests on D and let N denote the number of trials. Further, we can bound the mean $\mu = N \cdot P(D) \geq N \cdot (1/2 + \varepsilon)$ We are interested in

$$\Pr[\mathsf{test} = 0 | D \text{ is } \varepsilon\text{-good}] = \Pr[X < N \cdot (1/2 + \varepsilon/2)],$$

and applying the Hoeffding inequality with $t = N\varepsilon/2$, we get that

$$\Pr[\mathsf{test} = 0 | D \text{ is } \varepsilon\text{-good}] \leq \Pr[X \leq \mu - t] \leq \exp\left(-\frac{N\varepsilon^2}{2}\right),$$

which is negligible for $N \geq 2\lambda/\varepsilon^2$. Since we set $Q = 8\lambda/\varepsilon^2 \geq 2\lambda/\varepsilon^2$ in the construction, the probability is negligible. ∎

As a result, we get that, $\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}\right] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$.

Hence, in order to show τ'-traceability, it remains to show that for all PPT adversaries $\mathcal{A}$, it holds that

$$\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(0)}\right] \geq \Pr[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}] - \mathsf{negl}(\lambda) \text{ and } \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(0)}\right] \leq \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(K)}\right] + \mathsf{negl}(\lambda)$$

for a negligible function negl.

Let $\mathcal{A}$ be any PPT adversary. We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ against τ-traceability of Π against ρ partial insiders as follows: First, $\mathcal{B}$ picks an index $u \xleftarrow{\$} [K]$ uniformly at random. For all indices $j \neq u$, $\mathcal{B}$ runs the key generation algorithm of Π to generate pk_j, $\{\mathsf{sk}_i\}_{i \in B_j}$, and tk_j according to protocol. $\mathcal{B}$ embeds the challenge public key pk^*, as received by $\mathsf{Game\text{-}Tr}$ of Π, for instance u, i.e., $\mathsf{pk}_u \leftarrow \mathsf{pk}^*$. Then, $\mathcal{B}$ runs $\mathcal{A}$, simulating corruption queries for parties in B_u by making use of its own corruption oracle for Π. If applicable, $\mathcal{B}$ simulates $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ queries as follows. Whenever $\mathcal{A}$ queries $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ on input (i, r_{Enc}), $\mathcal{B}$ parses r_{Enc} as $(r_{\mathsf{Enc}}^{(1)}, \ldots, r_{\mathsf{Enc}}^{(K)})$. Let $j \in [K]$ such that party $i \in B_j$.

- If $j \neq u$, $\mathcal{B}$ computes $(k, c) \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}_j; r_{\mathsf{Enc}}^{(j)})$ and returns $d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i, c)$.
- If $j = u$, $\mathcal{B}$ queries its own challenger's $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle on input $(i, r_{\mathsf{Enc}}^{(j)})$ and forwards the result to $\mathcal{A}$.

As a result, $\mathcal{B}$ receives a decoder D and the set $\mathcal{I}_{\mathsf{pd}}$. Next, $\mathcal{B}$ constructs the decoder D_u as prescribed by the Trace algorithm of Π'. To do so, $\mathcal{B}$ performs the test on the decoder D, and outputs a dummy decoder if the test fails. $\mathcal{B}$ simulates the test as follows: If $\rho' > 0$, $\mathcal{B}$ needs to

[20] Note that in those bins where $\mathcal{I}_{\mathsf{pd}}{}^{(j)} > \rho$, the construction sets $\mathcal{T}'_j$ to $\emptyset$ and thus $\mathcal{T}'_j \subseteq \mathcal{T}_j$ holds trivially for such bins.

generate partial decryptions on behalf of $\mathcal{I}_{\mathsf{pd}}$ for the ciphertexts used in the test. For all $j \neq u$, $\mathcal{B}$ can trivially generate partial decryptions on behalf of parties in ${\mathcal{I}_{\mathsf{pd}}}^{(j)}$, since the reduction knows all secret keys except for the challenge instance u. To simulate partial decryptions on behalf of parties in ${\mathcal{I}_{\mathsf{pd}}}^{(u)}$, $\mathcal{B}$ makes use of the $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$ oracle provided by its challenger as described above for the simulation of $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$. This is possible because $\mathcal{B}$ knows the randomness r_{Enc} for all ciphertexts it generates to test the decoder.

Then, $\mathcal{B}$ submits D_u and ${\mathcal{I}_{\mathsf{pd}}}^{(u)}$ to its challenger. Denote with $\mathsf{GoodTr}_{\mathcal{B}}$, $\mathsf{BadTr}_{\mathcal{B}}$, and $\mathsf{GoodDec}_{\mathcal{B}}$ the events GoodTr, BadTr, and $\mathsf{GoodDec}$ for $\mathcal{B}$ when running in the τ-traceability game of Π against ρ partial insiders.

Before we bound $\mathcal{A}$'s probability to distinguish adjacent hybrids with $\mathcal{B}$'s success probability, we first prove the following intuitive claim, which says that a decoder D_u constructed from a decoder D that tested positive is good with overwhelming probability.

Claim D.3 For all adversaries $\mathcal{A}$ and for all $j \in [K]$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr\Big[\mathsf{test}_{\mathcal{A}} = 1 \wedge |{\mathcal{I}_{\mathsf{pd}}}^{(j)}| \leq \rho\Big] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda).$$

Proof of Claim D.3. Suppose that $|{\mathcal{I}_{\mathsf{pd}}}^{(j)}| \leq \rho$, and thus $\mathcal{B}$ does not hit the abort condition to output a dummy decoder. Suppose that test happens. We will show that, in this case, $\mathcal{B}$ produces a good decoder with overwhelming probability or in other words $\Pr[\mathsf{test}_{\mathcal{A}} = 1 \wedge D_u \text{ is not } \varepsilon/4\text{-good}] \leq \Pr[\mathsf{test}_{\mathcal{A}} = 1 \mid D_u \text{ is not } \varepsilon/4\text{-good}] \leq \mathsf{negl}(\lambda)$. Denote with X the random variable counting the successful responses of D in the test. We can apply Lemma D.1 and bound

$$\begin{aligned}
&\Pr\Big[\mathsf{test} = 1 | P(D_u) < \frac{1}{2} + \frac{\varepsilon}{4}\Big] = \Pr\Big[\mathsf{test} = 1 \mid P(D) < \frac{1}{2} + \frac{\varepsilon}{4}\Big] \\
&\leq \Pr\Big[\mathsf{test} = 1 \mid P(D) = \frac{1}{2} + \frac{\varepsilon}{4}\Big] \\
&= \Pr\Big[X \geq N \cdot \frac{1}{2} + \frac{\varepsilon}{2} \mid P(D) = \frac{1}{2} + \frac{\varepsilon}{4}\Big],
\end{aligned}$$

where N denotes the number of repetitions in the test. Observe that, if $P(D) = 1/2 + \varepsilon/4$, then $\mu \coloneqq \mathbb{E}[X] = Q \cdot (1/2 + \varepsilon/4)$, and let $t = Q\varepsilon/4$. We can apply Hoeffding's inequality and bound

$$\Pr\Big[\mathsf{test} = 1 | P(D_u) < \frac{1}{2} + \frac{\varepsilon}{4}\Big] \leq \Pr[X \geq \mu + t] \leq \exp\left(-\frac{Q\varepsilon^2}{8}\right),$$

which is negligible for $Q \geq 8\lambda/\varepsilon^2$. Since we set $Q \coloneqq 8\lambda/\varepsilon^2$ in the construction, the claim holds. ■

Equipped with Claim D.3, we next bound $\mathcal{A}$'s advantage in distinguishing subsequent hybrids with respect to GoodTr and BadTr.

Claim D.4 For all $j \in [K]$ there exists a negligible function negl such that

$$\Pr\Big[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\Big] - \Pr\Big[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\Big] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda)$$

as well as

$$\Pr\Big[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\Big] - \Pr\Big[\mathsf{BadTr}_{\mathcal{A}}^{(j)}] = \Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j\Big].$$

Proof of Claim D.4. We start with the latter part of the claim. Observe that the only way for $\mathcal{A}$ to set $\mathsf{BadTr} = 1$ in $\mathcal{H}_{j-1}$ but not in $\mathcal{H}_j$ is to cause $\mathsf{BadTrBin}_j = 1$, i.e., causing the tracing procedure to output some $\mathcal{T}'_j$ such that $\mathcal{T}'_j \not\subseteq \mathcal{T}_j$. Denote with $\mathcal{T}'_{j,\mathcal{A}}$ and $\mathcal{T}_{j,\mathcal{A}}$ the random variable describing the set of identified traitors in B_j and the set of real traitors from B_j, respectively, when $\mathcal{A}$ runs in the tracing experiment of Π'. Similarly, denote with $\mathcal{T}'_{j,\mathcal{B}}$ and $\mathcal{T}_{j,\mathcal{B}}$ the same

variables, when $\mathcal{B}$ runs in the tracing experiment of Π. Applying the observation above, we can impose the following bound:

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right] = \Pr\left[\mathcal{T}'_{j,\mathcal{A}} \not\subseteq \mathcal{T}_{j,\mathcal{A}}\right].$$

By construction of $\Pi'.\mathsf{Trace}$ and $\mathcal{B}$, it must also hold that

$$\Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] = \Pr\left[\mathcal{T}'_{j,\mathcal{B}} \not\subseteq \mathcal{T}_{j,\mathcal{B}} \mid \mathcal{B}.u = j\right] = \Pr\left[\mathcal{T}'_{j,\mathcal{A}} \not\subseteq \mathcal{T}_{j,\mathcal{A}}\right],$$

and thus

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right] = \Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j].$$

Next, we apply a similar argument to the first part of the claim. Again, the only way that $\mathsf{GoodTr}_{\mathcal{A}}^{(j)} = 1$, but $\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)} = 0$ is if $\mathcal{A}$ has hit the modified condition in $\mathcal{H}_j$, i.e., $\mathsf{GoodTrBin}_j = 0$, $|\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho$, and $\mathsf{test} = 1$. Applying the law of total probability, we get that

$$\begin{aligned}
&\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right] \\
\le &\Pr\left[\neg\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho \wedge \mathsf{test}\right] \\
= &\Pr\left[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho \wedge \mathsf{test}\right].
\end{aligned}$$

Again, by the law of total probability, this is equal to

$$\begin{aligned}
&\Pr\left[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] \\
&\qquad + \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho \wedge \mathsf{test} = 0\right]
\end{aligned}$$

Now, consider that, whenever $\mathsf{test} = 0$, $\mathcal{B}$ outputs a dummy decoder. Hence, we get that

$$\begin{aligned}
&\Pr\left[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] \\
&\quad + \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho \wedge \mathsf{test} = 0\right] \\
&\le \Pr\left[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] + \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}} \mid \mathsf{test} = 0] \\
&\le \Pr\left[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] \\
&\quad + \Pr\left[\Pi.\mathsf{Trace}^{\mathsf{Dummy}(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon/4) \neq \emptyset\right],
\end{aligned}$$

where pk, tk, and $\mathcal{I}_{\mathsf{pd}}$ are treated as random variables. Observe that due to the unframeability of Π, its tracing algorithm can never output a non-empty set of traitors when queried with the dummy decoder, except with negligible probability. It is easy to see that otherwise the trivial adversary who corrupts no one and outputs $D = \mathsf{Dummy}$ would have a non-negligible advantage against the traceability of Π. Hence, there exists a negligible function negl such that

$$\Pr\left[\Pi.\mathsf{Trace}^{\mathsf{Dummy}(\cdot)}(\mathsf{pk}, \mathsf{tk}, \varepsilon/4) \neq \emptyset\right] \le \mathsf{negl}(\lambda).$$

Applying additionally Claim D.3 to substitute $\Pr[\mathsf{test} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho]$, and simplifying negligibilities yields the following:

$$\begin{aligned}
&\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right] \\
&\le \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr\left[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(j)}| \le \rho\right] + \mathsf{negl}(\lambda) \\
&= \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr\left[\mathcal{T}'_{\mathcal{B}} \subseteq \mathcal{T}_{\mathcal{B}} \wedge |\mathcal{T}'_{\mathcal{B}}| \ge \tau \wedge |\mathcal{I}_{\mathsf{pd}}{}^{(u)}| \le \rho \mid \mathcal{B}.u = j\right] + \mathsf{negl}(\lambda) \\
&= \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{j,\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda).
\end{aligned}$$

■

Making use of Claim D.4, we get:

$$\begin{aligned}
&\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(0)}\right] = \sum_{j=1}^{K}\left(\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right]\right)\\
&\overset{\text{Claim D.4}}{\leq} \sum_{j=1}^{K}\left(\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}'(\lambda)\right)\\
&= \sum_{j=1}^{K}\left(\frac{\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.u = j]}{\Pr[\mathcal{B}.u = j]} + \mathsf{negl}'(\lambda)\right)\\
&= K \cdot \sum_{j=1}^{K}\left(\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.u = j]\right) + \mathsf{negl}'(\lambda)\\
&= K \cdot \left(\Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \Pr[\mathsf{GoodTr}_{\mathcal{B}}]\right) + K \cdot \mathsf{negl}'(\lambda)\\
&\leq K \cdot \mathsf{negl}(\lambda)
\end{aligned}$$

for a negligible function negl, following traceability of Π. Hence, we get that

$$\begin{aligned}
\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(0)}\right] &\geq \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}\right] - K \cdot \mathsf{negl}(\lambda)\\
&\geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - K \cdot \mathsf{negl}(\lambda).
\end{aligned}$$

A similar argument applies to BadTr:

$$\begin{aligned}
&\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(0)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(K)}\right]\\
&= \sum_{j=1}^{K}\left(\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right]\right)\\
&\overset{\text{Claim D.4}}{=} \sum_{j=1}^{K}\Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j]\\
&= \sum_{j=1}^{K}\frac{1}{\Pr[\mathcal{B}.u = j]}\Pr[\mathsf{BadTr}_{\mathcal{B}} \wedge \mathcal{B}.u = j]\\
&= K \cdot \sum_{j=1}^{K}\Pr[\mathsf{BadTr}_{\mathcal{B}} \wedge \mathcal{B}.u = j]\\
&= K \cdot \Pr[\mathsf{BadTr}_{\mathcal{B}}]\\
&\leq K \cdot \mathsf{negl}(\lambda),
\end{aligned}$$

for a negligible function $\mathsf{negl}(\lambda)$. Hence, we get that

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(0)}\right] \leq \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(K)}\right] + K \cdot \mathsf{negl}(\lambda) = K \cdot \mathsf{negl}(\lambda).$$

■

D.2.2 Correctness, Robustness, and Semantic Security of Construction 3

Lemma D.5 *For $\delta \geq 2\sqrt{\lambda K/n}$, Construction 3 is probabilistically correct, $\frac{n(1+\delta)}{2K}$-robust, and semantically secure.*

Proof of Probabilistic Correctness of Construction 3. Consider a random set $|S| \geq n/2$. We prove in Part 1 of Lemma 6.1 that S has at least t_{bin} parties from every one of the K bins. Thus, correctness follows. ■

Proof of Robustness of Construction 3. Clearly, an adversary can only break robustness by causing sufficient crashes in at least one bin to break decryption. Since Π is assumed to be perfectly robust, any adversary must crash more than

$$\frac{n}{K} - t_{\mathsf{bin}} = \frac{n}{2K} \cdot (1+\delta)$$

parties in any of the bins. ∎

Proof of Semantic Security of Construction 3. Fix arbitrary $\lambda, n \in \mathbb{N}$, and $\varepsilon \in (0, 1/2]$. Let $r = n/2$, $t = (n/2) \cdot (1-\delta)$, and $t_{\mathsf{bin}} = t/K$.

