Traceable Threshold Encryption without Trusted Dealer*

Jan Bormet [1] Jonas Hofmann [1] Hussien Othman [1]

[1] Technical University of Darmstadt, Darmstadt, Germany
{jan.bormet,jonas.hofmann1}@tu-darmstadt.de, hussien.othman@gmail.com

Abstract

The fundamental assumption in t-out-of-n threshold encryption is that the adversary can only corrupt less than t parties. Unfortunately, it may be unfounded in practical scenarios where shareholders could be incentivized to collude. Boneh, Partap, and Rotem (Crypto'24) recently addressed the setting where t or more shareholders work together to decrypt illegally. Inspired by the well-established notion of traitor tracing in broadcast encryption, they added a traceability mechanism that guarantees identifying at least one of the colluders. They provide several constructions that enable traceability, all of which require a trusted dealer to distribute the secret shares. While the trusted dealer can be replaced with a DKG for conventional threshold encryption, it is unclear how to do so without compromising traceability. As thresholdizing is meant to mitigate a single point of failure, a natural question that remains is: Can we construct an efficient traceable threshold encryption scheme that does not rely on a trusted party to distribute the secret shares?

In this paper, we achieve two dealerless traceable threshold encryption constructions with different merits by extending the PLBE primitive of Boneh et al. (Eurocrypt'06) and combining it with the silent setup threshold encryption construction of Garg et al. (Crypto'24). Our first construction achieves an amortized ciphertext of size $O(1)$ (for $O(n)$ ciphertexts). Our second construction achieves constant ciphertext size even in the worst case but requires a less efficient preprocessing phase as a tradeoff. Both our constructions enjoy a constant secret key size and do not require any interaction between the parties.

An additional restriction in the constructions of Boneh et al. is that they can only guarantee to find at least one colluder, leaving techniques to identify more traitors as an open problem. In this paper, we take a first step towards solving this question by formalizing a technique and applying it to our first construction. Namely, our first construction enables tracing t traitors.

1 Introduction

Threshold encryption [22, 23] is a fundamental tool for secure encryption amongst multiple parties, as it allows generating a *succinct* ciphertext which can be decrypted by a committee of parties. At its core, semantic security holds only as long as less than t-out-of-n parties are compromised. While assuming that an adversary cannot obtain t shares seems reasonable, this model is insufficient in practice. In the real world, committee members might be incentivized to give away or sell their secret keys or decryption shares, potentially allowing an adversary to break the scheme's security.

*This work was partially supported by the European Research Council (ERC) under the European Union's Horizon 2020 and Horizon Europe research and innovation programs (grant CRYPTOLAYER-101044770) and the German Research Foundation (DFG) via the DFG CRC 1119 CROSSING (project S7).

Recently, Boneh, Partap, and Rotem [9] addressed this problem by establishing the notion of traceability in threshold encryption schemes. Namely, they consider an adversarially produced stateless decoder box built by a set of at least t colluders, where the decoder takes a ciphertext as input and outputs its decryption. Then, given black-box access to the decoder, a party called the tracer can trace back the decoder to at least one of the colluders who built it. In [9], they provide three constructions with different merits. Their approach is to adapt existing traitor tracing schemes (in the non-threshold setting) to the threshold encryption setting. In particular, they convert the schemes of [8] and [30]. However, all of their constructions suffer from a major drawback: the requirement of a trusted dealer to distribute the secret shares among the parties. In many applications of threshold encryption, this dealer imposes a severe compromise on security. For example, in encrypted mempools for blockchains [3, 18, 25], which motivated the work of [9], a trusted authority could violate the encrypted transactions' privacy, resulting in a drastic financial loss.

Unfortunately, the standard technique of replacing the trusted dealer with a traditional distributed key generation (DKG) [37, 29, 41] protocol is not feasible since, in such protocols, each coalition of t parties can compute the secret share of any other party. Consequently, such a coalition can frame an honest party by embedding its secret share in the generated decoder box. One might consider replacing the trusted dealer with a generic MPC protocol, but this solution is much less efficient than DKG solutions in terms of communication and computation complexity. Therefore, an interesting direction that was left open by [9] is to construct a DKG protocol among the parties and the tracer, where the tracer learns only the information needed for tracing, and the parties only learn the information required for decryption. In particular, a crucial requirement from such DKG is that a set of parties of size t must not learn the secret shares of the remaining $n - t$ parties. Motivated by the importance of getting rid of a trusted dealer in threshold encryption and the significance of the succinct ciphertext in threshold encryption, the main question we study in this paper is:

Question 1.1. *Can we construct an efficient traceable threshold encryption scheme with constant-size ciphertext*[1] *that does not require a trusted dealer?*

In this paper, we answer this question in the affirmative by introducing new traceable threshold encryption constructions in the *silent setup model* [27, 28, 20]. In this setting, each party samples its secret share and publishes some auxiliary public information that facilitates the decryption. Thus, since secret shares are entirely independent, such a setting maintains the aforementioned crucial property in a traceable threshold encryption scheme without requiring a trusted dealer. Furthermore, as there is no interaction between the parties in the setup phase (hence called silent), this setting is much more practical than the generic MPC-based and traditional DKG solutions.

Another drawback of the constructions of [9] is that they can only guarantee to trace the decoder back to at most one traitor from the set of colluders. Indeed, this restriction is inherent to the techniques they use in their constructions. This is because they adapt non-threshold traitor tracing schemes to the threshold setting, but the techniques used in these schemes are designed to catch a single traitor, which is the best one can hope for in the non-threshold setting. In threshold traitor tracing, however, catching up to t traitors could be achievable since a minimum of t colluders are required to construct a successful decoder. Realizing constructions that support tracing more traitors requires fundamentally new definitions and techniques. Therefore, the following is another vital question left open by [9].

Question 1.2. *Can we construct an efficient traceable threshold encryption scheme that guarantees to trace any decoder back to more than one, optimally t, colluders?*

[1] Note that there is a trivial construction with $O(n)$ size ciphertext.

In this paper, we take a first step towards answering this question. In particular, we put forward formal definitions that capture this problem and present a construction with an $O(1)$ amortized ciphertext size (and linear in the worst case), which is guaranteed to trace t colluders.

The Model. Following [9], we consider a model where colluders build a stateless decoder. The tracing is done via black-box access to the decoder. For successful tracing, we require that the tracing algorithm outputs a subset of colluders without framing any innocent parties. We emphasize that our first construction works in the public tracing model, i.e., there is no trusted authority for tracing, and anyone can trace a decoder. Conversely, in our second and main construction, tracing requires some *private* information (i.e., the tracing key tk). Hence, only the party that knows the tracing key can trace decoders. Crucially, the tracer is not trusted for confidentiality. That is, an adversary who obtains the tracing key cannot break the semantic security of the scheme as long as it has corrupted less than t parties. In other words, in the worst case, if the tracer acts maliciously, the security guarantees in our construction will be as in traditional threshold encryption schemes. This feature distinguishes our construction from [9], which requires a trusted tracer and a trusted dealer.

1.1 Our Contributions

Our contributions can be summarized as follows.

- *New Model:* We introduce a new model for constructing traceable threshold encryption schemes. Namely, we define *Traceable Silent Threshold Encryption* (TSTE). In our model, we require neither a trusted dealer nor an interaction between the parties. Furthermore, we also present a model with a preprocessing phase, which enables us to construct more efficient schemes.

- *New Building Block:* We give a formal treatment of *Oblivious Silent Threshold Encryption* (OSTE), a new building block that we define. It extends silent threshold encryption (STE) [28], allowing it to be used for tracing. We define the primitive and provide formal security notions. In particular, we elevate a well-known primitive called *Private Linear Broadcast Encryption* (PLBE) [11] to the threshold encryption setting. We introduce relaxed requirements compared to PLBE in our definition and show how to realize a more efficient TSTE scheme using them. Furthermore, we transform any OSTE scheme, which satisfies our relaxed requirements, to *Traceable Silent Threshold Encryption* (TSTE).

- *New Constructions:* We present two OSTE constructions (and hence TSTE constructions) that are built upon [28]. Compared to [9], our constructions require neither a trusted dealer nor interaction between the parties. Importantly, we achieve constructions with parameter sizes comparable to [9]. In particular, our main construction enjoys constant-size ciphertext and secret keys. We summarize and compare our constructions to [9] in Figure 1.

- *More Traitors:* We formalize a technique to guarantee that more than one traitor is identified in tracing. We apply the technique to our first construction, enabling it to trace t traitors.

1.2 Overview of Our Constructions

In this paper, we follow the line of work on revocation-based traitor tracing. In particular, we adapt Private Linear Broadcast Encryption (PLBE) [11] to the threshold setting. Briefly, there are two encryption methods in PLBE: Enc and TrEnc. The Enc method is used for regular

Construction	|sk|	|pk|	|ct|	No trusted dealer?	Public tracing?	Traitors found
Naive [9]	$O(1)$	$O(n)$	$O(n)$	✓	✓	t
First [9]	$O(n^2)$	$O(n^2)$	$O(1)$	✗	✗	1
Second [9]	$O(n^2)$	$O(1)$	$O(1)$	✗	✗	1
Third [9]	$O(1)$	$O(n^{1/3})$	$O(n^{1/3})$	✗	✗	1
This paper: Sec. 5.2	$O(1)$	$O(n^2)$	$O(n)$ worst case $O(1)$ amortized	✓	✓	t
This paper: Sec. 6	$O(1)$	$O(n^4)$	$O(1)$	✓	✗	1

Figure 1: Our constructions compared to [9]. The comparatively best results are highlighted in green.

encryption and TrEnc for tracing. TrEnc takes an index $i \in \{0, \ldots, n\}$ as input and encrypts the message such that only parties with index $> i$ can decrypt. Then, 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 revoked. Tracing from PLBE relies on three critical PLBE properties. The first is indistinguishability, by which we require Enc and $\mathsf{TrEnc}(0)$ to be indistinguishable except with negligible probability. The second is index hiding, by which we require that any adversary that does not know sk_i can distinguish between $\mathsf{TrEnc}(i-1)$ and $\mathsf{TrEnc}(i)$ only with negligible probability. Third, message privacy requires that ciphertexts from $\mathsf{TrEnc}(n)$ (i.e., everyone is revoked) can not be decrypted except with negligible probability. Intuitively, these properties facilitate tracing 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.

In our constructions, we show how to achieve traceability from the construction of [28] using the aforementioned revocation technique. We realize the equivalent of PLBE in the silent setup threshold encryption setting of [28]. Garg et al. propose a novel threshold encryption scheme in the silent setup setting, where parties can non-interactively publish individual public keys using a one-time setup (CRS). A useful extension of the construction of [28] is that an encryptor can dynamically choose which parties are in the decryption committee for their ciphertext. We utilize this feature to realize TrEnc, allowing the tracer to revoke parties. Notably, our key challenge is to realize TrEnc without revealing which parties have been revoked. Our constructions are proven secure in the Generic Group Model (GGM) [40].

Next, we summarize this paper's main techniques and building blocks.

Oblivious Silent Threshold Encryption. In Section 3, we define *Oblivious Silent Threshold Encryption* (OSTE), which extends silent threshold encryption STE [28] by a revocation functionality (as in PLBE). In Section 4.1, we show how to transform any OSTE scheme to *Traceable Silent Threshold Encryption* (TSTE). In OSTE, we allow a ciphertext ct to be encrypted to a set of parties S such that any entity that does not know the secret share sk_i is oblivious to whether the i-th party is among the set S or not. This extension is powerful in tracing, where the tracer can revoke parties and use a similar algorithm as for PLBE. The OSTE definition introduces the TrEnc algorithm, which is used for tracing queries. Informally, TrEnc takes as parameter a message m and a set of revoked parties $\mathcal{R}$ and encrypts the message m to the set $[n] \setminus \mathcal{R}$.