Let $\mathcal{A}$ be any PPT adversary against $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}(1^\lambda, n, r, t, \varepsilon)$. We construct a PPT reduction $\mathcal{B}$ against $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi}(1^\lambda, n/K, t_{\mathsf{bin}}, t_{\mathsf{bin}}, \varepsilon/4)$ as follows: First, $\mathcal{B}$ runs $\mathcal{A}(1^\lambda, n, r, t, \varepsilon)$ to obtain a set of corrupted parties Cor. Assume w.l.o.g. that $\mathsf{Cor} \subseteq [n]$ and $|\mathsf{Cor}| < t$. Then, $\mathcal{B}$ simulates key generation by first sampling a random (K, n)-balanced partition $P = \{B_j\}_{j\in[K]}$. $\mathcal{B}$ then chooses any $j^* \in [K]$ such that $|B_{j^*} \cap \mathsf{Cor}| < t_{\mathsf{bin}}$. Note that such j^* must always exist if $|\mathsf{Cor}| < t$, because $t_{\mathsf{bin}} \cdot K = t > |\mathsf{Cor}|$. Then, $\mathcal{B}$ forwards $\mathsf{Cor}_{j^*} = \mathsf{Cor} \cap B_{j^*}$ to its challenger,[21] and receives $\mathsf{pk}_{j^*}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}_{j^*}}, \mathsf{tk}^*_j$, as well as the challenge (k^*_j, c^*_j). Then, $\mathcal{B}$ simulates key generation in all remaining bins according to protocol, generating

$$\left(\mathsf{pk}_j, \{\mathsf{sk}_i\}_{i\in B_j}, \mathsf{tk}_j\right) \xleftarrow{\$} \Pi.\mathsf{KGen}(1^\lambda, B_j, t_{\mathsf{bin}}, t_{\mathsf{bin}}, \varepsilon/4) \quad \text{for all } j \neq j^*.$$

To simulate the challenge ciphertext, $\mathcal{B}$ samples $(k_j, c_j) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}_j)$ for all $j \neq j^*$, and sets $k^* \leftarrow \bigoplus_{j\in[K]} k_j$ as well as $c^* \leftarrow \{c_j\}_{j\in[K]}$. $\mathcal{B}$ runs

$$\mathcal{A}\left(\{\mathsf{pk}_j\}_{j\in[K]}, \{\mathsf{tk}_j\}_{j\in[K]}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}}, k^*, c^*\right),$$

obtaining the bit b', and outputs the same bit b' to its challenger.

Analysis. Let b be the internal bit of $\mathcal{B}$'s challenger in $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi}_{\mathcal{B}}$ and let β be the hypothetical internal bit of $\mathcal{B}$'s simulation of $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}_{\mathcal{A}}$. If $b = 0$, then k_{j^*} is a *real* key, and thus $\mathcal{B}$ simulates $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}_{\mathcal{A}}$ perfectly for $\beta = 0$. If $b = 1$, then k_{j^*} is a uniform random value from $\mathcal{K}_\lambda$, and thus k^* is also uniformly random by the perfect security of the one-time pad, and thus $\mathcal{B}$ perfectly simulates $\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}_{\mathcal{A}}$ for $\beta = 1$. We get that

$$\Pr[\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}_{\mathcal{A}} = 1] = \Pr[\mathsf{Game\text{-}IND\text{-}CPA}^{\Pi'}_{\mathcal{B}} = 1],$$

which is $\leq 1/2 + \mathsf{negl}(\lambda)$ by assumption for a negligible function negl. ∎

D.3 A Proof of Theorem 6.4 (Multi-Instance Expansion Retains Traceability)

Proof of Theorem 6.4. Let $\mathcal{A}$ be a PPT adversary against τ-traceability of Π' against ρ partial insiders, and let ε be any inverse polynomial. We will bound the advantage of $\mathcal{A}$ in breaking traceability by the advantage of a PPT reduction $\mathcal{B}$ against traceability of the underlying scheme Π. We start by showing that, whenever $\mathcal{A}$ frames an honest party (i.e., causes $\mathsf{BadTr} = 1$), then $\mathcal{B}$ does so with probability $1/\ell$, and thus

$$\Pr[\mathsf{BadTr}_{\mathcal{A},\Pi'}] \leq \ell \cdot \Pr[\mathsf{BadTr}_{\mathcal{B},\Pi}] \leq \mathsf{negl}(\lambda),$$

for a negligible function negl. To see this, consider a PPT reduction $\mathcal{B}$ that initially samples a random index $j^* \in [\ell]$. Upon receiving a challenge public key pk^* from its Π-challenger, $\mathcal{B}$ simulates key generation by embedding its challenge public key at index j^*: $\mathsf{pk}^{(j^*)} \leftarrow \mathsf{pk}^*$. For all other instance, $\mathcal{B}$ simulates key generation according to protocol.

Simulating $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$: On input (i, r_{Enc}), $\mathcal{B}$ parses r_{Enc} as $(k, r^{(1)}_{\mathsf{Enc}}, r^{(2)}_{\mathsf{Enc}}, \ldots, r^{(\ell)}_{\mathsf{Enc}})$, where $r^{(j)}_{\mathsf{Enc}}$ is the randomness used to encrypt in instance $j \in [\ell]$. $\mathcal{B}$ then computes $d_i = \{d_i^{(j)}\}_{j\in[\ell]}$ as follows:

[21] To avoid unnecessary notation overhead of relabeling parties, assume that the challenger runs key genereation for participant indices $\mathcal{I}_{\mathsf{pd}}$.

- For $j \neq j^*$, compute $(k_j, c_j) \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}^{(j)}; r_{\mathsf{Enc}}^{(j)})$ and $d_i^{(j)} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(j)})$.
- For $j = j^*$, $\mathcal{B}$ queries $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}}^{(j^*)})$ to obtain $d_i^{(j^*)}$ from its challenger.

To simulate the corruption oracle queries $\mathcal{O}^{\mathsf{corrupt}}(i)$, $\mathcal{B}$ queries the corruption oracle of its own challenger to obtain $\mathsf{sk}_i^{(j^*)}$. Hence, it holds that $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$. All other shares of party i are known to the reduction. When $\mathcal{A}$ outputs a decoder D, the reduction constructs D_{j^*} as indicated by the $\Pi'.\mathsf{Trace}$ and outputs it to its challenger.

Observe that $\mathcal{B}$ simulates Π' perfectly to $\mathcal{A}$, and in particular, $\mathcal{A}$'s view is independent of $\mathcal{B}$'s choice of j^*. Suppose that $\mathsf{BadTr}_{\mathcal{A},\Pi'}$ happens, i.e., $\mathcal{T}'_{\mathcal{A}} \not\subseteq \mathcal{T}_{\mathcal{A}}$. Then, there must exists at least one index $j \in [\ell]$ such that $\mathcal{T}'_{j,\mathcal{A}} \not\subseteq \mathcal{T}_{\mathcal{A}}$. $\mathcal{B}$ will choose this index with probability $1/\ell$, and thus

$$\Pr[\mathsf{BadTr}_{\mathcal{A}}] = \Pr[\mathcal{T}'_{\mathcal{A}} \not\subseteq \mathcal{T}_{\mathcal{A}}] \leq \ell \cdot \Pr[\mathcal{T}'_{\mathcal{B}} \not\subseteq \mathcal{T}_{\mathcal{B}}] = \ell \cdot \Pr[\mathsf{BadTr}_{\mathcal{B}}].$$

Next, we will show that if D is a good decoder, then at least one decoder D_j produced by $\Pi'.\mathsf{Trace}$ is good with overwhelming probability, and will be traced.

Suppose D is a good decoder, i.e., $\mathsf{GoodDec}$ happens. By definition, this means that

$$\Pr\left[D\left(\{(c_j,u_j)\}_{j\in[\ell]}, k^b, \{d_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}}, j\in[\ell]}\right) : \begin{array}{c} b \xleftarrow{\$} \{0,1\}, (k^0,k^1) \xleftarrow{\$} \mathcal{K}^2 \\ \text{For all } j\in[\ell]\colon \\ (c_j,k_j) \xleftarrow{\$} \Pi.\mathsf{Enc}(\mathsf{pk}^{(j)}) \\ u_j \leftarrow k_j \oplus k^0 \\ \forall i \in \mathcal{I}_{\mathsf{pd}}\colon d_i^{(j)} \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(j)}, c_j) \end{array} \right] \geq \frac{1}{2} + \varepsilon.$$

We define a series of distributions $X_j(b)$ for $j \in [0,\ell]$: Let $X_0(b)$ be the above distribution on the tuple $\left(\{(c_q,u_q)\}_{q\in[\ell]}, k^b, \{d_i^{(q)}\}_{i\in\mathcal{I}_{\mathsf{pd}}, q\in[\ell]}\right)$, where b is fixed. For all $j \in [\ell]$, let X_j be the same distribution, except that for all $q \leq j$, u_q is replaced with a uniformly random element from $\mathcal{K}$, respectively. We set $p_j := \Pr[D(X_j(b)) = b\colon b \xleftarrow{\$} \{0,1\}]$. Then, by definition, it holds that $p_0 \geq 1/2 + \varepsilon$. Also, observe that in $X_\ell(b)$, all u_j are just uniform random elements from $\mathcal{K}$ and thus $X_\ell(b)$ does not depend on b. Hence, $X_\ell(0)$ and $X_\ell(1)$ are identically distributed and $p_\ell = 1/2$. As a consequence, there must be at least one $j \in [\ell]$ such that $p_j \leq p_{j-1} - \varepsilon/\ell$. Let j^* be such a j and consider the decoder D_{j^*} on input $(c_{j^*} k_{j^*}, \{d_i^{(j^*)}\}_{i\in\mathcal{I}_{\mathsf{pd}}})$. Suppose k_{j^*} is the correct key, i.e., (c_{j^*}, k_{j^*}) is sampled from $\Pi.\mathsf{Enc}(\mathsf{pk}^{(j^*)})$. In this case, D_{j^*} samples a random bit b and outputs $b \oplus D(X_{j^*-1}(b))$. As a result, D_{j^*} outpus 0, whenever D correctly guesses b and 1 otherwise, which means if k_{j^*} is valid, then D_{j^*} outputs 0 with probability p_{j^*-1}, i.e., D_{j^*} correctly guesses that k_{j^*} is valid with probability p_{j^*-1}. Also, observe that if k_{j^*} is invalid (i.e., if it does not fit c_{j^*}), then D_{j^*} runs D on distribution $X_{j^*}(b)$ for a random bit b and outputs 1, whenever D incorrecly guesses b. Hence D_{j^*} correctly guesses that k_{j^*} is invalid with probability $1 - p_{j^*}$. As a consequence D_{j^*} correctly *predicts* the validity of k_{j^*} with the following probability:

$$\frac{1}{2}(p_{j^*-1} + 1 - p_{j^*}) = \frac{1}{2} + \frac{p_{j^*-1} - p_{j^*}}{2} \geq \frac{1}{2} + \frac{\varepsilon}{2\ell}.$$

Above, we have shown that, if D is an ε-good decoder for Π', there must exist some D_j^* that is $\varepsilon/2\ell$-good with respect to Π.

Suppose for now that we use exactly the same reduction as in the BadTr case. Since this reduction commits in the beginning on a random index j^* to attack, we would get that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodTr}_{\mathcal{B}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \mathsf{negl}(\lambda) \geq \frac{\Pr[\mathsf{GoodDec}_{\mathcal{A}}]}{\ell} - \mathsf{negl}(\lambda),$$

which is insufficient by a factor of $1/\ell$.

To fix this, we modify the reduction to test the decoders in a certain way such that the (modified) reduction $\mathcal{B}$ can only ever trigger $\mathsf{GoodTr}_{\mathcal{B}}$ if it guesses j^* correctly, which essentially

concentrates the ability of $\mathcal{B}$ to achieve GoodTr to only happen for correct guesses, in which case it implies $\mathsf{GoodTr}_{\mathcal{A}}$. This will allow us to show that

$$\Pr[\mathsf{GoodTr}_{\mathcal{B}}] \leq \overbrace{\frac{\ell-1}{\ell}\cdot 0}^{\text{incorrect guess}} + \overbrace{\frac{1}{\ell}\cdot \Pr[\mathsf{GoodTr}_{\mathcal{A}}]}^{\text{correct guess}} = \frac{\Pr[\mathsf{GoodTr}_{\mathcal{A}}]}{\ell},$$

which eliminates the loss of ℓ encountered in the naive approach.

To achieve this, the reduction $\mathcal{B}$ constructs *all* decoders, and tests whether each D_j is good. For $j = j^*$, $\mathcal{B}$ constructs D_j^* according $\Pi'.\mathsf{Trace}$. In particular, it can generate all partial decryptions on behalf of $\mathcal{I}_{\mathsf{pd}}$ on the generated ciphertexts c_j, where $j \neq j^*$, because it knows all $\mathsf{sk}_i^{(j)}$ for $j \neq j^*$.

For all decoders D_j, where $j \neq j^*$, $\mathcal{B}$ can simulate the behavior of the decoder and generate $d_i^{(j^*)}$ on behalf of $i \in \mathcal{I}_{\mathsf{pd}}$ by querying its challenger's partial decryption oracle $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}$. Note that simulation is possible, because when given on input $(c_j, k_j, \{d_i^{(j)}\}_{i\in\mathcal{I}_{\mathsf{pd}}})$, the decoder D_j is supposed to generate c_{j^*} *honestly*, and thus $\mathcal{B}$ knows $r_{\mathsf{Enc}}^{(j^*)}$. For all other ciphertexts c_q where $q \neq j$ and $q \neq j^*$, $\mathcal{B}$ again simulates partial decryptions using the known secret keys.

Then $\mathcal{B}$ performs the following test on all simulated decoders D_j ($j \neq j^*$) and the decoder D_{j^*} Let $Q := 32\ell^2\lambda/\varepsilon^2$, $ctr := 0$ and repeat Q times:

- Sample $r_{\mathsf{Enc}} \xleftarrow{\$} \Pi.\mathcal{R}_{\mathsf{Enc}}$ and generate $(c, k^0) \leftarrow \Pi.\mathsf{Enc}(\mathsf{pk}^{(j)}; r_{\mathsf{Enc}})$. Sample $k^1 \xleftarrow{\$} \mathcal{K}$, and $b \xleftarrow{\$} \{0,1\}$. If $j \neq j^*$, compute $d_i \leftarrow \Pi.\mathsf{PDec}(\mathsf{sk}_i^{(j)}, c)$ for all $i \in \mathcal{I}_{\mathsf{pd}}$. If $j = j^*$, obtain $d_i^{(j)}$ by querying the challenger on $\mathcal{O}^{\mathsf{PDec\text{-}Hon}}(i, r_{\mathsf{Enc}})$.
- Run $b' \xleftarrow{\$} D_j(c, k^b)$. If $b' = b$, increment ctr.
- If $ctr \geq Q \cdot (1/2 + 3\varepsilon/8\ell)$, set the flag $\mathsf{test}_j \leftarrow 1$ (in this case, we say that D_j "tested good"). Otherwise, set $\mathsf{test}_j \leftarrow 0$.

Suppose $\mathcal{B}$ has chosen index j^* to embed the instance of the challenger initially. We make the following modification to the decoder that is returned by $\mathcal{B}$: If $\mathsf{test}_{j^*} = 1$ and $\mathsf{test}_j = 0$ for all $j < j^*$ (i.e., if D_{j^*} is the *first* decoder that tested good) then $\mathcal{B}$ outputs D_{j^*} to the challenger as before. Otherwise, however, $\mathcal{B}$ instead outputs a dummy decoder Dummy that always outputs 0.

Before we can close the argument above, we first need to show that (a) if D is a good decoder, then with overwhelming probability at least one D_j will test good, and (b) except with negligible probability, it can not happen that D_j tests good, but is not $\varepsilon/4\ell$-good for instance j of Π.

Claim D.6 If $\mathsf{GoodDec}$ happens, then with overwhelming probability, at least one D_j will test good.

Proof of Claim D.6. Our PLBE-style analysis has already shown that, if D is ε-good, then for at least one j it must hold that D_j is $\varepsilon/2\ell$-good. Suppose w.l.o.g. that $P(D_j) = 1/2 + \varepsilon/2\ell$, then by Hoeffding's inequality the probability that $ctr < Q \cdot (1/2 + 3\varepsilon/8\ell)$ is negligible in λ if $Q \geq 32\ell^2\lambda/\varepsilon^2$. ∎

Claim D.7 For all $j \in [\ell]$, the event that D_j tests good, but is not $\varepsilon/4\ell$-good with respect to instance j can happen only with negligible probability.

Proof of Claim D.7. Again, applying Hoeffding's inequality yields a negligible bound for $Q \geq 32\ell^2\lambda/\varepsilon^2$. ∎

Let s_j be the probability that test_j happens and test_q does not happen for any $q < j$. Following Claim D.6, we get that $\sum_{j\in[\ell]} s_j \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$ and Claim D.7 says that $\Pr[\mathsf{GoodDec}_{\mathcal{B}}] \geq 1/\ell \sum_{j\in[\ell]} s_j - \mathsf{negl}(\lambda)$, which when combined gives us

$$\Pr[\mathsf{GoodDec}_{\mathcal{B}}] \geq \frac{\Pr[\mathsf{GoodDec}_{\mathcal{A}}]}{\ell} - \mathsf{negl}(\lambda).$$

Next, we make use of the fact that, except with negligible probability, $\mathcal{B}$ will only trace good if it guesses the correct j^* such that $\mathsf{test}_{j^*} = 1$ and $\mathsf{test}_j = 0$ for all $j < j^*$. Observe that the respective events that test_j happens are independent of $\mathcal{B}$'s choice of j^*. Hence, given that $\mathsf{test}_j = 1$ for at least one j, $\mathcal{B}$ will play against that instance with probability $1/\ell$. If $\mathcal{B}$ guesses incorrectly, then $\mathcal{B}$ outputs a dummy decoder. In this case, $\mathsf{GoodDec}_{\mathcal{B}}$ must not happen because $P(\mathsf{Dummy}) = 1/2$ and $\mathsf{GoodTr}_{\mathcal{B}}$ can happen only with negligible probability, since otherwise a trivial adversary that corrupts no parties and outputs the dummy decoder could frame honest parties with non-negligible probability in Π. If, however, $\mathcal{B}$ guesses correctly, then $\mathsf{GoodTr}_{\mathcal{B}}$ implies $\mathsf{GoodTr}_{\mathcal{A}}$, because $\mathcal{B}$ corrupts the same parties as $\mathcal{A}$ and

$$\Pr[\mathsf{GoodTr}_{\mathcal{B}}] \leq \frac{\ell - 1}{\ell} \cdot \mathsf{negl}(\lambda) + \frac{\Pr[\mathsf{GoodTr}_{\mathcal{A}}]}{\ell}.$$

Putting it all together (and simplifying negligibilities), we get that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda),$$

for a negligible function negl. ∎

Lemma D.8 (Semantic Security of Π'). *If Π is a semantically secure* TT-KEM*, then multi-instance expansion of Π across $\ell \in \mathsf{poly}(\lambda)$ instances retains semantic security.*

Proof of Lemma D.8. Let Π be a semantically secure TT-KEM, and let Π' be a multi-instance expansion of Π across $\ell \in \mathsf{poly}(\lambda)$ bins. Then, due to the nature of 1-out-of-ℓ sharing the KEM key across instances, a simple hybrid argument shows that for all PPT adversaries $\mathcal{A}$ there exists a PPT reduction $\mathcal{B}$ such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Pi'}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B},\Pi}(\lambda),$$

which is negligible by assumption. ∎

D.4 A Proof of Theorem 7.1

We start by proving the following lemma.

Lemma D.9 *For all $u \in K$ where $\mathsf{test}_u = 1$ it holds that $P_{u,\Pi'}(D) = P_{\Pi}(D_u)$, where D is the decoder that is traced in Π' and D_u is the decoder for the u-th instance of Π that is constructed by $\Pi'.\mathsf{Trace}$.*

Proof of Lemma D.9. Fix an arbitrary $u \in [K]$ and let $\left(\mathsf{pk} := \{\mathsf{pk}^{(\ell)}\}_{\ell \in [K]}, \mathsf{sk}, \mathsf{tk}\right) \xleftarrow{\$} \Pi'.\mathsf{KGen}(*)$. Let D be an arbitrary decoder and D_u be the decoder that is constructed by $\Pi'.\mathsf{Trace}$ for instance u before it is potentially replaced with the dummy decoder. When instantiating the definition of P with Π', we get that

$$\begin{aligned}
&P_{u,\Pi'}(D) \\
&= \Pr\left[D\Big(k^b, ((c_1, \ldots, c_K), u)\Big) = b \colon \begin{array}{c} \{(c_\ell, k_\ell) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}_2^{(\ell)})\}_{\ell \in [K] \setminus \{u\}} \\ (c_u, k_u) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)}) \\ k^0 \xleftarrow{\$} \bigoplus_{\ell=1}^{K} k_\ell \\ k^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \end{array} \right] \\
&= \Pr\left[D(k, ((c_1, \ldots, c_K), u)) = b \colon \begin{array}{c} \{(c_\ell, k_\ell) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}_2^{(\ell)})\}_{\ell \in [K] \setminus \{u\}} \\ (c_u, k_u) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)}) \\ k_u^0 \leftarrow k_u, k_u^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \\ k \leftarrow k_u^b \oplus \bigoplus_{\ell \in [K] \setminus \{u\}} k_\ell \end{array} \right].
\end{aligned}$$

The above equality holds. To see this, observe that for $b = 0$, both experiments are identical. When $b = 1$, we have $k_u^b = k_1^1 \xleftarrow{\$} \mathcal{K}$, and thus the two distributions are are identical:

$$\left(k \leftarrow k_u^b \oplus \bigoplus_{\ell \in [K] \setminus \{u\}} k_\ell\right) \equiv \left(k \xleftarrow{\$} \mathcal{K}\right).$$

Next, let us instantiate $P_\Pi(D_u)$. We get that

$$\begin{aligned}
&P_\Pi(D_u) \\
&= \Pr\left[D_u\left(k_u^b, c_u\right) = b\colon \begin{array}{c}(c_u, k_u^0) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)}) \\ k_j^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}\end{array}\right] \\
&= \Pr\left[D(k, (c_1, \ldots, c_u, \ldots, c_K)) = b\colon \begin{array}{c}\{(c_\ell, k_\ell) \xleftarrow{\$} \mathsf{TE.Enc}(\mathsf{pk}_2^{(\ell)})\}_{\ell \in [K] \setminus \{u\}} \\ (c_u, k_u^0) \xleftarrow{\$} \Pi_{\mathsf{t\text{-}tr}}.\mathsf{Enc}(\mathsf{pk}_1^{(u)}) \\ k_u^1 \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\} \\ k \leftarrow k_u^b \oplus \bigoplus_{\ell \in [K] \setminus \{u\}} k_\ell\end{array}\right] \\
&= P_{u,\Pi'}(D),
\end{aligned}$$

where in the second step, we inlined the construction of D_u with respect to D as defined by $\Pi_{\mathsf{t\text{-}tr}}.\mathsf{Trace}$. ∎

Claim D.10 If D_j is ε'-good, then $\mathsf{test}_j = 0$ can happen with at most negligible probability.

Proof of Claim D.10. Suppose D_j is ε'-good. Let the random variable X count the number of successful tests on D_j and let N denote the number of trials. Further, we can bound the mean $\mu = N \cdot P_\Pi(D_j) \geq N \cdot (1/2 + \varepsilon')$ We are interested in

$$\Pr[\mathsf{test} = 0 | D_j \text{ is } \varepsilon'\text{-good}] = \Pr[X < N \cdot (1/2 + \varepsilon'/2)],$$

and applying the Hoeffding inequality with $t = N\varepsilon'/2$, we get that

$$\Pr[\mathsf{test} = 0 | D_j \text{ is } \varepsilon'\text{-good}] \leq \Pr[X \leq \mu - t] \leq \exp\left(-\frac{N\varepsilon'^2}{2}\right),$$

which is negligible for $N \geq 2\lambda/\varepsilon'^2$. Since we set $Q = 8\lambda/\varepsilon'^2 \geq 2\lambda/\varepsilon'^2$ in the construction, the probability is negligible. ∎

Proof of Theorem 7.1. Intuitively, our proof works as follows: We will show that in all bins B_j such that D_j is at least $\varepsilon'/4$-good, tracing will succeed, and we will find at least t_{bin} traitors from each one of these bins due to t_{bin}-traceability of $\Pi_{\mathsf{t\text{-}tr}}$. Then, as this is true for at least $(1-\gamma)K$ bins (Lemma 7.2), we get at least $(1-\gamma)K \cdot t_{\mathsf{bin}}$ traitors.

To formalize this intuition, we follow proof of Theorem 6.3, with minor modifications to reflect the new construction. We include the details for completeness.

First, we define the following two events for all $j \in [K]$:

- $\mathsf{GoodTrBin}_j$: This event occurs whenever both $|\mathcal{T}_j'| \geq t_{\mathsf{bin}}$ and $\mathcal{T}_j' \subseteq \mathcal{T}_j$.
- $\mathsf{BadTrBin}_j$: This event occurs whenever $\mathcal{T}_j' \not\subseteq \mathcal{T}_j$.

Let $\mathcal{H}_0$ be the τ-traceability game (Definition 4.1) instantiated with Π' for $\tau = (1-\gamma)Kt_{\mathsf{bin}}$. For $u \in [K]$, let $\mathcal{H}_u$ be the same game instantiated except with a modified winning condition. In $\mathcal{H}_u$, the additional condition triggers if the adversary manages to avoid being traced or to frame an honest party in B_j for some $j \leq u$. More formally, in $\mathcal{H}_u$, we set BadTr to 0, whenever $\mathsf{BadTrBin}_j = 1$ for some $j \leq u$, and we set GoodTr to 1, whenever there exists some $j \leq u$ for

which $\mathsf{GoodTrBin}_j = 0$ and $\mathsf{test}_j = 1$, where test_j is the event that the decoder D_j passes the test described in the tracing procedure.