We establish three requirements for OSTE, analogous to the requirements for PLBE. As

for PLBE, the indistinguishability property requires that normal encryption is indistinguishable from $\mathsf{TrEnc}(\emptyset)$ (i.e., the message is encrypted to all parties). A key difference is the index hiding requirement, which we call membership hiding in OSTE. Informally, membership hiding captures the property of obliviousness, as discussed above. Only the party that holds sk_i can distinguish whether the i-th party is revoked or not. Membership hiding extends index hiding, defined in PLBE, in the sense that we do not fix a specific order of revocations. That is, index hiding is defined for a fixed revocation order $A = \{i_1, i_2, \dots, \}$, but we allow arbitrary orders in our definition. This extension enables us to guarantee tracing more than one traitor. We elaborate on this in Section 4.2. Furthermore, in our definition, we relax the indistinguishability and membership hiding requirements, as follows. In the relaxed indistinguishability, we allow the adversary to distinguish between Enc and $\mathsf{TrEnc}(\emptyset)$ with some fixed non-negligible probability. Similarly, in the relaxed membership hiding, we allow the adversary to distinguish if party sk_i is revoked with some fixed non-negligible probability. Interestingly, in Section 4.1, we transform any OSTE scheme that fulfills our relaxed definition to a traceable threshold encryption scheme. We highlight that this relaxation is essential for our second construction, which requires preprocessing, as discussed later in this section.

We next discuss the main techniques used to achieve our OSTE constructions.

Rerandomization. In the STE construction of [28], when a ciphertext ct is encrypted to a set of parties S (i.e., only parties in S can participate in decryption) and threshold t, each party in S can compute a partial decryption share non-interactively. Afterward, anyone can build an aggregation key ak, combining (at least) t partial decryption shares to decrypt ct. The aggregation key ak is S-specific and can be computed from public information (i.e., the CRS and the public keys of all parties in S). However, the restriction here is that the ak can be computed if and only if the set S is known. As we discussed, for our use case in tracing, we would like to hide whether a party i is in the set S or not (membership hiding of OSTE). Hence, in Section 5, we modify the STE scheme to achieve this property. To this end, we rerandomize all public keys of all the parties by masking them with a random element such that $\mathsf{pk}'_i \leftarrow \alpha \mathsf{pk}_i$ if $i \in S$ and $\mathsf{pk}'_i \leftarrow \bar{\alpha}_i \mathsf{pk}_i$ if $i \notin S$, i.e., all keys of unrevoked parties are masked with the same value. We then compute the aggregation key ak with respect to the new public keys and include it in the ciphertext. We adjust the encryption of [28], accounting for the rerandomization, such that any party can still use its secret share sk_i to compute its partial decryption and prove that parties outside of S cannot participate in decryption. Intuitively, this trick conceals whether a party has been revoked since a decryptor that does not know sk_i cannot distinguish $\alpha \mathsf{pk}_i$ from $\bar{\alpha}_i \mathsf{pk}_i$. We show this by a reduction to the external Diffe Hellman (XDH) assumption (dealing with practical attacks such as rushing adversary).

A drawback of this approach is that we increase the size of the ciphertext by the size of ak, which is linear in the number of parties. However, we observe that ak can be reused in multiple ciphertexts. Hence, we propose two techniques to decrease the ciphertext size: amortization and preprocessing. We sketch the amortization approach in Section 5.3 and construct a second scheme using the preprocessing approach in Section 6.

Preprocessing. In Section 6, we reduce the size of the ciphertext of our first construction by adding a preprocessing phase. This allows us to remove the rerandomized aggregation key ak, which is of size $O(n)$, from the ciphertext. Informally, after all parties have published their public keys, we run a preprocessing phase, in which we generate a polynomial-size set of predefined encryption and aggregation keys $\mathcal{K} = \{(\mathsf{ek}_1, \mathsf{ak}_1), \dots, (\mathsf{ek}_N, \mathsf{ak}_N)\}$. Importantly, the set $\mathcal{K}$ includes keys used to encrypt to all parties and keys used to encrypt only to subsets of parties $S \subseteq [n]$. Let's assume, for simplicity, that $\mathcal{R}_i = \{1, \dots, i\}$, then for $0 \leq i \leq n$, the set

$\mathcal{K}$ should contain a subset of keys $\mathcal{K}_i$ that enables only the parties in $S_i = [n] \setminus \mathcal{R}_i$ to decrypt. Observe that these keys are essential for tracing, where we revoke subsets of parties. All keys in $\mathcal{K}$ are computed using the same algorithm as for the first OSTE construction. Crucially, no one except for the tracer should be able to distinguish which keys correspond to which set, as indistinguishability and membership hiding are not fulfilled otherwise. In Enc, the encryption key ek is chosen uniformly at random from the set $\mathcal{K}$. We next briefly describe the tracing procedure. Let's first consider an attempt to realize the TrEnc algorithm. Recall that the keys in set $\mathcal{K}_i$ revoke $\mathcal{R}_i$, and let's assume that the tracer, in $\mathsf{TrEnc}(\mathcal{R}_i)$, samples a key at random from the set $\mathcal{K}_i$ and encrypts using this key. Unfortunately, this attempt does not satisfy indistinguishability since the keys in $\mathsf{TrEnc}(\emptyset)$ and Enc are chosen from different distributions. To see why this is a problem, consider an adversary that chooses a set of keys $\mathcal{L} \subset \mathcal{K}$ and generates a decoder that decrypts only if it sees one of these keys. As the adversary can distinguish the subset $\mathcal{K}_{i-1}$ if i is a colluder, it can choose, e.g., $\mathcal{L} = \mathcal{K}_{i-1}$. Therefore, such a decoder will decrypt ciphertexts that are encrypted using Enc with some probability ϵ but decrypt $\mathsf{TrEnc}(\emptyset)$ ciphertexts with probability 0, breaking the indistinguishability property. To overcome this, for each S_j for $j \in \{0, \ldots, n\}$, we ensure that the probability of getting a ciphertext that is encrypted to the set S_j (i.e., sampling a key from $\mathcal{K}_j$ in encryption) is the same (up to a negligible factor) in both Enc and $\mathsf{TrEnc}(\mathcal{R}_0)$. While this solution solves the indistinguishability issue, it introduces a similar problem for membership hiding. We solve it similarly by ensuring that the distribution of keys in $\mathsf{TrEnc}(\mathcal{R}_{i-1})$ ciphertexts is indistinguishable from $\mathsf{TrEnc}(\mathcal{R}_i)$ for any adversary who does not know sk_i. Informally, in $\mathsf{TrEnc}(\mathcal{R}_i)$, the key is sampled from the sets $\mathcal{K}_j$, where $j \geq i$ with a certain probability. The intuition is that due to membership hiding property of the first OSTE construction, an adversary that does not hold sk_j cannot distinguish $\mathcal{K}_j$ from $\mathcal{K}_{j-1}$.

Yet, indistinguishability and membership hiding of this construction are not immediate. For example, the adversary could win the membership hiding game for index i with non-negligible probability. To see this, note that the adversary could sample a random key $k \in \mathcal{K}$ and assume that it is in $\mathcal{K}_{i-1}$. Then, given a challenge ciphertext ct, which is encrypted under $\mathsf{TrEnc}(\mathcal{R}_{i-1})$ or $\mathsf{TrEnc}(\mathcal{R}_i)$, it would return 1 if the key used to encrypt ct is k and 0 otherwise. Since the size of $\mathcal{K}$ is polynomial and in $\mathsf{TrEnc}(\mathcal{R}_i)$ the keys are not sampled from $\mathcal{K}_{i-1}$, the adversary would win with non-negligible advantage. However, we prove that by choosing the appropriate parameters, we can upper bound the advantage of the adversary by some δ that is sufficient for traceability following Theorem 4.5. This holds due to our relaxation on the indistinguishability and membership hiding advantage requirements. Intuitively, we prove that the adversary cannot sample a subset of keys $\mathcal{L} \subseteq \mathcal{K}$ such that the probability of sampling a key from $\mathcal{L}$ in $\mathsf{TrEnc}(\mathcal{R}_i)$ falls apart by more than δ from sampling a key from $\mathcal{L}$ in $\mathsf{TrEnc}(\mathcal{R}_{i-1})$. Finally, to amplify the correctness of the scheme, we require the encryptor to choose λ_c keys from the set $\mathcal{K}$ and encrypt the message using them.

1.3 Discussion and Future Work

We next discuss interesting extensions, limitations, and open problems left by our work.

Multiverse Threshold Encryption. As mentioned in [28], their silent threshold encryption construction also implies a threshold encryption construction in the multiverse setting [1]. In this setting, a universe is defined as a tuple (U, t_U), where $U \subseteq [n]$ and t_U is the associated threshold; only parties in U can participate in decryption. In multiverse, we support arbitrary universes, so each party needs to publish a single $(\mathsf{sk}, \mathsf{pk})$ pair for use in any of its universes. Our constructions inherit this feature from [28]. In particular, in our second construction, which requires preprocessing, we can run preprocessing per universe. Note that this is another advantage

of our approach compared to the generic MPC solution. In MPC, a party must be involved in the execution of MPC for each of its universes, while in our approach, it needs to publish a single public key only once.

Broadcast Encryption. Our constructions also imply a traitor tracing scheme in the broadcast encryption setting (when $t = 1$) without a central authority. This may be compared to the tracing scheme without central authority of [14]. In particular, we achieve a constant size ciphertext where they achieve a $\sqrt{n}$ ciphertext. However, in their construction, they rely on a transparent setup to generate the CRS, whereas in our construction, as in the underlying STE construction of [28], we rely on a structured CRS which is similar to the KZG polynomial commitment scheme [34]. We leave the problem of getting rid of the non-transparent CRS as an interesting future direction.

Decryption Oracle. In our main construction, we assume an adversary that does not have access to a decryption oracle. We note that this is consistent with other traitor tracing schemes, particularly the work of [9]. Unfortunately, traceability is not guaranteed in our preprocessing construction when the adversary gets access to a decryption oracle since then the adversary can learn non-trivial information about the tracing key (by inspecting which encryption keys produce correct decryption and which do not). Hence, an interesting question for future research is how to achieve traceability in the presence of a decryption oracle. We note that a possible direction is to restrict ourselves to the model in which the number of decryption oracle calls is bounded by a polynomial [19] (that is fixed in the setup), thus restricting the information the adversary can reveal about the tracing key.

More traitors. Our first construction guarantees catching t traitors. However, our second construction guarantees catching only a single traitor. A possible direction for future work is to adapt our technique to enable tracing more than one traitor with our second construction.

Efficiency and Practicality. While we achieve efficient sizes asymptotically, our second construction does not scale well practically, as it requires a relatively expensive preprocessing of size $O(n^4)$. However, we provide the construction as a proof of concept for a dealerless traceable threshold encryption construction with constant ciphertext size, which does not require any interaction between the parties. An interesting direction is to explore how to improve the efficiency of the preprocessing phase. In particular, to reduce the dependency on n.

1.4 Additional Related Work

To the best of our knowledge, there are no other traceable threshold encryption constructions besides [9]. Some other recent works study collusion resistance of secret sharing [10, 24, 33], but it is not clear how to extend these techniques to construct traceable threshold cryptography schemes. The notion of traitor tracing was first introduced by Chor et al. [17], initially referring to piracy protection for proprietary data. Since then, there has been extensive research on traitor tracing in the broadcast encryption setting (see, e.g., [42] for a good overview). Mostly, the works on traitor tracing aim to achieve optimal parameter sizes, in particular, obtaining a sublinear ciphertext. The most recent advance in this direction was due to Zhandry [43], who constructed a pairing-based traitor tracing scheme with optimal parameters, i.e., all the parameters are independent of the number of parties. Some other works achieved similar parameters but using other tools such as LWE [31, 16] and indistinguishability obfuscation (iO) [12, 32].

The most related traitor tracing construction to our work is the construction of Branco et al. [14]. They address a similar problem but in the non-threshold setting. Namely, they aim to get rid of the central authority that distributes keys among the parties. As a building block, they construct a pairing-based registered functional encryption scheme (RFE) [26] in which, like the silent setup setting of [28], the parties sample their keys independently and publish auxiliary information. They show that traitor tracing can be reduced to realizing Quadratic RFE and achieve a construction with sub-linear ciphertext. Other recent works also studied paring-based QFE in the registered setting [21, 44]. Another work addressing traitor tracing without central authority is due to Luo [35], but this construction uses indistinguishability obfuscation (iO), an inefficient tool.

2 Preliminaries

Notation. We denote the security parameter by λ and the correctness parameter by λ_c. We write $\mathsf{negl}(\lambda)$ for functions that are negligible in λ and $\mathsf{poly}(\lambda)$ for polynomials in λ. We denote the number of parties by n. In the O notation we ignore factors other than n. 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. Further, we use $\approx_c$ to denote that two random variables are computationally indistinguishable. For a security game Game, we denote by $\mathbf{Adv}^{\mathsf{Game}}_{\mathcal{A}}(x)$ the success probability of the adversary $\mathcal{A}$ when running in the security game $\mathsf{Game}(x)$ ($\mathbf{Adv}^{\mathsf{Game}}_{\mathcal{A}}(x) = \Pr[\mathsf{Game}_{\mathcal{A}}(x) = 1]$). If Game is a distinguishing game (i.e. if the adversary is tasked with guessing a random bit b), then the advantage is the success probability over guessing the bit $\mathbf{Adv}^{\mathsf{Game}}_{\mathcal{A}}(x) = \Pr[\mathsf{Game}_{\mathcal{A}}(x) = 1] - 1/2$. We denote by $\mathbb{H} = \{\omega, \omega^2, \ldots, \omega^l\}$ the multiplicative subgroup generated by the l-th root of unity $\omega \in \mathbb{Z}_p$ ($\omega^l = 1$) for $l = |\mathbb{H}| = n + 1$. Further, we denote by $L_i(x)$ the Lagrange basis polynomial of element i with respect to $\mathbb{H}$. By $Z(x)$, we denote the vanishing polynomial on $\mathbb{H}$, i.e., $Z(x) = x^l - 1$.

Lemma 2.1 (Univariate Sumcheck [4, 38]). *Let $n = \mathsf{poly}(\lambda)$, $A(x) = \sum_{i=1}^{|\mathbb{H}|} a_i \cdot L_i(x)$ and $B(x) = \sum_{i=1}^{|\mathbb{H}|} b_i \cdot L_i(x)$. It holds that*

$$A(x) \cdot B(x) = \frac{\sum_i a_i b_i}{|\mathbb{H}|} + Q_x(x) \cdot x + Q_Z(x) \cdot Z(x),$$

where where both Q_x and Q_Z are polynomials with degree $\leq |\mathbb{H}| - 2$ defined as

$$Q_x(x) = \sum_i a_i b_i \frac{L_i(x) - L_i(0)}{x}$$

$$Q_Z(x) = \sum_i a_i b_i \frac{L_i^2(x) - L_i(x)}{Z(x)} + \sum_{i \neq j} a_i b_j \frac{L_i(x) L_j(x)}{Z(x)}$$

The original sumcheck is concretely stated for $b_i = 1$. In this case, we treat general inner products as a straightforward generalization (see [15]).

Non-Interactive Zero-Knowledge Proofs of Knowledge. Our constructions rely on Non-Interactive Zero-Knowledge Proofs of Knowledge (NIZK-PoKs). A NIZK-PoK proof system PS_{R} is a tuple of algorithms $\mathsf{PS}_{\mathsf{R}} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}, \mathsf{Ext})$ to prove knowledge of witnesses w to statements χ in the corresponding relation R. A detailed definition can be found in Appendix A.

Bilinear Pairings. A bilinear pairing ensemble $E = (\mathbb{G}_1, [1]_1, \mathbb{G}_2, [1]_2, \mathbb{G}_T, p, \circ)$ is an ensemble of cyclic groups $\mathbb{G}_1$ with generator $[1]_1$, $\mathbb{G}_2$ with generator $[1]_2$ and $\mathbb{G}_T$ of prime order p with a pairing operation $\circ\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies bilinearity and non-degeneracy. We usually write $\circ$ infix.

We rely on the external Diffie-Hellman assumption [2, 5].

Definition 2.2 (XDH). Let $\mathcal{G}$ be an algorithm to generate a pairing ensemble E. The external Diffie-Hellman problem (XDH) is hard for $\mathcal{G}$ if for all PPT adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\left|\Pr[\mathcal{A}(1^\lambda, E, [x]_1, [y]_1, [xy]_1) = 1] - \Pr[\mathcal{A}(1^\lambda, E, [x]_1, [y]_1, [v]_1) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $E \xleftarrow{\$} \mathcal{G}(1^\lambda)$ and $x, y, v \xleftarrow{\$} \mathbb{Z}_p$.

Some of our security proofs rely on the Generic Group Model (GGM) [40]. We also stress that the XDH assumption holds in generic groups. We list the necessary definitions and theorems in Appenidix A.

3 Oblivious Silent Threshold Encryption

In this section, we define *oblivious silent threshold encryption* (OSTE), extending the definition of *silent threshold encryption* [28].

Definition 3.1 (Oblivious Silent Threshold Encryption). An oblivious silent threshold encryption scheme consists of a tuple of algorithms (Setup, KGen, IsValid, Prep, Enc, TrEnc, PartDec, PartVfy, DecAggr) with the following syntax:

- $\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$: On input the security parameter λ and number of parties n, Setup outputs a common reference string CRS.
- $(\mathsf{sk}, \mathsf{pk}, \mathsf{hint}, \pi) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, \mathsf{CRS})$: On input the CRS, KGen outputs a secret key sk, a public key pk, a hint, and a proof of knowledge π that the party knows sk corresponding to pk and hint.
- $1/0 \leftarrow \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}, \mathsf{hint}, \pi)$: On input the CRS, a public key pk, a hint hint, and a proof π, IsValid outputs 1 if the public key is valid and 0 otherwise.
- $(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]})$: The Prep algorithm receives the CRS and all public keys and hints $\{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}$ as input. It outputs a preprocessing $\mathcal{K}$, a proof π, and a tracing key tk.
- $1/0 \leftarrow \mathsf{PreVfy}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{K}, \pi)$: Given the CRS, n public keys and hints as well as a preprocessing $\mathcal{K}$ and corresponding proof π, PreVfy outputs 1 if the preprocessing proof is valid and 0 otherwise.
- $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m})$: On input the CRS, a preprocessing $\mathcal{K}$, a threshold t, and a message m, Enc outputs a ciphertext ct.
- $\mathsf{ct} \xleftarrow{\$} \mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}, \mathcal{R})$: Given the CRS, the tracing key tk, a preprocessing $\mathcal{K}$, a threshold t, the message m, and a set of revoked parties $\mathcal{R} \subseteq [n]$, TrEnc outputs a ciphertext ct. Sometimes, we abuse notation and write $\mathsf{TrEnc}(\mathcal{R})$ to refer to trace-encrypt with revoked set $\mathcal{R}$ whenever the remaining inputs are clear from the context.

- $\sigma \leftarrow \mathsf{PartDec}(\mathsf{sk}, \mathsf{ct})$: On input a secret key sk and a ciphertext ct, $\mathsf{PartDec}$ outputs a partial decryption σ.
- $\mathsf{m} \leftarrow \mathsf{DecAggr}(\mathsf{CRS}, \mathcal{K}, \mathsf{ct}, \{\sigma_i\}_{i \in S})$: On input the CRS, the set of keys $\mathcal{K}$, a ciphertext ct and a set of partial decryptions $\{\sigma_i\}_{i \in S}$, $\mathsf{DecAggr}$ outputs a message m.

The correctness is standard, i.e., we require that every coalition $S \subseteq [n]$, where $|S| \geq t$, decrypts with overwhelming probability (see Appendix B for a formal definition).

We require three tracing-related properties named *indistinguishability*, *membership hiding*, and *message privacy*. To formally define these, we first introduce an additional parameter $\mathcal{X}$. For an OSTE scheme, $\mathcal{X}$ denotes the set of all possible sets of parties that can be revoked using TrEnc (hence $\mathcal{X} \subseteq \mathcal{P}([n])$). In standard PLBE, one can only revoke parties linearly (i.e. $\mathcal{X} = \{\emptyset, \{1\}, \{1,2\}, \ldots, [n]\}$). Looking ahead, we want to construct OSTE schemes that allow for more than one, or even all possible revocation orders ($\mathcal{X} = \mathcal{P}([n])$), which we will use to identify more than 1 and up to t traitors (see Section 4.2).

Goyal et al. [31] show that only decoder-based definitions of indistinguishability and membership hiding imply a (private) tracing scheme. In the following, we present definitions of distinguishing decoders as discussed in [31].

Definition 3.2 (Distinguishing Decoders [31]). For any $\delta \in [-1/2, 1/2]$, PPT algorithm $\mathcal{D}$, $n = \mathsf{poly}(\lambda)$, $t < n$, any CRS, any $\mathsf{m} \in \mathbb{G}_T$, and any preprocessing $\mathcal{K}$ we say that

- $\mathcal{D}$ is δ-$\mathrm{Dist}^{\mathsf{Enc},\mathsf{TrEnc}}$, if

$$\Pr\left[\mathcal{D}(\mathsf{ct}_b) = b \;\middle|\; \begin{array}{c} b \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m}); \\ \mathsf{ct}_1 \leftarrow \mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}, \emptyset) \end{array}\right] \geq \frac{1}{2} + \delta.$$

- $\mathcal{D}$ is δ-$\mathrm{Dist}^{\mathcal{R}, \mathcal{R}\cup\{i\}}$ for revoked set $\mathcal{R} \subseteq [n]$ if

$$\Pr\left[\mathcal{D}(\mathsf{ct}_b) = b \;\middle|\; \begin{array}{c} b \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}_0 \leftarrow \mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}, \mathcal{R}); \\ \mathsf{ct}_1 \leftarrow \mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}, \mathcal{R} \cup \{i\}) \end{array}\right] \geq \frac{1}{2} + \delta.$$

Indistinguishability. The indistinguishability notion captures that one cannot construct a decoder that has advantage δ for distinguishing if a given ciphertext ct was created using either Enc or TrEnc with no revoked parties ($\mathcal{R} = \emptyset$). In the indistinguishability game $\mathsf{GameInd}$ (Figure 2), the adversary can control *all* parties. The adversary wins the indistinguishability game if the decoder is δ-$\mathrm{Dist}^{\mathsf{Enc},\mathsf{TrEnc}}$ according to Definition 3.2.

Definition 3.3 (δ-Indistinguishability.). An OSTE scheme $\mathcal{E}$ is δ-$\mathsf{indistinguishable}$ for $\delta \in [-1/2, 1/2]$ if for all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that $\mathbf{Adv}^{\mathsf{GameInd}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n, \delta) \leq \mathsf{negl}(\lambda)$, where $\mathsf{GameInd}_{\mathcal{A}}$ is defined in Figure 2.

δ-Membership Hiding. The δ-membership hiding notion is a generalization of the index hiding property for PLBE [11]. Intuitively, our definition states that an adversary can not build a decoder that has advantage δ of distinguishing whether a party i is revoked for a ciphertext or not, even if the adversary corrupts *all* parties other than i and chooses the remaining set of revoked parties $\mathcal{R}$. The membership hiding game GameMH accounts for this by verifying that both $\mathcal{R} \in \mathcal{X}$ and $\mathcal{R} \cup \{i\} \in \mathcal{X}$ (i.e., both sets *can* be revoked by the scheme).

$\mathsf{GameInd}_{\mathcal{A}}(1^\lambda, n, \delta)$:

$\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$
$(\{\mathsf{pk}_\ell, \mathsf{hint}_\ell, \pi_\ell\}_{\ell\in[n]}, t, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{CRS})$
$(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]})$
$(\mathcal{D}, \mathsf{m}) \xleftarrow{\$} \mathcal{A}(\mathcal{K}, \pi, \mathsf{st})$
return $\delta\text{-Dist}^{\mathsf{Enc},\mathsf{TrEnc}}(\mathcal{D})$

$\mathsf{GameMH}_{\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta)$:

$\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$
$(i, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{CRS})$
$(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, \mathsf{CRS})$
$(\mathcal{R}, \{\mathsf{pk}_\ell, \mathsf{hint}_\ell, \pi_\ell\}_{\ell\in[n]\setminus\{i\}}, t, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\mathsf{pk}_i, \mathsf{hint}_i, \pi_i, \mathsf{st})$
$(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]})$
$(\mathcal{D}, \mathsf{m}) \xleftarrow{\$} \mathcal{A}(\mathcal{K}, \pi, \mathsf{st})$
return $\delta\text{-Dist}^{\mathcal{R},\mathcal{R}\cup\{i\}}(\mathcal{D})$
$\wedge\ \mathcal{R} \in \mathcal{X} \wedge \mathcal{R}\cup\{i\} \in \mathcal{X} \wedge i \in [n] \wedge i \notin \mathcal{R}$
$\wedge\ \forall i \in [n]\colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i)$
$\wedge\ \forall i, j \in [n], i \neq j\colon \mathsf{pk}_i \neq \mathsf{pk}_j$

Figure 2: The indistinguishability and membership hiding games for OSTE schemes.