Now, observe that Π' is naively τ-traceable with respect to $\mathcal{H}_K$. To show this, we prove that for all PPT adversaries $\mathcal{A}$ it holds that $\Pr[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}^{(K)}] - \mathsf{negl}(\lambda)$, and $\Pr[\mathsf{BadTr}_{\mathcal{A}}^{(K)}] = 0$. In particular, let $\mathcal{A}$ be any PPT adversary. If $\mathcal{A}$ hits any event $\mathsf{BadTrBin}_j$ for $j \in [K]$, then $\mathsf{BadTr}_{\mathcal{A}}^{(K)}$ is set to 0 by definition. If, however, $\mathcal{A}$ avoids triggering this condition, it must cause $\mathcal{T}'_j \subseteq \mathcal{T}_j$ for all $j \in [K]$. As a consequence, the event BadTr cannot occur, because

$$\mathcal{T}' = \bigcup_{j=1}^{K} \mathcal{T}'_j \subseteq \bigcup_{j=1}^{K} \mathcal{T}_j = \mathcal{T}.$$

A similar argument holds for $\mathsf{GoodTr}_{\mathcal{A}}^{(K)}$. Suppose $\mathcal{A}$ avoids hitting the condition, i.e. for all $j \in [K]$ with $\mathsf{test}_j = 1$ it holds that $\mathsf{GoodTrBin}_j = 1$. Then, $\mathcal{T}'_j \subseteq \mathcal{T}_j$ for all $j \in [K]$, and because $|\mathcal{T}'_j| \geq t_{\mathsf{bin}}$ for all $j \in [K]$ such that $\mathsf{test}_j = 1$, it holds that

$$|\mathcal{T}'| = \sum_{j=1}^{K} |\mathcal{T}'_j| \geq \sum_{j \in [K] \text{ s.t. } \mathsf{test}_j = 1} t_{\mathsf{bin}} \geq (1-\gamma)K \cdot t_{\mathsf{bin}} = \tau,$$

which causes $\mathsf{GoodTr}^{(K)}$ to be set to 1. The last inequality follows form Lemma 7.2 and Claim D.10. If, however, $\mathcal{A}$ does trigger the additional condition, then $\mathsf{GoodTr}^{(K)}$ is set to 1.

Hence, in order to show τ-traceability, it remains to show that for all PPT adversaries $\mathcal{A}$, it holds that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}^{(0)}] \geq \Pr[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}] - \mathsf{negl}(\lambda) \text{ and } \Pr[\mathsf{BadTr}_{\mathcal{A}}^{(0)}] \leq \Pr[\mathsf{BadTr}_{\mathcal{A}}^{(K)}] + \mathsf{negl}(\lambda)$$

for a negligible function negl.

Let $\mathcal{A}$ be any PPT adversary. We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ against t_{bin}-traceability of $\Pi_{\mathsf{t\text{-}tr}}$ as follows: First, $\mathcal{B}$ picks an index $u \xleftarrow{\$} [K]$ uniformly at random. For all indices $j \neq u$, $\mathcal{B}$ runs the key generation algorithm of $\Pi_{\mathsf{t\text{-}tr}}$ and TE to generate $\mathsf{pk}_1^{(j)}, \mathsf{pk}_2^{(j)}$, $\{\mathsf{sk}_{1,i}, \mathsf{sk}_{2,i}\}_{i \in B_j}$, and tk_j according to protocol. $\mathcal{B}$ embeds the challenge public key pk^*, as received by $\mathsf{Game\text{-}Tr}$ of $\Pi_{\mathsf{t\text{-}tr}}$, for instance u, i.e., $\mathsf{pk}_1^{(u)} \leftarrow \mathsf{pk}^*$. The reduction also generates $\mathsf{pk}_2^{(u)}, \{\mathsf{sk}_{2,i}\}_{i \in B_u}$. Then, $\mathcal{B}$ runs $\mathcal{A}$, simulating corruption queries for parties in B_u by making use of its own corruption oracle for $\Pi_{\mathsf{t\text{-}tr}}$. Secret for other bins, or for TE, can be simulated without oracle queries, as the key pairs were generated by the reduction.

As a result, $\mathcal{B}$ receives a decoder D. Next, $\mathcal{B}$ constructs the decoder D_u as prescribed by the Trace algorithm of Π'. Then, $\mathcal{B}$ submits D_u to its challenger. As part of this, $\mathcal{B}$ runs the test on decoder D_u. If $\mathsf{test}_u = 0$, then $\mathcal{B}$ outputs a Dummy decoder which always returns 0. Denote with $\mathsf{GoodTr}_{\mathcal{B}}$, $\mathsf{BadTr}_{\mathcal{B}}$, and $\mathsf{GoodDec}_{\mathcal{B}}$ the events GoodTr, BadTr, and $\mathsf{GoodDec}$ for $\mathcal{B}$ when running in the t_{bin}-traceability game of $\Pi_{\mathsf{t\text{-}tr}}$.

Before we bound $\mathcal{A}$'s probability to distinguish adjacent hybrids with $\mathcal{B}$'s success probability, we first prove the following intuitive claim.

Claim D.11 For all adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr[\mathsf{test}_{j,\mathcal{A}} = 1] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda).$$

Proof. Suppose that test_j happens and $u = j$. We will show that, in this case, $\mathcal{B}$ produces a good decoder with overwhelming probability or in other words $\Pr[\mathsf{test}_{j,\mathcal{A}} \wedge D_u \text{ is not } \varepsilon'/4\text{-good}] \leq \Pr[\mathsf{test}_{j,\mathcal{A}} = 1 \mid D_u \text{ is not } \varepsilon'/4\text{-good}] \leq \mathsf{negl}(\lambda)$. Denote with X the random variable counting

the successful responses of D in the test. We can apply Lemma D.9 and bound

$$\begin{aligned}
&\Pr\left[\mathsf{test}_j = 1 | P(D_u) < \frac{1}{2} + \frac{\varepsilon'}{4}\right] = \Pr\left[\mathsf{test}_j = 1 \mid P_{u,\Pi'}(D) < \frac{1}{2} + \frac{\varepsilon'}{4}\right] \\
&\leq \Pr\left[\mathsf{test}_j = 1 \mid P_{u,\Pi'}(D) = \frac{1}{2} + \frac{\varepsilon'}{4}\right] \\
&= \Pr\left[X \geq N \cdot \frac{1}{2} + \frac{\varepsilon'}{2} \mid P_{u,\Pi'}(D) = \frac{1}{2} + \frac{\varepsilon'}{4}\right],
\end{aligned}$$

where N denotes the number of repetitions in the test. Observe that, if $P(D_u) < 1/2 + \varepsilon'/4$, then $\mathbb{E}[X] \leq \mu \coloneqq N \cdot (1/2 + \varepsilon'/4)$. Let $t = N\varepsilon'/4$. We can apply Hoeffding's inequality and bound

$$\Pr\left[\mathsf{test}_j = 1 | P(D_u) < \frac{1}{2} + \frac{\varepsilon'}{4}\right] \leq \Pr[X \geq \mu + t] \leq \exp\left(-\frac{N\varepsilon'^2}{8}\right),$$

which is negligible for $N \geq 8\lambda/\varepsilon'^2$. Since we set $Q \coloneqq 8\lambda/\varepsilon'^2$ in the construction, the claim holds. ■

Equipped with Claim D.11, we next bound $\mathcal{A}$'s advantage in distinguishing subsequent hybrids with respect to GoodTr and BadTr.

Claim D.12 For all $j \in [K]$ there exists a negligible function negl such that

$$\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda)$$

as well as

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right] = \Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j].$$

Proof of Claim D.12. We start with the latter part of the claim. Observe that the only way for $\mathcal{A}$ to set $\mathsf{BadTr} = 1$ in $\mathcal{H}_{j-1}$ but not in $\mathcal{H}_j$ is to cause $\mathsf{BadTrBin}_j = 1$, i.e., causing the tracing procedure to output some $\mathcal{T}'$ such that $\mathcal{T}'_j \not\subseteq \mathcal{T}_j$. Denote with $\mathcal{T}'_{j,\mathcal{A}}$ and $\mathcal{T}_{j,\mathcal{A}}$ the random variable describing the set of identified traitors in B_j and the set of real traitors from B_j, respectively, when $\mathcal{A}$ runs in the tracing experiment of Π'. Similarly, denote with $\mathcal{T}'_{j,\mathcal{B}}$ and $\mathcal{T}_{j,\mathcal{B}}$ the same variables, when $\mathcal{B}$ runs in the tracing experiment of $\Pi_{\mathsf{t\text{-}tr}}$. Applying the observation above, we can impose the following bound:

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right] = \Pr\left[\mathcal{T}'_{j,\mathcal{A}} \not\subseteq \mathcal{T}_{j,\mathcal{A}}\right].$$

By construction of $\Pi'.\mathsf{Trace}$ and $\mathcal{B}$, it must also hold that

$$\Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] = \Pr\left[\mathcal{T}'_{j,\mathcal{B}} \not\subseteq \mathcal{T}_{j,\mathcal{B}} \mid \mathcal{B}.u = j\right] = \Pr\left[\mathcal{T}'_{j,\mathcal{A}} \not\subseteq \mathcal{T}_{j,\mathcal{A}}\right],$$

and thus

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j-1)}\right] - \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(j)}\right] = \Pr[\mathsf{BadTr}_{\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda).$$

Next, we apply a similar argument to the first part of the claim. Again, the only way that $\mathsf{GoodTr}_{\mathcal{A}}^{(j)} = 1$, but $\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)} = 0$ is if $\mathcal{A}$ has hit the modified condition in $\mathcal{H}_j$, i.e., $\mathsf{GoodTrBin}_j = 0$ and $\mathsf{test}_j = 1$.

$$\begin{aligned}
&\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right] \\
&\leq \Pr[\neg\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge \mathsf{test}_j = 1] \\
&= \Pr[\mathsf{test}_j = 1] - \Pr[\mathsf{test}_j = 1 \wedge \mathsf{GoodTrBin}_{j,\mathcal{A}}] \\
&= \Pr[\mathsf{test}_j = 1] - (\Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}}] - \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge \mathsf{test}_j = 0]) \\
&= \Pr[\mathsf{test}_j = 1] - \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}}] + \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}} \wedge \mathsf{test}_j = 0] \\
&\leq \Pr[\mathsf{test}_j = 1] - \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}}] + \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}} \mid \mathsf{test}_j = 0] \\
&\leq \Pr[\mathsf{test}_j] - \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}}] + 0
\end{aligned}$$

where pk and tk are treated as random variables. Hence, there exists a negligible function negl such that

$$\Pr\left[\Pi.\mathsf{Trace}^{\mathsf{Dummy}(\cdot)}(\mathsf{pk},\mathsf{tk},\varepsilon'/4) \neq \emptyset\right] \leq \mathsf{negl}(\lambda).$$

Applying additionally Claim D.11 to substitute $\Pr[\mathsf{test}_j]$, and simplifying negligibilities yields the following:

$$\begin{aligned}
&\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j)}\right] - \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(j-1)}\right] \\
&\leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr[\mathsf{GoodTrBin}_{j,\mathcal{A}}] + \mathsf{negl}(\lambda) \\
&= \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr\left[\mathcal{T}'_{\mathcal{B}} \subseteq \mathcal{T}_{\mathcal{B}} \wedge |\mathcal{T}'_{\mathcal{B}}| \geq t_{\mathsf{bin}} \mid \mathcal{B}.u = j\right] + \mathsf{negl}(\lambda) \\
&= \Pr[\mathsf{GoodDec}_{\mathcal{B}} | \mathcal{B}.u = j] - \Pr[\mathsf{GoodTr}_{j,\mathcal{B}} \mid \mathcal{B}.u = j] + \mathsf{negl}(\lambda).
\end{aligned}$$

■

We next continue as in the proof of Theorem 6.3 and get:

$$\begin{aligned}
\Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(0)}\right] &\geq \Pr\left[\mathsf{GoodTr}_{\mathcal{A}}^{(K)}\right] - K \cdot \mathsf{negl}(\lambda) \\
&\geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - K \cdot \mathsf{negl}(\lambda),
\end{aligned}$$

and

$$\Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(0)}\right] \leq \Pr\left[\mathsf{BadTr}_{\mathcal{A}}^{(K)}\right] + K \cdot \mathsf{negl}(\lambda) = K \cdot \mathsf{negl}(\lambda).$$

■

D.5 Correctness and Semantic Security of Construction 6

In this section, we prove probabilistic correctness, and semantic security of Construction 6 as stated in Theorem 8.1.

We first recall the parameters of Theorem 8.1. Let $a \in (0,1)$ and $c_1, c_2 \in (0, 1/2)$, let $\delta_1 = 6n^{-a/2}$ and $\delta_2 = 6\sqrt{n}/(n - n^a)$, and thus $t_1 = (1-\delta_1)\frac{n^a}{2}$ and $t_2 = (1-\delta_2)\frac{n-n^a}{2}$. Let

$$\ell \;=\; \max\left\{3\lambda n^{1-a},\; \frac{10n^{1-a}(1-c_1)}{c_1^2},\; \frac{20n^{1-a}(2-3c_2)}{c_2^2}\right\} \text{ and } t_{\mathsf{parts}} = 0.8\ell.$$

Let Π be a perfect $\mathsf{TT\text{-}KEM}$ that is 1-traceable against $t_1 - 1$ partial insiders and TE be a perfect $\mathsf{T\text{-}KEM}$. Let Π' be instantiated as in Construction 6 with the parameters above. Then, the following lemmas apply.