Definition 3.4 (δ-Membership Hiding). An OSTE scheme $\mathcal{E}$ is δ-membership hiding for a subset $\mathcal{X} \subseteq \mathcal{P}([n])$ and $\delta \in [-1/2, 1/2]$ if for all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that $\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n, \mathcal{X}, \delta) \leq \mathsf{negl}(\lambda)$, where $\mathsf{GameMH}_{\mathcal{A}}$ is defined in Figure 2.

$\mathsf{GameMP}_{\mathcal{A}}(1^\lambda, n, \mathcal{X})$:

$\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$
$(\mathcal{T}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{CRS})$
for $i \in [n], i \notin \mathcal{T}$ **do**
$(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, \mathsf{CRS})$
$(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\in\mathcal{T}}, t, \mathcal{R}, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\notin\mathcal{T}}, \mathsf{st})$
$(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]})$
$b \xleftarrow{\$} \{0, 1\}$
$\mathsf{ct} \xleftarrow{\$} \mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}_b, \mathcal{R})$
$b' \xleftarrow{\$} \mathcal{A}^{\mathsf{TrEnc}(\mathsf{CRS},\mathsf{tk},\mathcal{K},t,\cdot,\mathcal{R})}(\mathsf{st}, \mathcal{K}, \pi, \mathsf{ct})$
return $b = b' \wedge |\mathcal{T} \setminus \mathcal{R}| < t \wedge \mathcal{T} \subseteq [n] \wedge \mathcal{R} \in \mathcal{X}$
$\wedge\ \forall i \in \mathcal{T}\colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i)$
$\wedge\ \forall i, j \in [n], i \neq j\colon \mathsf{pk}_i \neq \mathsf{pk}_j$
$\wedge\ \mathsf{PreVfy}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{K}, \pi)$

Figure 3: The message privacy game for OSTE schemes.

Message Privacy. For message privacy, we ensure that an adversary that holds less than t shares of *unrevoked* parties cannot decrypt the message. This is different from normal IND-CPA since the adversary may corrupt more than t parties overall.

Definition 3.5 (Message Privacy for OSTE). An OSTE scheme $\mathcal{E}$ fulfills message privacy for a subset $\mathcal{X} \subseteq \mathcal{P}([n])$ if for all PPT adversaries $\mathcal{A}$, it holds that $\mathbf{Adv}^{\mathsf{GameMP}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n, \mathcal{X}) \leq \mathsf{negl}(\lambda)$, where $\mathsf{GameMP}_{\mathcal{A}}$ is defined in Figure 3.

Security. Our security notion is a standard IND-CPA definition, i.e., an adversary who corrupts a set of parties $|\mathcal{T}| < t$ cannot decrypt ciphertexts. Note that in this game, the adversary generates the preprocessing $\mathcal{K}$ itself (it outputs $\mathcal{K}$ and π). This captures the requirement that the party that generates the preprocessing *is untrusted for security*, but only for traceability. We refer to the full definition in Appendix B.

4 Traceable Silent Threshold Encryption

In this section, we define *traceable silent threshold encryption* (TSTE) and present a reduction from OSTE to TSTE.

Definition 4.1 (Traceable Silent Threshold Encryption). A traceable silent threshold encryption scheme consists of a tuple of algorithms (Setup, KGen, IsValid, Prep, PreVfy, Enc, PartDec, PartVfy, DecAggr, Trace), such that:

- $\mathsf{ct} \xleftarrow{\$} \mathsf{Trace}^{\mathcal{D}}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \epsilon)$: Given the CRS, the preprocessing $\mathcal{K}$, the tracing key tk, a bound on the quality of a decoder ϵ, and blackbox access to a decoder $\mathcal{D}$, Trace outputs a list of colluding parties $\mathcal{T}$.

The remaining algorithms are defined as for OSTE schemes (Definition 3.1).

The correctness and CPA-security of TSTE are defined similarly to OSTE. We present them in Appendix C.

$\mathsf{GameTR}_{\mathcal{A}}(1^\lambda, n, \epsilon)$:

$\mathsf{CRS} \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda, n, c)$
$(\mathcal{T}, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}(1^\lambda, \mathsf{CRS})$
for $i \notin \mathcal{T}$ **do**
  $(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \leftarrow\!\!\$\ \mathsf{KGen}(1^\lambda, \mathsf{CRS}, n)$
$(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\in\mathcal{T}}, t, \mathsf{st}) \leftarrow \mathcal{A}(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\notin\mathcal{T}}, \mathsf{st})$
$(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \epsilon)$
$(\mathcal{D}, \mathsf{m}_0, \mathsf{m}_1) \xleftarrow{\$} \mathcal{A}(\mathcal{K}, \pi, \mathsf{st})$
$\mathcal{T}' \leftarrow \mathsf{Trace}^{\mathcal{D}}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \epsilon)$
return $\epsilon\text{-Dist}^{\mathsf{m}_0,\mathsf{m}_1}(\mathcal{D}) \wedge (\mathcal{T}' = \emptyset \vee \mathcal{T}' \not\subseteq \mathcal{T})$
  $\wedge\, \forall i \in \mathcal{T}\colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \wedge \forall i, j \in [n], i \neq j\colon \mathsf{pk}_i \neq \mathsf{pk}_j$

Figure 4: Decoder-based traceability definition (GameTR) for TSTE schemes. Note that the Prep algorithm is executed by a trusted party in GameTR, but is run by the adversary in IND-CPA.

Traceability. Our decoder-based definition of traceability makes use of distinguishing decoders.

Definition 4.2 (Distinguishing Decoders for TSTE [31]). For any $\epsilon \in \left[-\frac{1}{2}, \frac{1}{2}\right]$, PPT algorithm $\mathcal{D}$, $n = \mathsf{poly}(\lambda)$, $t < n$, any CRS, and any preprocessing $\mathcal{K}$ and tracing key tk, we say $\mathcal{D}$ is $\epsilon\text{-Dist}^{\mathsf{m}_0,\mathsf{m}_1}$ for messages $\mathsf{m}_0, \mathsf{m}_1$, if

$$\Pr\left[\mathcal{D}(\mathsf{ct}_b) = b \;\middle|\; \begin{array}{c} b \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}_b \leftarrow \mathsf{Enc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}_b); \end{array}\right] \geq \frac{1}{2} + \epsilon.$$

We say that $\mathcal{D}$ is an ϵ-good decoder if $\mathcal{D}$ is $\epsilon\text{-Dist}^{\mathsf{m}_0,\mathsf{m}_1}$.

Definition 4.3 (Traceability for TSTE). We call a TSTE scheme $\mathcal{E}$ traceable for $\epsilon \in [-1/2, 1/2]$, if for all PPT adversaries $\mathcal{A}$ and $n = \mathsf{poly}(\lambda)$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that $\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n, \epsilon) \leq \mathsf{negl}(\lambda)$, where $\mathsf{GameTR}_{\mathcal{A}}$ is defined in Figure 4.

4.1 Transforming OSTE to Traceable Silent Threshold Encryption

In this section, we show that an OSTE scheme can be used to build a TSTE scheme. Let $\mathcal{E}$ be an OSTE scheme as defined in Definition 3.1. We construct a TSTE scheme Π_{TSTE} that builds upon $\mathcal{E}$. In fact, all TSTE algorithms that are also defined for OSTE run the corresponding algorithm of $\mathcal{E}$. In addition, we construct a Trace algorithm for Π_{TSTE}, for which we present pseudocode in Figure 5. The Trace algorithm proceeds similarly to tracing algorithms built for PLBE schemes [11]. However, in the threshold encryption setting, if the decoder shows a high success probability after we revoke $n - t$ parties, we consider all t remaining parties to be colluders.

Remark 4.4 For simplicity, we assume the revocation order $\mathcal{L} = \{\mathcal{R}_i\}_{i\in[0,n]}$, where $\mathcal{R}_i = \{1, 2, \ldots, i\}$ and $\mathcal{R}_0 = \emptyset$. However, our Trace algorithm does not require a specific order of revocation. That is, a tracer is free to choose any order of revocation from the orders that are supported by the underlying OSTE construction (i.e., within the set $\mathcal{X}$ in Definition 3.4). In Section 4.2, we discuss how we utilize this property to guarantee tracing more than one traitor.

$\mathsf{Trace}^{\mathcal{D},\mathsf{m}_0,\mathsf{m}_1}(\mathsf{CRS}, \mathcal{K}, \mathsf{tk}, t, \epsilon)$

$W = 2^7 \cdot 3n^2\lambda/\epsilon$; $p_{-1} \leftarrow \epsilon$; $\mathcal{T}' \leftarrow \emptyset$
for $i \in [0, n-t]$
  $a_i \leftarrow 0$
  for $j \in [W]$
    $b \xleftarrow{\$} \{0,1\}$
    $\mathsf{ct} \xleftarrow{\$} \mathcal{E}.\mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}_b, \mathcal{R}_i)$
    if $\mathcal{D}(\mathsf{ct}) = b$ **then** $a_i \leftarrow a_i + 1$
  $\hat{p}_i \leftarrow a_i/(W)$
  CONDITION 1. **if** $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ **then** $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{i\}$
CONDITION 2. **if** $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$ **then** $\mathcal{T}' \leftarrow \mathcal{T}' \cup \{n-t+1, \ldots, n\}$
return $\mathcal{T}'$

Figure 5: Π_{TSTE}: Blackbox tracing from an OSTE scheme, denoted by $\mathcal{E}$.

Theorem 4.5 *Let $\mathcal{E}$ be an OSTE scheme for the linear revocation order $\mathcal{X} = \mathcal{L}$, as defined in Definition 3.1, with $(\frac{\epsilon}{16})$-Indistinguishability (by Definition 3.3) and $(\frac{\epsilon}{16n})$-Membership hiding (by Definition 3.4) for $\epsilon \in [-1/2, 1/2]$ where $\epsilon = 1/\mathsf{poly}(\lambda)$. Then, for every $n \in \mathsf{poly}(\lambda)$, and every $\lambda \in \mathbb{N}$, the Π_{TSTE} scheme, depicted in Figure 5, is a traceable silent threshold encryption scheme (by Definition 4.1) that fulfills traceability for ϵ. Furthermore, the secret share size, the public key size, and the message size are as in $\mathcal{E}$.*

The correctness and security are straightforward. For completeness, we put the proofs in Appendix D. We next discuss traceability. The major challenge in proving traceability compared to classical PLBE scheme is that in our setting, we allow the adversary to distinguish between Enc and $\mathsf{TrEnc}(\mathcal{R}_0)$ ciphertexts and between $\mathsf{TrEnc}(\mathcal{R}_{i-1})$ and $\mathsf{TrEnc}(\mathcal{R}_i)$ with non-negligible

advantage. Nevertheless, we show that we can build a TSTE scheme that fulfills traceability from an OSTE scheme that is δ_1-indistinguishable and δ_2-membership hiding, per Definitions 3.3 and 3.4, respectively, where δ_1 and δ_2 are non-negligible and fixed in the proof. That is, we prove the following lemma.

Lemma 4.6 *For every $n \in \mathsf{poly}(\lambda)$, every $\epsilon \in [-1/2, 1/2]$ where $\epsilon = 1/\mathsf{poly}(\lambda)$, every $\lambda \in \mathbb{N}$, and every probabilistic polynomial-time adversary $\mathcal{A}$ there exist probabilistic polynomial-time adversaries $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_3$ such that*

$$\begin{aligned}\boldsymbol{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}\left(1^\lambda, n, \epsilon\right) \leq & \boldsymbol{Adv}^{\mathsf{GameInd}}_{\mathcal{A}_1,\mathcal{E}}\left(1^\lambda, n, \delta_1\right) + \boldsymbol{Adv}^{\mathsf{GameMP}}_{\mathcal{A}_2,\mathcal{E}}\left(1^\lambda, n\right) \\ & + n \cdot \boldsymbol{Adv}^{\mathsf{GameMH}}_{\mathcal{A}_3,\mathcal{E}}\left(1^\lambda, n, \delta_2\right) + \mathsf{negl}(\lambda)\end{aligned}$$

for $\delta_1 = \frac{\epsilon}{16}$, $\delta_2 = \frac{\epsilon}{16n}$, and a negligible function negl.

The full proof is given in Appendix E. We next discuss the ideas behind the proof.

Proof Sketch of Lemma 4.6. Recall that in the traceability game, the adversary provides a distinguishing decoder $\mathcal{D}^{\mathsf{m}_0,\mathsf{m}_1}$ that can distinguish encryptions of m_0 from encryptions of m_1, per Definition 4.2. Let p_i for $0 \leq i \leq n-t$ be the distinguishing advantage of $\mathcal{D}^{\mathsf{m}_0,\mathsf{m}_1}$ when it is queried with the ciphertext $\mathsf{ct} = \mathsf{TrEnc}(\mathsf{m}_b, \mathcal{R}_i)$. We show that the probability of winning the traceability game, as defined in Figure 4, is upper bounded by the probability of winning one of the three games of $(\epsilon/16)$-indistinguishability, $(\epsilon/16n)$-membership hiding, or message privacy, where the probability is over the random choice of b from $\{0,1\}$. First, observe that the adversary wins the traceability game only in the following two cases:

- Case 1: $\mathcal{T}' = \emptyset$. This happens only if $\hat{p}_{i-1} - \hat{p}_i < \frac{\epsilon}{4(n-t)}$ for any $i \in [n-t]$ (Condition 1 in Π_{TSTE}) and $\hat{p}_{n-t} < \frac{\epsilon}{4}$ (Condition 2 in Π_{TSTE}).
- Case 2: $\mathcal{T}' \neq \emptyset$ and $\mathcal{T}' \not\subseteq T$. This happens only if $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ for some honest party $i \in [n-t]$ or when $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$ and at least one of the parties $\{n-t+1, \ldots, n\}$ is honest.

We show that if Case 1 holds, then $\hat{p}_0$ should be much smaller than ϵ, that is, $\epsilon - \hat{p}_0 > \epsilon/4$. Indeed, if $\hat{p}_0 \geq \frac{3\epsilon}{4}$, then either for at least one i it must hold that $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ or, otherwise, it holds that $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$, so Case 1 will not hold. Hence, we show that if $\epsilon - \hat{p}_0 > \frac{\epsilon}{4}$ (i.e., Case 1 holds), then the adversary can construct a distinguishing decoder in the indistinguishability game with advantage $\delta_1 = \epsilon/16$. To show this, we first show by Chernoff bound that $p_0 - \hat{p}_0 < \frac{\epsilon}{8}$ with overwhelming probability. Thus, $\epsilon - p_0 > \frac{\epsilon}{8}$. The distinguishing decoder works as follows. Given a ciphertext ct, which is either $\mathsf{Enc}(\mathsf{m}_b)$ or $\mathsf{TrEnc}(\mathsf{m}_b, \mathcal{R}_0)$, it returns 0 if $\mathcal{D}^{\mathsf{m}_0,\mathsf{m}_1}(\mathsf{ct}) = b$ and 1 otherwise. Observe that the distinguishing decoder wins with advantage ϵ if ct is encrypted using Enc and with advantage $1 - p_0$ if ct is encrypted using $\mathsf{TrEnc}(\mathcal{R}_0)$ (since in this case, it returns 1 if and only if $\mathcal{D}$ fails). Therefore, the distinguishability advantage of the distinguishing decoder is $\frac{1}{2}\epsilon + \frac{1}{2}(1 - p_0) \geq \frac{\epsilon}{16}$, as required.

Next, we show that if Case 2 holds, then:

- If $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$ holds, then the adversary can break the message privacy of the scheme. Indeed, since in this case, the adversary can distinguish between $\mathsf{TrEnc}(\mathsf{m}_0, \mathcal{R}_{n-t})$ and $\mathsf{TrEnc}(\mathsf{m}_1, \mathcal{R}_{n-t})$ with less than t parties (since the first $n-t$ parties are revoked and, according to the condition, not all the remaining t parties are colluders).
- If $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ for some honest party, then the adversary can construct a distinguishing decoder in the membership hiding game with distinguishing advantage $\frac{\epsilon}{16(n-t)} > \frac{\epsilon}{16n}$. To

see this, we first show using the Chernoff bound that $\hat{p}_i - p_i < \frac{\epsilon}{16(n-t)}$ with overwhelming probability. Thus, $p_{i-1} - p_i \geq \frac{\epsilon}{8(n-t)}$ with overwhelming probability. The distinguishing decoder works as follows. Given a ciphertext ct, which is either $\mathsf{TrEnc}(\mathsf{m}_b, \mathcal{R}_{i-1})$ or $\mathsf{TrEnc}(\mathsf{m}_b, \mathcal{R}_i)$ for $b \in \{0,1\}$, it outputs 0 if $\mathcal{D}^{\mathsf{m}_0,\mathsf{m}_1}(\mathsf{ct}) = b$ and 1 otherwise. Then, the distinguishing decoder wins with probability $\frac{1}{2}p_{i-1} + \frac{1}{2}(1-p_i) \geq \frac{\epsilon}{16(n-t)} > \frac{\epsilon}{16n}$. This holds since the probability to output 0 when ct is encrypted to $\mathcal{R}_{i-1}$ is exactly the advantage of the decoder $\mathcal{D}^{\mathsf{m}_0,\mathsf{m}_1}$ in decrtpting ct, which, by definition, happens with probability p_{i-1}. On the other hand, when ct is encrypted to $\mathcal{R}_i$, the distinguishing decoder returns 1 only when the decoder $\mathcal{D}$ fails to decrypt, which, by definition, happens with probability $1 - p_i$.

4.2 k-Traceability

We next discuss how our Π_{TSTE} can be extended to achieve k-traceability, that is, catching k traitors for some $1 \leq k \leq t$. The idea relies on the fact that in our tracing algorithm Trace in Figure 5, the tracer executes at most $n - t$ iterations, where in the i-th iteration, it revokes i parties. Therefore, at the end of the tracing, at least t parties are never revoked in any of the previous iterations. Hence, by the nature of the tracing algorithm, if a traitor is found due to Condition 1, it cannot be any of these t parties. In the rest of the discussion, w.l.g, we assume that a single traitor is returned in each execution of Trace (if we hit Condition 2, then t traitors are returned anyway). For a given k, we execute Trace k times with respect to different revocation sets. That is, let $\mathcal{T}_i$ be the set of i traitors that were detected after the i-th execution, then in the $(i+1)$-th execution, we run tracing for revocation sets that do not revoke any party in $\mathcal{T}$. That is, we use the set of revocation sets $S_{\mathcal{R}} = \{\emptyset, \{j_1\}, \{j_1, j_2\}, \ldots, \{j_1, \ldots, j_{n-t}\}\}$, which is induced by the revocation order $[j_1, \ldots, j_{n-t}]$, such that $j_v \neq \ell$ for every $\ell \in \mathcal{T}_i$ and every $1 \leq v \leq n - t$. Hence, by the discussion above, in the $(i+1)$-th execution, we get a traitor that is not in $\mathcal{T}_i$. In the end, after (at most) k executions, we get k traitors. To achieve this, however, we require the underlying OSTE to support all revocation sets used in all executions, i.e., all these revocation sets must be included in $\mathcal{X}$, as per Definition 3.4. We call such $\mathcal{X}$ a *k-trace set*.

Theorem 4.7 *Let $k \in \mathbb{N}$ such that $1 \leq k \leq t \leq n$. If there exists an OSTE scheme that fulfills $(\epsilon/16n)$-membership hiding and $(\epsilon/16)$-indistinguishability for a k-trace set $\mathcal{X}$, then there exists a traceable silent setup threshold encryption scheme with threshold t in which the tracing procedure returns at least k traitors.*

As instantiation of Theorem 4.7, in Section 5 we present an OSTE construction that fulfills membership hiding with $\mathcal{X} = \mathcal{P}([n])$, thus tracing t traitors.

5 The First OSTE Construction

In this section, we present our Π_{OSTE} construction. Our construction is based on the silent threshold encryption of [28]. We start by presenting a high-level overview of the construction of [28]. Then, we discuss the modifications we make to achieve the properties of OSTE.

5.1 Starting Point: The STE Scheme of [28]

The Setting. The construction relies on a powers-of-τ trusted setup, i.e., the CRS is $([1]_{1,2}, [\tau]_{1,2}, \ldots, [\tau^n]_{1,2})$, in an asymmetric pairing ensemble. Then, each party $i \in [n]$ samples its secret and public shares $(\mathsf{sk}_i, \mathsf{pk}_i)$, computes a $\mathsf{hint}_i = ([\mathsf{sk}_i\tau]_1, \ldots, [\mathsf{sk}_i\tau^n]_1)$ using the CRS, and publishes pk_i and hint_i. hint_i is an auxiliary information used during encryption and decryption. From $\{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}$, each entity can compute an encryption key $\mathsf{ek} = \sum_{i\in[0,n]}[\mathsf{sk}_i L_i(\tau)]_1$. Furthermore,

an aggregation key ak that corresponds to ek is computed from the hints. Then, using ek, anyone can encrypt a message that is decrypted using t secret shares and ak. Note that both ek and ak are public.

The Intuition. The idea follows that of the Boneh-Franklin identity-based encryption scheme [6], which is essentially a witness-encryption to the statement "I know a BLS signature on the identity" [7]. In particular, in [28], the decryption key is basically a witness that the decryptor knows σ^* and aPK such that σ^* is verified as $[\gamma]_2 \circ \mathsf{aPK} = [1]_1 \circ \sigma^*$, where σ^* is an aggregation of at least t partial signatures for a set of signers $S = \{i_1, \ldots, i_t\}$, where $i_j \in [n]$, and aPK is an aggregated public key that combines $\{\mathsf{pk}_0, \mathsf{pk}_{i_1}, \ldots, \mathsf{pk}_{i_t}\}$.

Linear Constraints. In order to compute a witness for knowing such an aPK, the aggregator computes polynomials B, Q_x, and Q_Z that fulfill the sumcheck equation (Lemma 2.1), as follows:

$$\mathsf{ek} \circ [B(\tau)]_2 = \mathsf{aPK} \circ [1]_2 + [Q_x(\tau)]_1 \circ [\tau]_2 + [Q_Z(\tau)]_1 \circ [Z(\tau)]_2 \tag{1}$$

Notably, B is the polynomial extension of the set vector (i.e., $B(\omega^i) = 0$ where $i \notin S$); all evaluations can be computed using the hints and the CRS. This proves that if σ^* is verified, then aPK is indeed an inner product of ek and the set S, which means that aPK is an aggregation of the public keys in S. Additionally, three critical constraints need to be verified linearly: First, that B represents an authorized subset. That is, the polynomial B is non-zero in at least t positions, which is achieved by a degree check that $\deg(B) \leq n - t$. Second, the degree of Q_x is at most $|\mathbb{H}| - 2$, which is a requirement of the sumcheck lemma. Third, that $B(\omega^0) = 1$ (including a dummy party) to ensure that the trivial solution of setting B to the zero-polynomial is not valid. For notation, we henceforth assume that $\mathsf{sk}_0 = 1$ for the dummy party. We summarize how to verify these constraints linearly in Appendix F. Each one of the restrictions is represented as a linear equation in the matrix A, which we present next.

Encryption and Decryption. The encryption proceeds as follows:

- Given a message $m \in \mathbb{G}_T$, sample a random tag $\gamma \in \mathbb{Z}_p$ and compute:

$$A = \begin{pmatrix} \mathsf{ek}(\tau) & [1]_2 & [Z(\tau)]_2 & [\tau]_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & [\tau]_2 & [1]_2 & 0 & 0 & 0 \\ 0 & [\gamma]_2 & 0 & 0 & 0 & [1]_1 & 0 & 0 \\ [\tau^t]_1 & 0 & 0 & 0 & 0 & 0 & [1]_2 & 0 \\ [1]_1 & 0 & 0 & 0 & 0 & 0 & 0 & [\tau - \omega^0]_2 \end{pmatrix} \begin{array}{l} /\!/\ \mathsf{aPK} \text{ valid (sumcheck)} \\ /\!/\ \deg(Q_x) \leq |\mathbb{H}| - 2 \\ /\!/\ \sigma^* \text{ is valid} \\ /\!/\ B \text{ is authorized} \\ /\!/\ \text{dummy party} \end{array}$$

  Note that all the entries in A are fixed (the same for all ciphertexts) except for $[\gamma]_2$, which must be unique for every ciphertext.