Lemma D.13 (Correctness of Construction 6). *Π' is probabilistically correct for $\mathcal{S}_n = \{n/2\}$.*

Proof of Lemma D.13. The lemma follows directly from the first part of Lemma 8.2. ■

Lemma D.14 (Semantic Security of Construction 6). *Construction 6 is semantically secure for $t = t_1 + t_2 = n/2 - o(n)$.*

Proof Sketch of Lemma D.14. Suppose an adversary corrupts less than $t = t_1 + t_2$ parties. As a result, to decrypt in any given partition, it must either decrypt break Π in the first bin while corrupting less than t_1 parties, or TE in the second bin, while corrupting less than t_2 parties. Then, even disregarding the fact that the adversary must achieve the above in at least t_{parts} out of the ℓ partitions, we can loosely bound any PPT adversary $\mathcal{A}$'s advantage against semantic security of Π' by

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Pi'} \leq 2\ell\left(\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_1,\Pi} + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{B}_2,\mathsf{TE}}\right)$$

for appropriate PPT reductions $\mathcal{B}_1$ and $\mathcal{B}_2$. ■

E Naive t-Traceable Construction with Large Ciphertext

The construction was outlined in [20], but without a formal security proof. In this section, we provide such a proof.

A naive t-traceable threshold encryption scheme is as follows: each party samples its own key pair $(\mathsf{sk}_i, \mathsf{pk}_i)$, and encryption proceeds by t-out-of-n secret-sharing the message m, with the i-th share encrypted under pk_i. Then, Traceability follows from PLBE.

As a starting point, we assume that the naive PLBE-scheme is 1-traceable, and make two straightforward observations about its tracing algorithm:

- We can carry out tracing following an arbitrary order of revocation, i.e., we assume that $\Pi_{\mathsf{naive}}.\mathsf{Trace}$ takes an additional input Ord, where $\mathsf{Ord} = (i_1, i_2, \ldots, i_n)$ is an ordering of the set of parties $[n]$.
- We obsere that, when instantiated with threshold t, for all decoders D, all public keys pk, all $\varepsilon \in (0, 1/2)$, and all orderings Ord, the algorithm $\Pi_{\mathsf{naive}}.\mathsf{Trace}^D(\mathsf{pk}, \varepsilon, \mathsf{Ord})$, either returns an empty set $\emptyset$, or a singleton $\{p\}$, where $p \in \mathsf{Ord}[\leq n - t + 1]$.

In summary, it holds that $\Pi_{\mathsf{naive}}.\mathsf{Trace}^D(\mathsf{pk}, \varepsilon, \mathsf{Ord}) \in \mathsf{Ord}[\leq n - t + 1]$.

We exploit this property to find t traitors by tracing D a total of t times. We start with an arbitrary initial ordering Ord_1, and in every step $i \in [t]$, we construct Ord_{i+1} by swapping the traitor p_i that was found in step i into the trailing $t - 1$ positions. More formally, we construct $\Pi^{\mathsf{t\text{-}trace}}_{\mathsf{naive}}$ as follows

Construction 11 (Naive t-traceable TT-KEM)**.** Let Π_{naive} be the naive PLBE scheme. Then $\Pi^{\mathsf{t\text{-}trace}}_{\mathsf{naive}}$ is identical to Π_{naive} except for the tracing algorithm.

$\Pi^{\mathsf{t\text{-}trace}}_{\mathsf{naive}}.\mathsf{Trace}^D(\mathsf{pk}, \varepsilon)$**:** First, test the decoder D as $\mathsf{test} \xleftarrow{\$} \mathsf{TestDec}(\mathsf{pk}, \ldots, Q := 8\lambda/\varepsilon^2, \varepsilon/2)$. If $\mathsf{test} = 0$, output an empty set. Otherwise, let $\mathsf{Ord}_1 = (1, 2, \ldots, n)$. Initialize $\mathcal{T}' \leftarrow \emptyset$, and run $i \in [t]$ steps:

- Compute $\mathcal{T}'_i \xleftarrow{\$} \Pi_{\mathsf{naive}}.\mathsf{Trace}^D(\mathsf{pk}, \varepsilon/4, \mathsf{Ord}_i)$. If $\mathcal{T}'_i = \emptyset$, return $\mathcal{T}'$.
- Else, it must hold that $\mathcal{T}'_i = \{p_i\}$ for some $p_i \in \mathsf{Ord}_i[\leq n - t + 1]$. Let pos_i be p_i's position in Ord_i, i.e., $p_i = \mathsf{Ord}_i[\mathsf{pos}_i]$.
- Initialize $\mathsf{Ord}_{i+1} \leftarrow \mathsf{Ord}_i$, and then swap p_i with the party at position $n - i + 1$

$$\mathsf{Ord}_{i+1}[\mathsf{pos}_i] \leftarrow \mathsf{Ord}_i[n - i + 1] \text{ and } \mathsf{Ord}_{i+1}[n - i + 1] \leftarrow p_i.$$

- Update $\mathcal{T}' \leftarrow \mathcal{T}' \cup \mathcal{T}'_i$.

Finally, after executing t steps, output $\mathcal{T}'$.

Theorem E.1 *There exists a t-traceable threshold encryption scheme with threshold t whose ciphertext and public key sizes are $O(n)$, and whose secret share size is $O(1)$.*

Proof. The theorem follows by instantiating $\Pi^{\mathsf{t\text{-}trace}}_{\mathsf{naive}}$ with Π_{naive}, which, as discussed in [20], satisfies the requirement that the revocation order is not fixed in advance. We then show that the tracing algorithm described above identifies t traitors. This follows from the argument that each step $i \in [t]$, the tracing procedure outputs a distinct traitor. The proof is a simplified version of the argument given in Section 8; we include it here for completeness.

Let $\mathsf{Cut} = n - t + 1$, and recall the invariant that $\mathcal{T}'_i \subseteq \mathsf{Ord}_i[\leq \mathsf{Cut}]$. We show that in each step $i \in [t]$, a traitor from $\mathsf{Ord}_i[\leq \mathsf{Cut}]$ is found, and that honest parties cannot be framed in any of the steps. To formalize this intuition, we first define what it means that tracing *succeeds* in a step i. Denote for steps $i \in [t]$ with $\mathsf{GoodStep}_i$ the following event:

$$\mathsf{GoodStep}_i = \begin{cases} 1 & \text{if } |\mathcal{T}'_i| > 0 \text{ and } \mathcal{T}'_i \subseteq \mathcal{T} \\ 0 & \text{otherwise.} \end{cases}$$

Denote with $\mathsf{GoodSteps}$ the event that $\mathsf{GoodStep}_i = 1$ for all $i \in [t]$. Also, denote with $\mathsf{BadStep}_i$ the event that any non-traitor is framed as traitor in step i, and with $\mathsf{BadSteps}$ the event that there exists an $i \in [t]$ such that $\mathsf{BadStep}_i = 1$.

Reduction $\mathcal{B}$ to 1-traceability of Π_{naive}. Let $\mathcal{A}$ be an adversary against the t-traceability of $\Pi_{\mathsf{naive}}^{\mathsf{t\text{-}trace}}$. We construct a reduction $\mathcal{B}$ that initially samples a random guess for a step $i^* \xleftarrow{\$} [t]$. The reduction runs against 1-traceability of Π_{naive} and simulates t-traceability of $\Pi_{\mathsf{naive}}^{\mathsf{t\text{-}trace}}$ to $\mathcal{A}$ by embedding the challenge public key. When $\mathcal{A}$ issues a corruption query for party p, then $\mathcal{B}$ queries its own corruption oracle on p to learn sk_p and forwards the result to $\mathcal{A}$. In particular, this means that $\mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$.

When $\mathcal{A}$ outputs a decoder D, $\mathcal{B}$ runs the tracing procedure up to step i^*. This is possible since the tracing is public (i.e., no tracing key is required). Thus, $\mathcal{B}$ simulates Ord_{i^*}, and returns the tuple (D, Ord_{i^*})[22] to its challenger.

Observe that $\mathcal{A}$'s probability of triggering $\mathsf{BadSteps}$ is independent of $\mathcal{B}$'s choice of i^*, and thus whenever $\mathcal{A}$ triggers $\mathsf{BadSteps}$, there must exist one $i \in [t]$ such that tracing in the i-th step outputs at least one non-traitor ($\mathcal{T}_i' \not\subseteq \mathcal{T}$). Hence, $\mathcal{B}$ guesses i correctly with probability at least $1/t$. Denote by $\mathcal{T}_{\mathcal{A}}'(i^*)$ the traitors found in step i^* from $\mathcal{T}_{\mathcal{A}}$. If $\mathcal{B}$ guesses correctly, then $\mathcal{T}_{\mathcal{B}}' = \mathcal{T}_{\mathcal{A}}'(i^*) \not\subseteq \mathcal{T}_{\mathcal{A}} = \mathcal{T}_{\mathcal{B}}$, and thus $\Pr[\mathsf{BadSteps}_{\mathcal{A}}] \leq t \cdot \Pr[\mathsf{BadTr}_{\mathcal{B}}]$.

Proving $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$. Next, we show that for any PPT adversary $\mathcal{A}$ it holds that $\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda)$. We define a series of hybrids $\mathcal{H}_i$ for $i \in [0, t]$ such that in $\mathcal{H}_0$, all events are defined as is, i.e., $\mathcal{H}_0$ is identical to the real game. In $\mathcal{H}_i$, however, we modify the definition of events as follows: If for any $u \leq i$ it holds that $\mathsf{GoodStep}_u = 0$, then we overwrite $\mathsf{GoodSteps} \leftarrow \mathsf{test}$, i.e., if the adversary manages that in any step $u \leq i$ either no traitors is found or non-traitors are framed, then we just set $\mathsf{GoodSteps}$ to 1, iff the decoder D tests good.

Denote with $\mathsf{GoodSteps}^{(i)}$ the random variable representing the event $\mathsf{GoodSteps}$ in hybrid $\mathcal{H}_i$. First, observe that in the last hybrid $\mathcal{H}_t$ it trivially holds that for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t)}] \geq \Pr[\mathsf{test}_{\mathcal{A}}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda),$$

since if a decoder is good, it will only test bad with negligible probability.[23] It remains to show that for all PPT adversaries there exists a negligible function negl such that

$$\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}] \geq \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t)}] - \mathsf{negl}(\lambda). \tag{5}$$

To prove this, we argue that for all subsequent hybrids, it holds that

$$\Pr\left[\mathsf{GoodSteps}_{\mathcal{A}}^{(i)}\right] - \Pr\left[\mathsf{GoodSteps}_{\mathcal{A}}^{(i-1)}\right] \leq \Pr\left[\mathsf{test}_{\mathcal{A}}\right] - \Pr\left[\mathsf{GoodStep}_{i,\mathcal{A}}\right] + \mathsf{negl}(\lambda).$$

To see this, observe that the only way for $\mathcal{A}$ to cause $\mathsf{GoodSteps}_{\mathcal{A}}^{(i)} = 1$, but $\mathsf{GoodSteps}_{\mathcal{A}}^{(i-1)} = 0$ is to set $\mathsf{GoodStep}_i = 0$, and $\mathsf{test} = 1$, and thus we obtain the following bound.

$$\begin{aligned}
&\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i)}] - \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i-1)}] \\
&\leq \Pr[\neg\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}}] = \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \wedge \mathsf{test}_{\mathcal{A}}] \\
&\leq \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + \Pr[\mathsf{GoodStep}_{i,\mathcal{A}} \mid \mathsf{test}_{\mathcal{A}} = 0] \\
&\leq \Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] + 0
\end{aligned}$$

[22] Recall that we use the observation here, that in Π_{naive}, the revocation ordering can be adversarially chosen at the time of tracing

[23] This follows a Hoeffding inequality identically to that of Claim D.2.

Next, we need to show that

$$\Pr[\mathsf{test}_{\mathcal{A}}] - \Pr[\mathsf{GoodStep}_{i,\mathcal{A}}] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i]) + \mathsf{negl}(\lambda). \quad (6)$$

This follows from the following two arguments. First, if $\mathcal{B}$ picks $i^* = i$, then $\mathsf{GoodTr}_{\mathcal{B}}$ implies $\mathsf{GoodStep}_{i,\mathcal{A}}$. Second, note that if $\mathsf{test}_{\mathcal{A}}$ happens, then D must be at least $\varepsilon/4$ good decoder except with negligible probability. Hence, since $\mathcal{B}$ returns the same decoder D, then it must hold that $\mathsf{GoodDec}_{\mathcal{B}}$ happens except with negligible probability, and thus

$$\Pr[\mathsf{test}_{\mathcal{A}}] \leq \Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] + \mathsf{negl}(\lambda).$$

Therefore,

$$\begin{aligned}
&\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t)}] - \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}] \\
&= \sum_{i=1}^{t} \left(\Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i)}] - \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(i-1)}]\right) \\
&\leq \sum_{i=1}^{t} (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \mid \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \mid \mathcal{B}.i^* = i]) + \mathsf{negl}'(\lambda) \\
&= \sum_{i=1}^{t} \frac{\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i]}{\Pr[\mathcal{B}.i^* = i]} + t \cdot \mathsf{negl}'(\lambda) \\
&= t \cdot \sum_{i=1}^{t} (\Pr[\mathsf{GoodDec}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i] - \Pr[\mathsf{GoodTr}_{\mathcal{B}} \wedge \mathcal{B}.i^* = i]) + t \cdot \mathsf{negl}'(\lambda) \\
&= t \cdot (\Pr[\mathsf{GoodDec}_{\mathcal{B}}] - \Pr[\mathsf{GoodTr}_{\mathcal{B}}]) + t \cdot \mathsf{negl}'(\lambda) \\
&\leq t \cdot \mathsf{negl}(\lambda) + t \cdot \mathsf{negl}'(\lambda).
\end{aligned}$$

The last inequality follows from the 1-traceability of Π_{naive}. Hence,

$$\Pr[\mathsf{GoodSteps}_{\mathcal{A}}] = \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(0)}] \geq \Pr[\mathsf{GoodSteps}_{\mathcal{A}}^{(t)}] - \mathsf{negl}(\lambda) \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}'(\lambda).$$

Taking into account our invariant that $\mathcal{T}_i' \subseteq \mathsf{Ord}_i$, we can conclude that, if $\mathsf{GoodSteps}$ happens, i.e., in every step, one unique traitor is found, then we must have found t distinct real traitors. Thus, we get that

$$\Pr[\mathsf{GoodTr}_{\mathcal{A}}] = \Pr[\mathsf{GoodSteps}] \geq \Pr[\mathsf{GoodDec}_{\mathcal{A}}] - \mathsf{negl}(\lambda).$$

■

F A Proof of Theorem 9.6

Lemma F.1 (Left-SPD Security of $\Sigma_{\mathsf{PD\text{-}BT}}$). *The* PD-BT-KEM $\Sigma_{\mathsf{PD\text{-}BT}}$ *is Left-SPD secure under the augmented bilinear decisional Diffie-Hellman assumption [11, Definition 8].*

Proof of Lemma F.1. Our Game-OSS-Left instantiated with $\Sigma_{\mathsf{PD\text{-}BT}}$ is identical to Game-OSS instantiated with the construction in [11, Section 5.2] for $d = 0$ except for the additional public key elements $\{X_i\}_{i\in[n]}$ and the simulated partial decryptions for the challenge ciphertext. The proof is almost identical to that in [11] for $d = 0$. In particular, observe that their reduction samples α as well as a sharing $s_1, \ldots, s_n$ of α itself. Thus, we can simulate the additional outputs without any additional assumptions as follows:

Simulating $\{X_i\}_{i\in[n]}$. Since the reduction knows the field elements s_i for the sharing of α, it can simulate X_i as $e(g_1^{yz}, g_2)^{s_i} = e(g_1, g_2)^{s_i yz}$ for all $i \in [n]$, where g_1^{yz} is already part of the reductions ABDDH challenge.