- Sample $s = (s_1, \ldots, s_5) \xleftarrow{\$} \mathbb{Z}_p^5$ and compute:

$$\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \mathsf{ct}_3) = ([\gamma]_2, s^\intercal \cdot A, s^\intercal \cdot b + \mathsf{m}),$$

  where $b = ([0]_T, [0]_T, [0]_T, [0]_T, [1]_T)^\intercal$.

For decryption, the aggregator collects partial decryptions from a set of parties $S \subseteq [n]$, where $|S| \geq t$. The partial signature of the i-th party is simply $\sigma_i \leftarrow \mathsf{sk}_i \cdot [\gamma]_2$. Then, the aggregator computes

$$\begin{aligned} \mathsf{m} &= \mathsf{ct}_3 - \mathsf{ct}_2 \circ w \\ &= [s_5]_T + \mathsf{m} - s^\intercal A \circ w \\ &= [s_5]_T + \mathsf{m} - [s_5]_T = \mathsf{m}, \end{aligned}$$

where w is the witness vector such that $A \circ w = b$. Importantly, such w can be computed using the aggregated key ak, and it guarantees that all constraints in A are fulfilled.

5.2 Our OSTE Construction

We first introduce a construction with ciphertext size linear in n and then present a technique in Section 5.3 to amortize the ciphertext size to $O(1)$.

To construct an OSTE scheme, we need a mechanism to encrypt a ciphertext such that membership hiding holds. In the STE construction of [28], the aggregation key ak alone clearly reveals which parties can participate in decryption. Let us inspect the encryption key ek again: $\mathsf{ek} = \sum_{i\in[0,n]}[\mathsf{sk}_i L_i(\tau)]_1$. Assume that we need to revoke a set $\mathcal{R} \subseteq [n]$ (the dummy party is never revoked), then the first attempt would be to randomize the public share of each party $j \in \mathcal{R}$ and use the new pk'_j for computing ek instead of $[\mathsf{sk}_j]_1$ (the old public share). Intuitively, any party $j \in \mathcal{R}$ cannot participate in the decryption since it does not hold the *new* secret share corresponding to the new public share pk'_j. However, to ensure correctness, we note that it is necessary to accommodate the new public shares in the computation of the aggregation key ak. Unfortunately, this requires revealing which public keys were randomized; hence, membership hiding would not hold.

To solve this issue, we propose to rerandomize *all* public keys instead of only for the parties in $\mathcal{R}$. That is, we sample a random field element α and a vector $\bar{\alpha} = (\bar{\alpha}_j)_{j\in\mathcal{R}}$ and rerandomize all *unrevoked* parties' public shares with α and that of each parties $j \in \mathcal{R}$ with $\bar{\alpha}_j$. We adjust the hints accordingly and publish the resulting aggregation key ak along with the ciphertext. This rerandomization solves the membership hiding issue because without knowing $\alpha, \bar{\alpha}_i$ and sk_i, but only $[\alpha]_1$ and $[\mathsf{sk}_i]_1$, adversaries are unable to determine if the rerandomized public key pk'_i is equal to $\alpha \cdot \mathsf{pk}_i$ or $\bar{\alpha}_i \cdot \mathsf{pk}_i$ (we will reduce this to the XDH problem to prove membership hiding in Lemma 5.3). For correctness, the signature verification equation is now verified against $[\alpha]_1$ instead of $[1]_1$. Thus, partial signatures from unrevoked parties will be accepted as their public keys are randomized with α. For message privacy, we show that an aggregator cannot include a partial signature from a revoked party in $\mathcal{R}$. Informally, this holds since verifying the aggregated signature against $[\alpha]_1$ requires the aggregated public key $\widehat{\mathsf{aPK}}$ to be $\alpha \cdot \mathsf{aPK}$ (where aPK is the aggregated key before the randomization), and thus the aggregation can only include keys that are masked by α.

Finally, we highlight that we obtain a linear ciphertext size in the worst case, but as we discuss in Section 5.3, the amortized size is $O(1)$. Furthermore, we elaborate on reducing the ciphertext size in Section 6.

5.2.1 The Π_{OSTE} Construction

Our construction relies on a rerandomization algorithm (Figure 6) that realizes the rerandomization idea as discussed above. This algorithm is used in Enc and TrEnc to generate a new encryption key, which is then used to encrypt the message in a similar way to STE (as discussed in Section 5.1). The Enc and TrEnc algorithms are depicted in Figure 7. Partial decryption works as in [28] by computing $\sigma_i \leftarrow \mathsf{sk}_i \cdot [\gamma]_2$. Aggregation of partial decryption shares is also unchanged compared to [28], but the aggregator now uses the rerandomized aggregation key ak to compute aPK and evaluate the polynomials Q_x, Q_Z and other components of the witness w. As such, the aggregation is performed as if $\mathsf{sk}'_i = \alpha\mathsf{sk}_i$ for unrevoked parties and $\mathsf{sk}'_i = \bar{\alpha}_i\mathsf{sk}_i$ for revoked parties. We present all algorithms omitted here in Appendix G.

We note that the rerandomization algorithm Rand supports arbitrary revoked sets $\mathcal{R} \in \mathcal{P}([n])$, which in turn allows us to use Π_{OSTE} to guarantee tracing t traitors (see Theorem 4.7). Our OSTE construction supports public tracing and does not have any preprocessing, so we omit the tracing key tk and set the preprocessing to $\mathcal{K} := \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}$. We next discuss the properties of the construction.

$\mathsf{Rand}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{R})$

$\alpha \xleftarrow{\$} \mathbb{Z}_p^*$

$\forall i \in \mathcal{R}\colon \bar{\alpha}_i \xleftarrow{\$} \mathbb{Z}_p^*$

Let $r_i \leftarrow \alpha \ \ \forall i \in [0,n] \setminus \mathcal{R}$ // Randomize with α for unrevoked parties.

Let $r_i \leftarrow \bar{\alpha}_i \ \ \forall i \in \mathcal{R}$ // Randomize with $\bar{\alpha}_i$ for revoked parties i.

for $i = 0, \ldots, n$ **do**

$\mathsf{ak}_{i,0} \leftarrow r_i \mathsf{pk}_i$

$\mathsf{ak}_{i,1} \leftarrow \left[r_i \mathsf{sk}_i (L_i(\tau) - L_i(0))\right]_1$

$\mathsf{ak}_{i,2} \leftarrow \left[r_i \mathsf{sk}_i \dfrac{L_i(\tau) - L_i(0)}{\tau}\right]_1$

$\mathsf{ak}_{i,3} \leftarrow \left[r_i \mathsf{sk}_i \dfrac{L_i(\tau)^2 - L_i(\tau)}{Z(\tau)}\right]_1$

$\mathsf{ak}_{i,4} \leftarrow \left[\sum_{j\in[n], j\neq i} r_j \mathsf{sk}_j \dfrac{L_i(\tau) L_j(\tau)}{Z(\tau)}\right]_1$

$\mathsf{ek} \leftarrow \sum_{i\in[0,n]} [r_i \mathsf{sk}_i L_i(\tau)]_1$

return $([\alpha]_1, \mathsf{ek}, \mathsf{ak} := \{(\mathsf{ak}_{i,0}, \ldots, \mathsf{ak}_{i,4})\}_{i\in[0,n]})$

Figure 6: The rerandomization algorithm. Note that all group elements can be computed using the powers-of-τ CRS and hints. The ak and ek are the same as in [28] with the rerandomization highlighted in gray.

Correctness. To see that correctness is not violated, consider how the aggregator combines t partial decryptions to decrypt a ciphertext with respect to the rerandomized public keys and hints. First, the aggregator computes an aggregated key $\widehat{\mathsf{aPK}} = \alpha \cdot \mathsf{aPK}$, which is consistent with the new public keys. Then, it can compute the polynomials Q_x and Q_Z that satisfy Equation 1 since the new public keys are consistent with the ek used to encrypt the message. To satisfy the third restriction (i.e., the aggregated signature σ^* is valid), we modify the signature verification equation to

$$[\alpha]_1 \circ [\sigma^*]_2 = \widehat{\mathsf{aPK}} \circ [\gamma]_2$$

instead of the original BLS verification. To enforce this, the encryptor replaces the $[1]_1$ in $A_{2,5}$ with $[\alpha]_1$. Hence, correctness follows from the discussion above, and the correctness of the STE construction of Garg et al. [27].

Before establishing the security of our Π_{OSTE} construction, we address a rushing adversary attack that is made possible by the silent setup setting. In this attack, the adversary selects its own public key based on those of honest parties to break membership hiding.

Rushing Adversary. The construction, as discussed above, is subject to a rushing adversary attack in the membership hiding game GameMH (Figure 2). To see this, observe that an adversary could choose the secret key for a corrupted party j dependent on an honest party's public key pk_i. Then, it can distinguish whether party i is revoked as follows. After receiving pk_i and hint_i, the adversary could publish $\mathsf{pk}_j = x \cdot \mathsf{pk}_i$ and $\mathsf{hint}_j = x \cdot \mathsf{hint}_i$ for an arbitrary $x \in \mathbb{Z}_p^*$ with $x \neq 1$, and choose, e.g., $\mathcal{R} = \emptyset$. After the adversary receives a ciphertext $\mathsf{ct} = ([\gamma]_2, \mathsf{ak}, \ldots)$, it reveals if i is revoked by the simple check that $\mathsf{ak}_{j,0} = x \cdot \mathsf{ak}_{i,0}$. First, to see why this check is

$\mathsf{Enc}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, t, \mathsf{m})$

$([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \xleftarrow{\$} \mathsf{Rand}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \emptyset)$

return $\mathsf{enc}(\mathsf{CRS}, [\alpha]_1, \mathsf{ek}, \mathsf{ak}, t, \mathsf{m})$

$\mathsf{TrEnc}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, t, \mathsf{m}, \mathcal{R})$

$([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \xleftarrow{\$} \mathsf{Rand}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{R})$

return $\mathsf{enc}(\mathsf{CRS}, [\alpha]_1, \mathsf{ek}, \mathsf{ak}, t, \mathsf{m})$

$\mathsf{enc}(\mathsf{CRS}, [\alpha]_1, \mathsf{ek}, \mathsf{ak}, t, \mathsf{m})$

$\gamma \xleftarrow{\$} \mathbb{Z}_p$

Assemble the matrix A with the following values:

$A_{0,0} \leftarrow \mathsf{ek}$

$A_{2,1} \leftarrow [\gamma]_2$

$A_{2,5} \leftarrow [\alpha]_1$

$s := (s_1, \ldots, s_5) \xleftarrow{\$} \mathbb{Z}_p^5$

$\mathsf{ct} \leftarrow ([\gamma]_2, \mathsf{ak}, s^\intercal \cdot A, s_5 + \mathsf{m})$

return ct

Figure 7: The Enc and TrEnc algorithms of our OSTE construction. enc acts as a helper algorithm to both Enc and TrEnc. In enc, we refer to matrix A from Section 5.1.

sufficient, note that if i is not revoked, $\mathsf{ak}_{i,0} = \alpha \cdot \mathsf{pk}_i$. Then, as party j is not revoked ($j \notin \mathcal{R}_0$), it must hold that $\mathsf{ak}_{j,0} = \alpha \cdot \mathsf{pk}_j = \alpha \cdot x \cdot \mathsf{pk}_i$.

To mitigate this attack, we require each party to attach a NIZK proof of knowledge of their corresponding secret key sk to their public key $(\mathsf{pk}, \mathsf{hint})$. We further require that all parties have distinct public keys to avoid replay attacks of these proofs. Following Definition A.1, we formally define a proof system $\mathsf{PS}_{\mathsf{KGen}}$ through the relation $\mathsf{R}_{\mathsf{KGen}}$ of statements $\chi = (\mathsf{CRS}, \mathsf{pk}, \mathsf{hint})$ and witnesses $w = \mathsf{sk} \in \mathbb{Z}_p^*$, such that $(\chi, w) \in \mathsf{R}_{\mathsf{KGen}}$ if $\mathsf{pk} = [\mathsf{sk}]_1$ and $\mathsf{hint} = ([\mathsf{sk}\tau]_1, \ldots, [\mathsf{sk}\tau^n]_1)$.

The following theorem summarizes our result for Π_{OSTE}.

Theorem 5.1 *The OSTE scheme Π_{OSTE} as described in Section 5.2 fulfills correctness, δ-indistinguishability, message privacy, δ'-membership hiding, and IND-CPA security for $n = \mathsf{poly}(\lambda)$ and any $\delta, \delta' \in [-1/2, 1/2]$, where $\delta, \delta' = 1/\mathsf{poly}(\lambda)$. It has ciphertext size $|c| = O(n)$ and amortized ciphertext of size $O(1)$.*

We defer the discussion on the amortization to Section 5.3. We next discuss the security of the construction. Indistinguishability is trivial, as Enc and $\mathsf{TrEnc}(\mathcal{R} = \emptyset)$ are identical. In the following, we discuss message privacy and membership hiding.

Lemma 5.2 (Message Privacy of Π_{OSTE}). *For $n = \mathsf{poly}(\lambda)$ and $\mathcal{X} = \mathcal{P}([n])$, Π_{OSTE} fulfills message privacy (Definition 3.5), if $\mathsf{PS}_{\mathsf{KGen}}$ is zero-knowledge (Definition A.1).*

For message privacy, we prove that an adversary who corrupts less than t *unrevoked* parties cannot decrypt a ciphertext. Our proof is carried out in the GGM, using the master theorem (Theorem A.3). Intuitively, message privacy holds since the signature verification is done against $[\alpha]_1$, and the aggregated public key $\widehat{\mathsf{aPK}}$ needs to be consistent with ek to fulfill the sumcheck restriction. Hence, $\widehat{\mathsf{aPK}}$ must equal $\alpha\mathsf{aPK}$ (where aPK is the aggregation before the randomization). Thus, aPK cannot use public keys not masked with α (i.e., keys of revoked parties). If the adversary tries to use such keys, it will need to find a linear combination such that the sum of these keys in $\widehat{\mathsf{aPK}}$ is zero. To find such a combination without knowing $\bar{\alpha}_i$, the adversary must fix points on the polynomial B, resulting in a degree larger than $n - t$. So, the degree check of B would not pass. We give a full proof of Lemma 5.2 in Appendix H.

We note that for the Π_{OSTE} construction IND-CPA-security can be considered as a special case of message privacy, where $\mathcal{R} = \emptyset$, because we define $\mathsf{Enc}(\cdot) = \mathsf{TrEnc}(\cdot, \mathcal{R} = \emptyset)$ for Π_{OSTE}. In

addition, the fact that the adversary is trusted for preprocessing in IND-CPA is irrelevant for Π_{OSTE}. Therefore, message privacy implies IND-CPA security of Π_{OSTE}.

Lemma 5.3 (Membership Hiding of Π_{OSTE}). *Let $n = \mathsf{poly}(\lambda)$. Our Π_{OSTE} construction, as described in Section 5.2, is membership hiding for arbitrary $\delta = \frac{1}{\mathsf{poly}(\lambda)}$ with $\delta \in [-1/2, 1/2]$ if the XDH assumption (Definition 2.2) holds and $\mathsf{PS}_{\mathsf{KGen}}$ is zero-knowledge and simulation-extractable (Definition A.1).*

At its core, our membership-hiding proof works by embedding an XDH challenge $[x]_1, [y]_1, T$ into the rerandomization for a ciphertext ct such that x corresponds to sk_i, and y corresponds to α. If the challenge $T = [xy]_1 \equiv [\alpha \mathsf{sk}_i]_1$, then party i is not revoked for ct and if $T = [v]_1$ for a random $v \equiv \bar{\alpha}_i \cdot \mathsf{sk}_i$, then party i is revoked for ct. We query the decoder on ct and a reference ciphertext, where we randomly choose to revoke i or not. We compare the results to determine whether $T = [xy]_1$ or $T = [v]_1$.

Proof. Intuitively, our OSTE construction is membership hiding because an adversary that only learns $[\alpha]_1$ and $[\mathsf{sk}_i]_1$ for some i should be unable to distinguish between $[\alpha \cdot \mathsf{sk}_i]_1$ (reflecting the case where party i is not revoked by TrEnc) and $[\bar{\alpha}_i \cdot \mathsf{sk}_i]_1$ (representing the case where party i is revoked), which is reminiscent of the XDH problem in bilinear groups (Definition 2.2). The same argument applies to the re-randomized ak.

We formally prove this through a series of game hops with a final reduction to the XDH assumption in $\mathbb{G}_1$. Let Game_0 be the membership hiding experiment for the Π_{OSTE} construction $\mathsf{GameMH}^{\Pi_{\mathsf{OSTE}}}$ (Figure 2). We proceed with the first game hop, where we simulate the proof π_i on behalf of the honest party.

$\underline{\mathsf{Game}_1}$: In this game, we modify how the challenger computes the proof π_i that is attached to pk_i and hint_i on behalf of the honest party at position i. In particular, we replace π_i by a simulated proof as $\pi_i \xleftarrow{\$} \mathsf{Sim}(\mathsf{td}, \mathsf{pk}_i, \mathsf{hint}_i)$, where Sim is the simulator for the proof system $\mathsf{PS}_{\mathsf{KGen}}$.

Claim 5.4 If the proof system $\mathsf{PS}_{\mathsf{KGen}}$ is zero-knowledge (Definition A.1), then Game_0 and Game_1 are computationally indistinguishable.

Proof. The computational indistinguishability follows directly from the computational indistinguishability of a real proof and a simulated proof as stated by the zero-knowledge property of $\mathsf{PS}_{\mathsf{KGen}}$. ∎

In our final reduction to XDH, we will need to know the secret keys that are chosen by the adversary for the parties at positions $[n] \setminus \{i\}$ so we can properly embed the XDH challenge into their re-randomized public keys and hints. We use the simulation-extractability of $\mathsf{PS}_{\mathsf{KGen}}$ to extract sk_ℓ from π_ℓ for all $\ell \in [n] \setminus \{i\}$.

$\underline{\mathsf{Game}_2}$: In Game_2, we use the extractor of $\mathsf{PS}_{\mathsf{KGen}}$ to extract $\{\mathsf{sk}_\ell\}_{\ell \in [n] \setminus \{i\}}$ from the adversary's proof as $\mathsf{sk}_\ell \xleftarrow{\$} \mathsf{PS}_{\mathsf{KGen}}.\mathsf{Ext}(\mathsf{td}, (\mathsf{CRS}, \mathsf{pk}_\ell, \mathsf{hint}_\ell), \pi_\ell)$. We add an additional abort condition, where Game_2 outputs 0, if there exists an ℓ such that $\mathsf{pk}_\ell \neq [\mathsf{sk}_\ell]_1$ or $\mathsf{hint}_\ell \neq ([\mathsf{sk}_\ell \tau]_1, \ldots, [\mathsf{sk}_\ell \tau^n]_1)$.

Claim 5.5 For all $n = \mathsf{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ such that

$$\left|\Pr[\mathsf{Game}_{1,\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta) = 1] - \Pr[\mathsf{Game}_{2,\mathcal{B}}(1^\lambda, n, \mathcal{X}, \delta) = 1]\right| \\ \leq (n-1) \cdot \mathbf{Adv}^{\text{sim-extract}}_{\mathcal{B}, \mathsf{PS}_{\mathsf{KGen}}}(1^\lambda).$$

Proof. First, observe that for all $\mathcal{A}$ it holds that

$$\Pr[\mathsf{Game}_{1,\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta) = 1] \geq \Pr[\mathsf{Game}_{2,\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta) = 1],$$

because we only add an additional abort condition. Let $\Pr[\mathsf{Game}_{1,\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta) = 1] - \Pr[\mathsf{Game}_{2,\mathcal{A}}(1^\lambda, n, \mathcal{X}, \delta) = 1] = \varepsilon$. Then, with probability ε, the adversary $\mathcal{A}$ hits the additional abort condition in Game_2. We construct a PPT reduction $\mathcal{B}$ that runs in the simulation-extractability game of $\mathsf{PS}_{\mathsf{KGen}}$. Given $\mathsf{CRS}_{\mathsf{PS}_{\mathsf{KGen}}}$, $\mathcal{B}$ embeds it in the global CRS and simulates Game_2 to $\mathcal{A}$ as follows. $\mathcal{B}$ runs $\mathcal{A}(1^\lambda, \mathsf{CRS})$ and receives $i \in [n]$. Then, the reduction samples a random $(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \_) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, \mathsf{CRS})$ and simulates the proof π_i using the $\mathsf{Sim}(\mathsf{td}, (\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i))$ oracle provided by the simulation-extractability game and sends $(\mathsf{pk}_i, \mathsf{hint}_i, \pi_i)$ to $\mathcal{A}$. In return $\mathcal{B}$ receives $\{\mathsf{pk}_\ell, \mathsf{hint}_\ell, \pi_\ell\}_{\ell \in [n] \setminus \{i\}}$. $\mathcal{B}$ chooses a random $\ell^* \in [n] \setminus \{i\}$ and outputs $(\chi^*, \pi^*) = ((\mathsf{CRS}, \mathsf{pk}_{\ell^*}, \mathsf{hint}_{\ell^*}), \pi_{\ell^*})$ to the simulation-extractability game.

Observe that $\chi^* = (\mathsf{CRS}, \mathsf{pk}_{\ell^*}, \mathsf{hint}_{\ell^*}) \notin Q = \{(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i)\}$[2], whenever either $\mathsf{Game}_{1,\mathcal{A}}$ or $\mathsf{Game}_{2,\mathcal{A}}$ would output 1 because of the check in GameMH that $\mathsf{pk}_i \neq \mathsf{pk}_\ell$ for all $i \neq \ell$. Further, it holds that with probability ε there exists an $\ell' \in [n] \setminus \{i\}$ such that $(\chi_{\ell'}, \mathsf{sk}_{\ell'}) \notin \mathsf{R}_{\mathsf{KGen}}$. $\mathcal{B}$ chooses $\ell^* = \ell'$, hence $\mathcal{B}$ wins the simulation-extractability game with probability at least $\varepsilon/(n-1)$. ∎

$\mathsf{SimRand}([x]_1, [y]_1, T, \tau, \{\mathsf{sk}_\ell\}_{\ell \in [n] \setminus \{i\}}, \mathcal{R}, i)$

Sample $\bar{\alpha}_\ell \xleftarrow{\$} \mathbb{Z}_p^* \; \forall \ell \in \mathcal{R}$

for $\ell \in [0, n] \setminus \mathcal{R}$ with $\ell \neq i$ **do**

$\mathsf{ak}_{\ell,0} \leftarrow \mathsf{sk}_\ell \cdot [y]_1 \quad (= [\alpha \mathsf{sk}_\ell]_1)$

$\mathsf{ak}_{\ell,1} \leftarrow \mathsf{sk}_\ell (L_\ell(\tau) - L_\ell(0)) \cdot [y]_1 \quad (= [\alpha \mathsf{sk}_\ell (L_\ell(\tau) - L_\ell(0))]_1)$

$\mathsf{ak}_{\ell,2} \leftarrow \mathsf{sk}_\ell \dfrac{L_\ell(\tau) - L_\ell(0)}{\tau} \cdot [y]_1$

$\mathsf{ak}_{\ell,3} \leftarrow \mathsf{sk}_\ell \dfrac{L_\ell(\tau)^2 - L_\ell(\tau)}{Z(\tau)} \cdot [y]_1$

$\mathsf{ak}_{\ell,4} \leftarrow \displaystyle\sum_{j \in [0,n] \setminus \mathcal{R}, j \neq \ell, j \neq i} \mathsf{sk}_j \frac{L_\ell(\tau) L_j(\tau)}{Z(\tau)} \cdot [y]_1 + \sum_{j \in \mathcal{R}} \bar{\alpha}_j \mathsf{sk}_j \frac{L_\ell(\tau) L_j(\tau)}{Z(\tau)} + \frac{L_\ell(\tau) L_i(\tau)}{Z(\tau)} \cdot T$

for $\ell \in \mathcal{R}$ **do**

$\mathsf{ak}_{\ell,0} \leftarrow [\bar{\alpha}_\ell \mathsf{sk}_\ell]_1$

$\ldots$

$\mathsf{ak}_{i,0} \leftarrow T$

$\mathsf{ak}_{i,1} \leftarrow (L_\ell(\tau) - L_\ell(0)) \cdot T$

$\ldots$

$\mathsf{ek} \leftarrow \displaystyle\sum_{\ell \in [0,n] \setminus \mathcal{R}, \ell \neq i} \mathsf{sk}_\ell L_\ell(\tau) [y]_1 + \sum_{\ell \in \mathcal{R}} [\bar{\alpha}_\ell \mathsf{sk}_\ell L_\ell(\tau)]_1 + L_i(\tau) \cdot T$

return $([y]_1, \mathsf{ek}, \mathsf{ak})$

Figure 8: The simulation of the rerandomization algorithm for the challenge ciphertext, where α corresponds to y. Note that the reduction can compute all values because it knows τ and $\{\mathsf{sk}_\ell\}_{\ell \in [n] \setminus \{i\}}$.