Simulating $\mathsf{SimPD}(\mathsf{pk}, i, c^*, j, r^*_{\mathsf{Enc}})$**.** The challenge here is that the reduction does not know the field element r corresponding to the challenge r^*_{Enc}. However, the reduction learns g_1^{yr} and g_2^z as part of its ABDDH challenge and thus can compute $d_i \leftarrow e(g_1^{yr}, g_2^z)^{s_i}$ on behalf of all $i \in \mathcal{I}_{\mathsf{pd}}$, which is consistent with the challenge ciphertext.

■

Game $\mathsf{Int\text{-}ABCDH}_{\mathcal{A},t}(1^\lambda)$

$E \coloneqq (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, p) \xleftarrow{\$} \mathcal{G}(1^\lambda)$

$S \leftarrow \{0, 1, \ldots, t-1\}$

$\forall k \in S$: Sample $\alpha_k \xleftarrow{\$} \mathbb{Z}_p$

$h_0, h_1, y, z, r, v \xleftarrow{\$} \mathbb{Z}_p$

$\mathsf{view}_1 \leftarrow \big(g_1^y, g_1^z, g_1^{yr+h_0 v}, g_1^{h_0}, g_1^{h_1}, g_1^{\alpha_0 yz}$
$\quad \{g_1^{\alpha_k}\}_{k\in S}, \{g_1^{h_0 y\alpha_k}\}_{k\in S}, \{g_1^{h_1 y\alpha_k}\}_{k\in S}, \{g_1^{h_1 z\alpha_k}\}_{k\in S}\big)$

$\mathsf{view}_2 \leftarrow \big(g_2^v, \{g_2^{y\alpha_k}\}_{k\in S}, \{g_2^{z\alpha_k}\}_{k\in S}\big)$

$(\mathcal{I}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, E, \mathsf{view}_1, \mathsf{view}_2)$

if $\mathcal{I} \not\subseteq \mathbb{N} \setminus \{0\} \vee |\mathcal{I}| \geq t$ **then return** 0

$\mathsf{view}_T \leftarrow \left\{ e(g_1, g_2)^{yzr \sum_{k=0}^{t-1} \lambda_{k,S}^{(i)} \cdot \alpha_k} \right\}_{i\in\mathcal{I}}$

$K \xleftarrow{\$} \mathcal{A}(\mathsf{st}, \mathsf{view}_T)$

if $K = e(g_1, g_2)^{\alpha_0 yzr}$ **then return** 1

else return 0

Figure 13: The $\mathsf{Int\text{-}ABCDH}$ experiment.

Assumption 1 (Interactive t-Augmented Bilinear Computational Diffie-Hellman (t-ABCDH)). We say that the interactive t-ABCDH problem is hard with respect to a bilinear group generation algorithm $\mathcal{G}$ and $t \in \mathbb{N}$ if for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that for all $\lambda \in \mathbb{N}$

$$\Pr[\mathsf{Int\text{-}ABCDH}_{\mathcal{A},t}(1^\lambda) = 1] \leq \mathsf{negl}(\lambda),$$

where the $\mathsf{Int\text{-}ABCDH}$ experiment is defined in Figure 13.

We prove that the t-ABCDH assumption holds in the GGM in Appendix G.

Lemma F.2 (Right-SPD Security of $\Sigma_{\mathsf{PD\text{-}BT}}$). *The* $\mathsf{PD\text{-}BT\text{-}KEM}$ $\Sigma_{\mathsf{PD\text{-}BT}}$ *is Right-SPD secure under the t-ABCDH assumption (Assumption 1).*

Proof of Lemma F.2. Let $\lambda, n, t \in \mathbb{N}$ and $0 < t \leq n$. Let $\mathcal{A}$ be a PPT adversary against $\mathsf{Game\text{-}OSS\text{-}Right}$ (Figure 8) instantiated with $\Sigma_{\mathsf{PD\text{-}BT}}$. Let $L \in \mathsf{poly}(\lambda)$ be an upper bound on the number of positions ℓ in $\mathsf{PD\text{-}BT\text{-}KEM}$. We construct a PPT reduction $\mathcal{B}$ that uses $\mathcal{A}$ to win the $\mathsf{Int\text{-}ABCDH}_t$ experiment (Figure 13). In the beginning, $\mathcal{B}$ receives the pairing ensemble $E = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, p)$ as well as the following challenge group elements:

$$\mathsf{view}_1 = (g_1^y, g_1^z, g_1^{yr+h_0 v}, g_1^{h_0}, g_1^{h_1}, g_1^{\alpha_0 yz}, \{g_1^{\alpha_k}\}_{k\in S}, \{g_1^{h_0 y\alpha_k}\}_{k\in S}, \{g_1^{h_1 y\alpha_k}\}_{k\in S}, \{g_1^{h_1 z\alpha_k}\}_{k\in S})$$

$$\mathsf{view}_2 = \left(g_2^v, \{g_2^{y\alpha_k}\}_{k\in S}, \{g_2^{z\alpha_k}\}_{k\in S}\right),$$

where $S = [0, t-1]$. Intuitively, $\mathcal{B}$ embeds the challenge into its simulation of $\mathsf{Game\text{-}OSS\text{-}Right}$ such that $\alpha \coloneqq \alpha_0$ and $s_i \coloneqq \sum_{k=0}^{t-1} \lambda_{k,S}^{(i)} \alpha_k$. This will cause all s_i to lie on a degree $t-1$ polynomial f with $f(k) = \alpha_k$ for $k \in S$. First, $\mathcal{B}$ samples a random $j^* \in [L]$ and runs $\mathcal{A}_1(1^\lambda)$, while simulating the random oracle $H(j)$ as follows:

- If $\mathcal{A}$ queries for $j = j^*$, output $g_1^{h_0}$ from the t-ABCDH challenge.
- If $\mathcal{A}$ queries for $j \neq j^*$, sample a random $w_j \xleftarrow{\$} \mathbb{Z}_p$ and output $(g_1^{h_1})^{w_j}$. Output consistently for subsequent identical queries.

As a result, $\mathcal{B}$ receives (ℓ, u) from $\mathcal{A}$ and aborts if $u \neq j^*$. Otherwise, $\mathcal{B}$ samples w_j for all remaining $j \in [\ell]$ and proceeds to simulate $\mathsf{Game\text{-}OSS\text{-}Right}$ to $\mathcal{A}$ as follows:

Public key pk: $\mathcal{B}$ embeds the challenge $X = g_1^{\alpha_0 yz}$, $Y = g_1^y$, and $Z = g_1^z$. It simulates X_i as $\prod_{k=0}^{t-1} e(g_1^y, g_2^{z\alpha_k})^{\lambda_{k,S}^{(i)}}$ for all $i \in [n]$, which is consistent with $e(g_1, g_2)^{s_i yz}$ for our simulation of s_i. $\mathcal{B}$ assembles $\mathsf{pk} \leftarrow (X, \{X_i\}_{i\in[n]}, Y, Z)$.

Both secret keys for $j \neq u$: For all $j \in [\ell] \setminus \{u\}$ and $i \in [n]$, $\mathcal{B}$ needs to simulate both left and right keys. It simulates

$$\mathsf{sk}_{i,0}^{(j)} \leftarrow \left(\left(\prod_{k=0}^{t-1} (g_1^{h_1 z\alpha_k})^{\lambda_{k,S}^{(i)}}\right)^{w_j}, \prod_{k=0}^{t-1} (g_2^{z\alpha_k})^{\lambda_{k,S}^{(i)}}\right) \left[= \left((g_1^{w_j h_1})^{zs_i}, (g_2^{zs_i})\right) = (H(j)^{zs_i}, g_2^{zs_i})\right]$$

$$\mathsf{sk}_{i,1}^{(j)} \leftarrow \left(\left(\prod_{k=0}^{t-1} (g_1^{h_1 y\alpha_k})^{\lambda_{k,S}^{(i)}}\right)^{w_j}, \prod_{k=0}^{t-1} (g_2^{y\alpha_k})^{\lambda_{k,S}^{(i)}}\right) \left[= \left((g_1^{w_j h_1})^{ys_i}, g_2^{ys_i}\right) = (H(j)^{ys_i}, g_2^{ys_i})\right].$$

Right secret keys for $j = u$: For $j = u$, the reduction is only required to simulate right keys for all parties $i \in [n]$. It does so as follows:

$$\mathsf{sk}_{i,1}^{(u)} \leftarrow \left(\prod_{k=0}^{t-1} (g_1^{h_0 y\alpha_k})^{\lambda_{k,S}^{(i)}}, \prod_{k=0}^{t-1} (g_2^{y\alpha_k})^{\lambda_{k,S}^{(i)}}\right) \left[= \left((g_1^{h_0})^{ys_i}, g_2^{ys_i}\right) = (H(u)^{ys_i}, g_2^{ys_i})\right].$$

Note that the reduction would not be able to simulate left keys for $j = u$, since it does not have access to any $g_1^{h_0 z\alpha_k}$ in the t-ABCDH challenge.

Challenge Ciphertext: Simulate the challenge ciphertext for $r^*_{\mathsf{Enc}} = (r, v, t_1^{(0)}) \equiv (r^{(0)}, t_0^{(0)}, t_1^{(0)})$, where r and v correspond to the respective elements in the ABCDH challenge and $t_1^{(1)}$ is irrelevant (since elements depending on it would only appear in $c_1^{(0)}$, which is outside the adversary's view). More formally, compute $c_0^{(0)} \leftarrow (g_2^v, g_1^{yr+h_0 v})$, which is a valid left ciphertext, since $H(u) = g_1^{h_0}$. Additionally, sample $r', t_1^{(1)} \xleftarrow{\$} \mathbb{Z}_p$ and compute $c_1^{(1)} \leftarrow (g_2^{t_1^{(1)}}, (g_1^z)^{r'} \cdot (g_1^{h_0})^{t_1^{(1)}})$, which corresponds to a random right ciphertext that is inconsistent with the key $K \equiv e(g_1, g_2)^{\alpha_0 yzr}$. Assemble $c^* \leftarrow (c_0^{(0)}, c_1^{(1)})$.

$\mathcal{B}$ assembles $\mathsf{shares} \leftarrow \left(\{(\mathsf{sk}_{i,0}^{(j)}, \mathsf{sk}_{i,1}^{(j)})\}_{i\in[n], j\in[\ell]\setminus\{u\}}, \{\mathsf{sk}_{i,1}^{(u)}\}_{i\in[n]}\right)$ and runs $\mathcal{A}_2(\mathsf{st}, \mathsf{pk}, \mathsf{shares}, c^*)$ to obtain $\mathcal{I}_{\mathsf{pd}}$. Let us assume from now on that $\mathcal{I}_{\mathsf{pd}} \subseteq [n]$ and $|\mathcal{I}_{\mathsf{pd}}| < t$ since $\mathcal{A}$ would otherwise trivially lose $\mathsf{Game\text{-}OSS\text{-}Right}$. Next, the reduction $\mathcal{B}$ forwards $\mathcal{I}_{\mathsf{pd}}$ to its challenger and receives

$$\mathsf{view}_T \leftarrow \left\{Q_i = e(g_1, g_2)^{yzr \sum_{k=0}^{t-1} \lambda_{k,S}^{(i)} \cdot \alpha_k}\right\}_{i\in\mathcal{I}_{\mathsf{pd}}}$$

as a response. Recall that $\mathcal{B}$ simulates such that, conceptually, $s_i = \sum_{k=0}^{t-1} \lambda_{k,S}^{(i)} \cdot \alpha_k$ for all $i \in [n]$, and thus $Q_i = e(g_1, g_2)^{yzr \sum_{k=0}^{t-1} \lambda_{k,S}^{(i)} \cdot \alpha_k} = e(g_1, g_2)^{s_i yzr}$.

Simulating SimPD: $\mathcal{B}$, sets the simulated partial decryptions to Q_i for all $i \in \mathcal{I}_{\mathsf{pd}}$, which are consistent with $e(g_1, g_2)^{s_i yzr}$ for $r^*_{\mathsf{Enc}} = (r, \ldots)$.

$\mathcal{B}$ runs $\mathcal{A}_3(\mathsf{st}, \{Q_i\}_{i\in\mathcal{I}_{\mathsf{pd}}})$ to receive a group element K, which it returns to its own challenger. Note that, if $\mathcal{B}$ initially guesses $j^* = u$ correctly (which $\mathcal{B}$ does with probability $1/L$), then $\mathcal{B}$'s

simulation to $\mathcal{A}$ is perfect. Further, $\mathcal{A}$ wins Game-OSS-Right, whenever $K = e(g_1, g_2)^{\alpha yzr} = e(g_1, g_2)^{\alpha_0 yzr}$ in which case $\mathcal{B}$ wins the Int-ABCDH experiment, and thus

$$\Pr[\mathsf{Game\text{-}OSS\text{-}Right}_{\mathcal{A}}(1^\lambda) = 1] = L \cdot \Pr[\mathsf{Int\text{-}ABCDH}_{\mathcal{B},t}(1^\lambda) = 1],$$

which is negligible as per the interactive t-ABCDH assumption. ∎

The following assumption appears identically in an earlier version of [10].

Assumption 2 (Augmented Bilinear Decisional Diffie-Hellman (ABDDH)). We say that the augmented bilinear decisional Diffie-Hellman (ABDDH) problem is hard with respect to a bilinear group generation algorithm $\mathcal{G}$ if for all PPT adversaries $\mathcal{A}$ the following function

$$\mathsf{Adv}^{\mathrm{abddh}}_{\mathcal{A},\mathcal{G}} := |\Pr[\mathcal{A}(E, \mathsf{view}, e(g_1, g_2)^{\alpha yzr}) = 1] - \Pr[\mathcal{A}(E, \mathsf{view}, e(g_1, g_2)^{v}]|$$

is negligible in $\lambda \in \mathbb{N}$, where $E = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2, p) \xleftarrow{\$} \mathcal{G}(1^\lambda)$, $(\alpha, r, y, z, v) \xleftarrow{\$} \mathbb{Z}_p^5$, and

$$\mathsf{view} \leftarrow (g_1^y, g_1^z, g_1^{yr}, g_1^{zr}, g_1^{\alpha yz}, g_2^y, g_2^z)\,.$$

The semantic security of our construction $\Sigma_{\mathsf{PD\text{-}BT}}$ follows from the same assumption as in [10].[24] Our modification of adding the individual public keys $X_i = e(g_1, g_2)^{s_i yz}$, however, causes a problem when trying to simulate the adaptive corruption oracle because publishing X_i for all parties $i \in [n]$ as part of the public key commits the reduction ahead of time to the shares s_i of α. To avoid this issue, we restrict ourselves to proving semantic security in the *static* setting, where the adversary is required to pick a set of corrupted parties Cor *before* observing the public key. We expect that adaptive semantic security would be provable in an idealized model (such as the GGM [52, 48] or AGM [38]) or with respect to a more tailored interactive assumption that is justifiable in such a model. Importantly, we note that this does not affect traceability, i.e., we can prove traceability even against adversaries who *adaptively corrupt parties*.

Lemma F.3 (Semantic Security of $\Sigma_{\mathsf{PD\text{-}BT}}$). *The* PD-BT-KEM $\Sigma_{\mathsf{PD\text{-}BT}}$ *is semantically secure (Definition 9.5) under ABDDH assumption (Assumption 2) against static corruptions.*

Proof of Lemma F.3. Our proof follows very closely that of [10]. Let $\mathcal{A}$ be an arbitrary PPT adversary against semantic security of PD-BT-KEM. We construct a PPT reduction $\mathcal{B}$ that uses $\mathcal{A}$ to break the ABDDH assumption (Definition 2).

The adversary receives as input an ABDDH tuple

$$(\mathsf{view}, T) = ((g_1^y, g_1^z, g_1^{yr}, g_1^{zr}, g_1^{\alpha yz}, g_2^y, g_2^z)\,, T)\,,$$

where either $T = e(g_1, g_2)^{\alpha yzr}$ or $T = e(g_1, g_2)^v$ for a random $v \in \mathbb{Z}_p$. The reduction runs $\mathcal{A}$ to obtain the number of positions ℓ as well as the set of corrupted parties Cor while simulating the random oracle as follows: On input $j \in \mathbb{N}$, sample $w_j \xleftarrow{\$} \mathbb{Z}_p$ and consistently reply with $g_1^{w_j}$ for this subsequent queries on the same input j. Then, the reduction samples all random oracle values w_j for inputs $j \in [\ell]$ that were not queried by $\mathcal{A}$ so far. W.l.o.g. assume that $\mathsf{Cor} \subseteq [n]$ and $|\mathsf{Cor}| = t - 1$.

Next, $\mathcal{B}$ is tasked with simulating the public key material $\mathsf{pk} = \left(X, \{X_i\}_{i\in[n]}, Y, Z\right)$ as well as the secret key for both sides on behalf of parties in Cor and the challenge ciphertext. To do so, the reduction samples a random $s_i \xleftarrow{\$} \mathbb{Z}_p$ for all $i \in \mathsf{Cor}$ and continues the simulation as follows:

[24] Note that in this paper, we work with a version of $\Sigma_{\mathsf{PD\text{-}BT}}$ that is semantically secure under a variant of *decisional* bilinear Diffie-Hellman, similar to an earlier version of [10] for simplicity. This construction can be transformed to be secure under the *computational* equivalent of this assumption by deriving the final KEM key through a random oracle evaluation on the current KEM key.

Public Key Material: The reduction embeds its ABDDH challenge and sets $X \leftarrow (g_1^{\alpha yz})$, $Y \leftarrow (g_1^y)$, and $Z \leftarrow (g_1^z)$. For all $i \in \mathsf{Cor}$ it sets $X_i \leftarrow e((g_1^y), (g_2^z))^{s_i}$ and for all $j \in [n] \setminus \mathsf{Cor}$ it interpolates

$$X_j \leftarrow X^{\lambda^{(j)}_{0,\mathsf{Cor}\cup\{0\}}} \cdot \prod_{i\in\mathsf{Cor}} X_i^{\lambda^{(j)}_{i,\mathsf{Cor}\cup\{0\}}}.$$

The reduction assembles $\mathsf{pk} \leftarrow \left(X, \{X_i\}_{i\in[n]}, Y, Z\right)$.

Secret Keys: For all corrupted parties $i \in \mathsf{Cor}$, the reduction simulates the left and right secret keys for all positions $j \in [\ell]$ as

$$\mathsf{sk}^{(j)}_{i,0} \leftarrow ((g_1^z)^{w_j s_i}, (g_2^z)^{s_i}).$$

Challenge Ciphertext: To compute the challenge ciphertext, the reduction samples $j^* \xleftarrow{\$} [\ell]$ and $t_0, t_1 \xleftarrow{\$} \mathbb{Z}_p$. It embeds the ABDDH tuple by setting

$$c_0 \leftarrow \left(g_2^{t_0}, (g_1^{yr}) \cdot g_1^{w_{j^*} t_0}\right) \text{ and } c_1 \leftarrow \left(g_2^{t_1}, (g_1^{zt}) \cdot g_1^{w_{j^*} t_1}\right)$$

and assembling $c^* \leftarrow (c_0, c_1)$. The reduction also embeds the ABDDH challenge in the KEM key and sets $k^* \leftarrow T$.

The reduction runs $\mathcal{A}(c^*, k^*, \mathsf{pk}, \{\mathsf{sk}_i\}_{i\in\mathsf{Cor}})$. Note that if the ABDDH challenge $T = e(g_1, g_2)^{\alpha yzr}$, then k^* corresponds to k_1 and $\mathcal{B}$ perfectly simulates $\mathsf{Game\text{-}IND\text{-}BT\text{-}CPA}$ for $b = 1$ to $\mathcal{A}$. If, however, $T = e(g_1, g_2)^v$ for a random $v \in \mathbb{Z}_p$, the k^* corresponds to k_0 and $\mathcal{B}$ perfectly simulates $\mathsf{Game\text{-}IND\text{-}BT\text{-}CPA}$ for $b = 0$ to $\mathcal{A}$. Hence, it holds that

$$\Pr[\mathsf{Game\text{-}IND\text{-}BT\text{-}CPA}_{\mathcal{A}}(1^\lambda) = 1] - \frac{1}{2} \le \mathsf{Adv}^{\mathrm{abddh}}_{\mathcal{B},\mathcal{G}}(1^\lambda),$$

which is negligible by assumption. ∎

G Deferred Proof of Assumption 1

We prove that Assumption 1 holds in the generic group model, making use of the master theorem.

Theorem G.1 (Master Theorem [6, 21]). *Let $p \in \mathbb{N}$ be a prime, let $n, v_1, v_2, v_t \in \mathbb{N}$, let $\mathbf{L}_1 \in \mathbb{F}_p[X_1, \ldots, X_n]^{v_1}$, $\mathbf{L}_2 \in \mathbb{F}_p[X_1, \ldots, X_n]^{v_2}$, $\mathbf{L}_T \in \mathbb{F}_p[X_1, \ldots X_n]^{v_t}$, and $f \in \mathbb{F}_p[X_1, \ldots, X_n]$ be tuples of n-variate polynomials of maximum degree $d_{\mathbf{L}_1}, d_{\mathbf{L}_2}$, $d_{\mathbf{L}_T}$, and d_f respectively. Denote $v = v_1 + v_2 + v_t$ and $d = \max(d_{\mathbf{L}_1} + d_{\mathbf{L}_2}, d_{\mathbf{L}_T}, d_f)$. If f is independent of $(\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_T)$, then for any generic adversary $\mathcal{A}$ that makes at most q group oracle queries:*

$$\left|\Pr\left[\mathcal{A}\begin{pmatrix} p, \mathsf{h}_1[\mathbf{L}_1(x)], \mathsf{h}_2[\mathbf{L}_2(x)], \\ \mathsf{h}_T[\mathbf{L}_T(x), f(x)] \end{pmatrix} = 1\right] - \Pr\left[\mathcal{A}\begin{pmatrix} p, \mathsf{h}_1[\mathbf{L}_1(x)], \mathsf{h}_2[\mathbf{L}_2(x)], \\ \mathsf{h}_T[\mathbf{L}_T(x), r] \end{pmatrix} = 1\right]\right| \le \frac{(q+v+2)^2 \cdot d}{2p},$$

where $\mathsf{h}_1, \mathsf{h}_2$, and h_T denote the corresponding handles, and the probabilities are taken over the choice of $x \xleftarrow{\$} \mathbb{F}_p^n$ and $r \xleftarrow{\$} \mathbb{F}_p$.

Definition G.2 (Independence). Let the completion $\mathbf{C}(\mathbf{L}) = \{\mathbf{L}_1 \otimes \mathbf{L}_2\} \cup \mathbf{L}_T$ of size D for lists $(\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_T)$ be of the form

$$\mathbf{C}(\mathbf{L}) = \{g_1(X_1, \ldots, X_n), \ldots g_D(X_1, \ldots, X_n)\}.$$

We say that an element f is dependent on lists $(\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_T)$, if there exist coefficients $\kappa_i \in \mathbb{F}_p$ such that

$$f(X_1, \ldots, X_n) = \sum_{i=1}^{D} \kappa_i \cdot g_i(X_1, \ldots, X_n).$$

Otherwise, we say that f is independent of $(\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_T)$.

Lemma G.3 *The interactive t-augmented bilinear computational Diffie-Hellman assumption holds in the generic group model.*

Proof. We prove Lemma G.3 by showing an upper bound on the advantage against the decisional version of the t-ABCDH problem in the GGM using the master theorem (Theorem G.1), which trivially implies the same upper bound on the advantage against the computational version. Fix an arbitrary $t \in \mathbb{N}$ and let $\mathcal{A}$ be any PPT adversary playing the $\mathsf{Int\text{-}ABCDH}$ game against the challenger in the GGM. We denote with $H_0, H_1, \{A_k\}_{k\in[0,t-1]}, Y, Z, R, V$ the formal variables corresponding to the field elements $h_0, h_1, \{\alpha_k\}_{k\in[0,t-1]}, y, z, r, v$ that are sampled by the challenger and hidden from the adversary. Instead of receiving group elements directly, the adversary receives corresponding handles and can perform generic operations on the handles using an addition and a pairing oracle. W.l.o.g., assume that the adversary, after receiving handles to view_1 and view_2 as well as the group elements in E, outputs a set $\mathcal{I} \subseteq \mathbb{N}$ with $|\mathcal{I}| < t$ and $0 \notin \mathcal{I}$. As a response, the adversary receives a set of handles to elements in view_T. Denote the set $S = \{0, 1, \ldots, t-1\}$. As a result, we can express the adversary's view as a list of polynomials in our formal variables as follows:

$$\begin{aligned}
\mathbf{L}_1 &= \Big(1, H_0, H_1, Y, Z, \{A_k\}_{k\in S}, \{H_b Y A_k\}_{b\in\{0,1\},k\in S}, \{H_1 Z A_k\}_{k\in S}, A_0 YZ, YR + H_0 V\Big)\\
\mathbf{L}_2 &= \left(1, V, \{YA_k\}_{k\in S}, \{ZA_k\}_{k\in S}\right)\\
\mathbf{L}_T &= \left(\left\{YZR\sum_{k=0}^{t-1}\lambda_k^{(i)}A_k\right\}_{i\in\mathcal{I}}\right)
\end{aligned}$$

with the challenge

$$f = A_0 YZR.$$

To apply the master theorem, we need to show that f is independent of the completion of the above list.

Claim G.4 It holds that f is independent of the completion $\mathbf{C} = \mathbf{C}(\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_T) = \{\mathbf{L}_1 \otimes \mathbf{L}_2\} \cup \mathbf{L}_T$.

Proof of Claim. Let us first list the completion, omitting redundant terms.

$$\begin{aligned}
\mathbf{C}(\mathbf{L}) = \Bigg\{ & 1, H_0, H_1, Y, Z, V, \{A_k\}_{k\in S}, \{H_b YA_k\}_{b\in\{0,1\},k\in S}, \{H_1 ZA_k\}_{k\in S}, A_0 YZ, YR + H_0 V,\\
& \{YA_k\}_{k\in S}, \{ZA_k\}_{k\in S}, \left\{YZR\sum_{k=0}^{t-1}\lambda_k^{(i)}A_k\right\}_{i\in\mathcal{I}}, A_0 YZV, \{A_0 Y^2 ZA_k\}_{k\in S}, \{A_0 YZ^2 A_k\}_{k\in S},\\
& \{H_0 ZA_k\}_{k\in S}, H_0 V, H_1 V, \{YA_k A_\ell\}_{k,\ell\in S}, \{ZA_k A_\ell\}_{k,\ell\in S}, \{A_k V\}_{k\in S},\\
& \{H_0 Y^2 A_k A_\ell\}_{k,\ell\in S}, \{H_0 YZA_k A_\ell\}_{k,\ell\in S}, \{H_0 YA_k V\}_{k\in S},\\
& \{H_1 Y^2 A_k A_\ell\}_{k,\ell\in S}, \{H_1 YZA_k A_\ell\}_{k,\ell\in S}, \{H_1 YA_k V\}_{k\in S},\\
& \{H_1 Z^2 A_k A_\ell\}_{k,\ell\in S}, \{H_1 ZA_k V\}_{k\in S}, \{Y^2 A_k\}_{k\in S}, \{YZA_k\}_{k\in S}, YV, \{Z^2 A_k\}_{k\in S}, ZV,\\
& YRV + H_0 V^2, \{Y^2 RA_k + H_0 YVA_k\}_{k\in S}, \{A_k YZR + A_k H_0 ZV\}_{k\in S}\Bigg\}
\end{aligned}$$

To prove independence, we have to show that f cannot be represented as a linear combination of elements in the completion. First, observe that the only elements in the completion that even contain $A_0 YZR$ as a summand are

$$A_0 YZR + A_0 H_0 ZV \text{ as well as } \left\{YZR\sum_{k=0}^{t-1}\lambda_k^{(i)}A_k\right\}_{i\in\mathcal{I}}.$$

We note that $A_0 YZR + A_0 H_0 ZV$ cannot be symbolically useful in a linear combination for f, because there is no way to cancel out the monomial $A_0 H_0 ZV$, which does not appear elsewhere

in the completion. Further, as there are no other elements in the completion that contain terms of $A_k YZR$ for any $k \in S$, none of the remaining elements can contribute to a linear combination of f in a non-trivial manner. As a result, we argue that the claim of independence is equivalent to whether there exists a linear combination $\beta_1, \ldots, \beta_{t-1} \in \mathbb{Z}_p$ such that

$$A_0 YZR = \sum_{i=1}^{t-1} \beta_i YZR \cdot \left(\sum_{k=0}^{t-1} \lambda_k^{(p_i)} A_k \right),$$

where $\{p_i | i \in [t-1]\} = \mathcal{I}$. Observe that we can divide YZR on both sides such that

$$A_0 = \sum_{i=1}^{t-1} \beta_i \sum_{k=0}^{t-1} \lambda_k^{(p_i)} A_k.$$

We can factor out A_k and rearrange the equation to

$$A_0 = \left(A_0 \cdot \sum_{i=1}^{t-1} \beta_i \lambda_0^{(p_i)} \right) + \sum_{k=1}^{t-1} A_k \cdot \sum_{i=1}^{t-1} \beta_i \lambda_k^{(p_i)}.$$

Note that this is equivalent to the following equation system

$$\underbrace{\begin{pmatrix} \lambda_0^{(p_1)} & \lambda_0^{(p_2)} & \cdots & \lambda_0^{(p_{t-1})} \\ \lambda_1^{(p_1)} & \lambda_1^{(p_2)} & \cdots & \lambda_1^{(p_{t-1})} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{t-1}^{(p_1)} & \lambda_{t-1}^{(p_2)} & \cdots & \lambda_{t-1}^{(p_{t-1})} \end{pmatrix}}_{M} \cdot \underbrace{\begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{t-1} \end{pmatrix}}_{\beta} = \underbrace{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\mathbf{b}},$$

where $M \in \mathbb{Z}_p^{t \times t-1}$, and $\lambda_i^{(p)}$ is the lagrange coefficient for index i to interpolate among set S at position p. Assume towards contradiction that the above equation system $M\beta = \mathbf{b}$ has a solution. We derive a contradiction by constructing a row vector $\mathbf{v} \in \mathbb{Z}_p^t$ such that

$$\mathbf{v} \cdot M = \mathbf{0} \text{ but } \mathbf{v} \cdot \mathbf{b} \neq 0.$$

First, we define a polynomial

$$P(X) = \prod_{i=1}^{t-1} (X - p_i),$$

which is the unique polynomial of degree $t-1$ that has roots at all positions $p_i \in \mathcal{I}$. Observe that, since $0 \notin \mathcal{I}$, it holds that $P(0) \neq 0$. We set

$$\mathbf{v} = (P(0), P(1), \ldots, P(t-1)).$$

Then, for each column j of $\mathbf{v} \cdot M$ we get that

$$\sum_{k=0}^{t-1} \lambda_k^{(p_j)} P(k) = P(p_j) = 0$$

and thus $\mathbf{v} \cdot M = \mathbf{0}$. On the other hand, observe that $\mathbf{v} \cdot \mathbf{b} = P(0) \neq 0$, because $0 \notin \mathcal{I}$ and thus f must be independent of $\mathbf{C}$. ■

Following Claim G.4 we can apply the master theorem (Theorem G.1) to bound $\mathcal{A}$'s advantage in winning the Int-ABCDH experiment in the generic group model by

$$\frac{(q + v + 2)^2 \cdot d}{2p},$$

where $q \in \mathsf{poly}(\lambda)$ is the number of group operation oracle queries made by $\mathcal{A}$, $v = 8 + 7 \cdot t$ is the number of handles in the adversary's view, $d = 5$ is the maximum degree of the elements in the completion $\mathbf{C}$, and p is a λ bit prime. Since $t \in \mathsf{poly}(\lambda)$, we conclude that $\mathcal{A}$'s advantage to break the t-ABCDH assumption is negligible in the GGM. ■

H Full-Insider Attacks in Existing Schemes

In this section, we explain why existing constructions are inherently vulnerable to full-insider attacks. This observation supports our view that achieving full-insider resilience for existing schemes seems out of reach with current techniques without resorting to multi-traitor tracing.

PLBE-based Constructions [20, 23, 28]. Briefly, PLBE includes two encryption methods: Enc and TrEnc. The Enc method is used for standard encryption, while TrEnc produces ciphertexts for tracing. The TrEnc algorithm takes an index $i \in \{0, \ldots, n\}$ as input and encrypts the message such that only parties with index $> i$ can decrypt it. In the threshold setting, this means that only parties $> i$ can issue a valid partial decryption that is useful for decryption. During tracing, the tracer queries the decoder with $\mathsf{TrEnc}(0), \mathsf{TrEnc}(1), \ldots, \mathsf{TrEnc}(n)$ and considers party $i \in [n]$ to be a traitor if the decryption probability drops significantly when i is newly revoked (i.e., when going from $\mathsf{TrEnc}(i-1)$ to $\mathsf{TrEnc}(i)$). Tracing in PLBE relies on three core properties:

- Indistinguishability: Enc and $\mathsf{TrEnc}(0)$ must be indistinguishable except with negligible probability.
- Index hiding: An adversary lacking sk_i can distinguish between $\mathsf{TrEnc}(i-1)$ and $\mathsf{TrEnc}(i)$ only with negligible probability.
- Message privacy: Ciphertexts encrypted via $\mathsf{TrEnc}(n)$ (i.e., all parties are revoked) must be undecryptable except with negligible probability.

These properties enable tracing good decoders because initially, for $\mathsf{TrEnc}(0)$, the decoder's success probability must be as high as for normal decryption due to indistinguishability, but in the end, for $\mathsf{TrEnc}(n)$, it must be negligible due to message privacy. Hence, for at least one i, the success probability must drop significantly. Index hiding ensures it can only drop if the decoder knows sk_i, so the identified party must be a colluder.

Let's now consider a full insider who attaches their secret key to each query. Given the insider's secret key, the index hiding guarantee no longer holds, and the decoder can determine when the insider is revoked during tracing and then refuses decryption for such queries. This trick will cause the tracing algorithm to return the insider. We illustrate the attack through an example.

Example H.1 Let $n = 6$ and $t = 2$, and suppose the colluders that constructed the decoder box D are parties $1, 4, 5$, while the insider is party 3. The tracing algorithm in PLBE begins by querying D with ciphertexts that every party can decrypt. Let ε denote the probability that D correctly decrypts such ciphertexts. Next, the tracing algorithm revokes party 1. The decoder D can detect this revocation. However, since D can still decrypt using the keys of parties 4 and 5, it can continue returning correct answers to these queries. Hence, the decryption probability does not decrease.

The tracer then queries D with ciphertexts in which parties 1 and 2 are revoked. Since party 2 is not a colluder, these ciphertexts are indistinguishable to D from the previous ones, and therefore the decryption probability remains unchanged. Finally, the tracer issues queries in which parties $1, 2, 3$ are revoked. Because the secret share of party 3 is attached to the queries, the decoder can detect that party 3 has been revoked and may therefore respond incorrectly. As a result, the tracer observes a noticeable drop in the decryption probability and identifies party 3, i.e., the insider, as traitor, leaving the original colluders unpunished.

A first attempt to mitigate this is to choose a random revocation order, hidden from the parties and thus potential colluders. Intuitively, in threshold encryption, the tracer can identify a traitor after revoking at most $n - t + 1$. As a result, there is a chance that the insider will never be revoked and therefore will not be identified as a traitor. This technique can be directly applied to some PLBE-based threshold traitor tracing schemes in the literature [20, 28], which would

decrease the success probability of a full-insider attack against such schemes, but not prevent them: For $t = n/2$, the probability that a single full insider is among the first $n - t + 1$ parties of a random but hidden revocation order, and thus can be framed, is $\approx 1/2$. If we consider more than 1 insider, the probability that one of them can be framed is even higher.

So one solution is to enable tracing without fixing the order in the setup, i.e, allowing the tracer to pick an order at the beginning of tracing, depending on the set of insiders specified by the adversary.[25] Note, however, that this is not possible for existing constructions with *succinct* ciphertext, where the order of revocation is fixed during the setup of the scheme [20, 28].

Overcoming this restriction and being able to trace in an arbitrary revocation order seems already sufficient to identify at least t traitors (see Appendix E), suggesting that, barring any new insights, full-insider resilience may be equivalent to multi-traitor tracing, at least for PLBE-based constructions.[26]

Fingerprinting-based Constructions [11, 3, 22]. In these constructions, each party i is assigned a private codeword $w_i \in \{0,1\}^\ell$ of length ℓ drawn from an underlying fingerprinting code. By interacting with the decoder, the tracer can extract a word that is guaranteed to be feasible with respect to the codewords of the colluding parties; i.e., in every position, the bit of the extracted word must be held by at least one of the colluders. The tracing algorithm of the fingerprinting code is then applied to the extracted word to identify one of the colluders responsible for constructing the decoder.

In the presence of a full insider, this approach breaks down. Since the full insider must input its secret key, and thus its private codeword, the decoder can be designed so that the extracted codeword is always feasible for the insider. As a result, the tracing algorithm could always output the full insider, irrespective of the actual colluders.

To build intuition, consider the tracing algorithm in such schemes. Each ciphertext is associated with a position j and consists of two components, left and right: $c = (j, c_0, c_1)$. Parties whose codeword has a 0 (respectively, 1) in position j can decrypt c_0 (respectively, c_1).

In the tracing algorithm, for every position j, the decoder is queried with three types of ciphertexts:

1. ciphertexts where c_1 is invalid (i.e., inconsistent with the plaintext),
2. ciphertexts where both c_0 and c_1 are invalid, and
3. ciphertexts where both c_0 and c_1 are valid.

For each position j, the tracer measures the decoder's success probability on each type. If there is a significant gap between the success probabilities for the first two types, the tracer concludes that at least one colluder has a 0 in position j. This follows because the only difference between these two types is that c_0 is valid in the first case and invalid in the second; distinguishing between them requires that some colluder can decrypt c_0. Similarly, if there is a significant difference between the success probabilities for types 1 and 3, the tracer concludes that at least one colluder has a 1 in position j. Again, the only difference is that c_1 is invalid in type 1 but valid in type 3, which can only be detected if some colluder can decrypt c_1.

Using this information, the tracer constructs a codeword that is *feasible* for the set of colluders. By feasibility, we mean that if the constructed codeword has a bit b in position j, then at least one colluder's codeword also has b in that position. The traceability property of the fingerprinting code guarantees that if the codeword is feasible for the colluders, then at least one colluder is identified.

An Attack in the Presence of an Insider. Now consider the presence of a full insider.

[25] I.e., choose a revocation order such that the first $n - t + 1$ parties do not include any insiders.

[26] One could try running many instances of the scheme with different revocation orders to find K traitors (i.e., $K - 1$ full insider resilience), but this requires $O(2^K)$ instances, which is inefficient for large K.

In this case, the colluders can design the decoder so that the word extracted during tracing always matches the insider's codeword. To see this, consider positions j where the colluders and the insider have mixed bits. In such positions, the decoder can distinguish all three types of ciphertexts. Suppose the insider has a 0 in position j. The decoder can be designed to answer correctly on type 1 ciphertexts and incorrectly on type 2 ciphertexts. According to the tracing rule, this forces the extracted codeword to have a 0 in position j. Similarly, if the insider has a 1 in position j, the decoder can be designed to fail on type 1 ciphertexts and succeed on type 3 ciphertexts. This ensures that the extracted codeword has a 1 in position j.

In positions where all colluders and the insider share the same bit, the extracted codeword will naturally match that bit as well.

Therefore, the extracted codeword will match the insider's codeword exactly. Next, the tracer applies the tracing algorithm of the underlying fingerprinting code to the extracted word. This procedure will return the insider. Indeed, if it returned a different codeword, this would violate the unframeability property of the fingerprinting code: a single holder of a codeword would then be able to embed it in an object such that the tracing algorithm identifies a different holder.

This shows that the colluders can evade tracing by framing the insider.