[2]With Q, we refer to the set of statements that the reduction requests a simulated proof for (see Definition A.1).

We conclude our proof with a reduction from Game_2 to the XDH assumption (Definition 2.2). Let $\mathcal{A}$ be a PPT adversary against Game_2. We construct a PPT reduction $\mathcal{B}$ that internally uses $\mathcal{A}$ to break the XDH assumption. Let b_{XDH} be the internal bit of the XDH experiment. Initially, $\mathcal{B}$ receives an XDH challenge $([x]_1, [y]_1, T)$, where either $T = [xy]_1$ for $(b_{\mathsf{XDH}} = 0)$ or $T = [v]_1$ for $(b_{\mathsf{XDH}} = 1)$.

First, $\mathcal{B}$ samples $\tau \xleftarrow{\$} \mathbb{Z}_p$ and generates the CRS. It queries $\mathcal{A}(1^\lambda, \mathsf{CRS})$ to receive $i \in [n]$. $\mathcal{B}$ now embeds the XDH challenge into the public key and hint that it generates on behalf of party i. In particular, $\mathcal{B}$ sets $\mathsf{pk}_i \leftarrow [x]_1$ and $\mathsf{hint}_i \leftarrow (\tau[x]_1, \ldots, \tau^n[x]_1)$. Observe that, in doing so, $\mathcal{B}$ simulates for $\mathsf{sk}_i = x$ without knowing x itself. After sending $\mathsf{pk}_i, \mathsf{hint}_i$ and the simulated proof π_i to $\mathcal{A}$, the reduction receives $\{\mathsf{pk}_\ell, \mathsf{hint}_\ell\}_{\ell \in [n]\setminus\{i\}}$ and the extracted $\{\mathsf{sk}_\ell\}_{\ell \in [n]\setminus\{i\}}$ as well as the set $\mathcal{R}$, and the threshold t. As the preprocessing for Π_{OSTE} is just equal to the public keys and hints of all parties, $\mathcal{B}$ sends those again to $\mathcal{A}$ and receives a decoder $\mathcal{D}$ as well as a message m. In the following, we write $\mathsf{TrEnc}(\mathcal{R})$ as a shorthand for $\mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk}, \mathcal{K}, t, \mathsf{m}, \mathcal{R})$ as well as $\mathcal{R}^0 = \mathcal{R}$, and $\mathcal{R}^1 = \mathcal{R} \cup \{i\}$.

Next, $\mathcal{B}$ samples a random bit β and computes $\mathsf{ct} \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}_\beta)$. Additionally, $\mathcal{B}$ samples a challenge ciphertext c', embedding the XDH challenge such that, in the rerandomization, α is equivalent to y. Further, T is equivalent to $[\alpha \mathsf{sk}_i]_1$ if $b_{\mathsf{XDH}} = 0$, or $[\bar{\alpha}_i \mathsf{sk}_i]_1$ if $b_{\mathsf{XDH}} = 1$. The details are shown in Figure 8.

Next, $\mathcal{B}$ computes ct' as $\mathsf{enc}(\mathsf{CRS}, [y]_1, \mathsf{ek}, \mathsf{ak}, \mathsf{m})$. Observe that now $\mathsf{ct} = \mathsf{TrEnc}(\mathcal{R}^{b_{\mathsf{XDH}}})$, i.e., if $T = [xy]_1$, then $\mathcal{B}$ simulates $\mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R})$, and if $T = [v]_1$, then $\mathcal{B}$ simulates $\mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R} \cup \{i\})$. Finally, $\mathcal{B}$ outputs $b = \beta$, if $\mathcal{D}(\mathsf{ct}') = \mathcal{D}(\mathsf{ct})$ and $b = 1 - \beta$ otherwise. If $\mathcal{A}$ would lose the game on some technicality, say, duplicate two honest party's public keys, then $\mathcal{B}$ just outputs a random bit.

Let p_w be the probability that $\mathcal{B}$ wins under the condition that $\mathcal{A}$ outputs a $\delta - \mathrm{Dist}^{\mathcal{R},\mathcal{R}\cup i}$ decoder $\mathcal{D}$. Then, the following holds:

$$
\begin{aligned}
p_w =& \Pr\left[b = b_{\mathsf{XDH}} \,\middle|\, \begin{array}{c} \beta \xleftarrow{\$} \{0,1\}; \mathsf{ct} \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^\beta); \\ b_{\mathsf{XDH}} \xleftarrow{\$} \{0,1\}; \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^{b_{\mathsf{XDH}}}); \\ b \leftarrow \beta \text{ if } D(\mathsf{ct}') = D(\mathsf{ct}) \text{ else } b \leftarrow 1 - \beta \end{array} \right] \\
=& \Pr\left[\mathcal{D}(\mathsf{ct}') = b_{\mathsf{XDH}} \,\middle|\, \begin{array}{c} b_{\mathsf{XDH}} \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^{b_{\mathsf{XDH}}}) \end{array} \right] \cdot \Pr\left[\mathcal{D}(\mathsf{ct}) = \beta \,\middle|\, \begin{array}{c} \beta \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^\beta) \end{array} \right] \\
&+ \Pr\left[\mathcal{D}(\mathsf{ct}') \neq b_{\mathsf{XDH}} \,\middle|\, \begin{array}{c} b_{\mathsf{XDH}} \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^{b_{\mathsf{XDH}}}) \end{array} \right] \cdot \Pr\left[\mathcal{D}(\mathsf{ct}) \neq \beta \,\middle|\, \begin{array}{c} \beta \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct} \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^\beta) \end{array} \right] \\
=& \Pr\left[\mathcal{D}(\mathsf{ct}) = \beta \,\middle|\, \begin{array}{c} \beta \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^\beta) \end{array} \right]^2 + \Pr\left[\mathcal{D}(\mathsf{ct}) \neq \beta \,\middle|\, \begin{array}{c} \beta \xleftarrow{\$} \{0,1\}; \\ \mathsf{ct}' \xleftarrow{\$} \mathsf{TrEnc}(\mathcal{R}^\beta) \end{array} \right]^2 \\
=& \left(\frac{1}{2} + \delta\right)^2 + \left(\frac{1}{2} - \delta\right)^2 \\
=& \frac{1}{4} + \delta + \delta^2 + \frac{1}{4} - \delta + \delta^2 \\
=& \frac{1}{2} + 2\delta^2
\end{aligned}
$$

We get the following advantage for $\mathcal{B}$ against XDH.

$$\begin{aligned}
&\frac{1}{2} + \mathbf{Adv}_{\mathcal{B}}^{\mathsf{XDH}}(1^\lambda, G) \\
&\geq \mathbf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, \delta) \cdot p_w + \left(1 - \mathbf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, \delta)\right)\left(\frac{1}{2}\right) \\
&= \mathbf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, \delta)\left(\frac{1}{2} + 2\delta^2\right) + \left(1 - \mathbf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, \delta)\right)\left(\frac{1}{2}\right) \\
&= \frac{1}{2} + \mathbf{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, \delta) \cdot 2\delta^2
\end{aligned}$$

Summarizing, we conclude that for all adversaries $\mathcal{A}$ there exist PPT reductions $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ such that

$$\begin{aligned}
&\mathbf{Adv}_{\mathcal{A}}^{\mathsf{GameMH}^{\Pi_{\mathsf{OSTE}}}}(1^\lambda, n, \mathcal{P}([n]), \delta) \leq \\
&\mathbf{Adv}_{\mathcal{B}_1}^{\mathsf{PS}_{\mathsf{KGen}}\text{-}\mathrm{zk}}(1^\lambda) + (n-1) \cdot \mathbf{Adv}_{\mathcal{B}_2}^{\mathsf{PS}_{\mathsf{KGen}}\text{-}\mathrm{sim\text{-}ext}}(1^\lambda) + \frac{1}{2\delta^2}\mathbf{Adv}_{\mathcal{B}_3}^{\mathsf{XDH}}(1^\lambda, \mathcal{G}).
\end{aligned}$$

■

5.3 Amortization

As we discussed, the ciphertext size in our OSTE is linear in the number of parties. We note that the bottleneck in this construction is the aggregation key ak that is attached to the ciphertext. An observant reader might have noticed that a triple of $([\alpha]_1, \mathsf{ek}, \mathsf{ak})$ could be reused in many ciphertexts. This, however, opens the door for hardcoding attacks, where an adversary hardcodes $([\alpha]_1, \mathsf{ek}, \mathsf{ak})$ into a decoder such that it refuses to decrypt for any other rerandomizations. This decoder can not be traced, as we need to generate new rerandomizations for tracing ciphertexts. We can find a tradeoff between efficiency and traceability guarantees by reusing a single rerandomization for a limited number of ciphertexts. Consequently, one could attach ak only once to the batch of ciphertexts, and as a result, we pay the overhead of $O(n)$ only once per batch. Hence, for a batch of size $O(n)$, we get an amortized size of $O(1)$. Also, note that we can reuse rerandomization without affecting $\mathsf{IND\text{-}CPA}$ security, even if the party who initially samples the rerandomization is malicious (see Section 6). We next discuss a practical setting where this amortization could be beneficial.

Epoch-based Traceable Threshold Decryption. In epoch-based threshold decryption, an epoch public key is generated for each epoch. Then, users can encrypt their messages under the public key of the current epoch. For some applications, the committee decrypts all ciphertexts at the end of the epoch, and the system moves to the next epoch. We note that this model is used in practice, e.g., in a threshold encrypted mempool [13, 18, 25], which is a tool for MEV prevention. This is the primary use case for traceable threshold decryption that motivated the work of [9]. In our setting, we sample and publish a new rerandomization $k_i = ([\alpha_i]_1, \mathsf{ek}_i, \mathsf{ak}_i)$ at the beginning of each epoch. It is crucial that k_i gets revealed only at the beginning of the epoch. Otherwise, if, for example, we publish many epoch keys in advance, the adversary could generate a decoder that decrypts ciphertexts for only a few epochs (by hardcoding the corresponding encryption keys) and refuse to decrypt for others. By publishing the epoch key in the beginning, we can trace any decoder that aims to decrypt ciphertexts for *more than a single epoch*, since such a decoder cannot hardcode the encryption keys of the epochs, as they are published only in the future. The requirement of a decoder that decrypts for more than one single epoch is reasonable

in practice, as decoders that refuse to decrypt for new k_i are no longer useful starting from the next epoch and can be considered somewhat less valuable.

Finally, we need to ensure that whoever samples k_i can not break the IND-CPA security of the scheme. This attack is introduced since the entity running the rerandomization is not the encryptor itself and, therefore, may act maliciously by embedding some corrupted keys into ek and using the corresponding secret keys to decrypt. To circumvent this issue, we can attach a NIZK proof of knowledge of $\mathcal{R}, \alpha$, and $\bar{\alpha}$ such that $([\alpha]_1, \mathsf{ek}, \mathsf{ak})$ is a result of an honest evaluation of the rerandomization algorithm Rand. Further, we can prove that knowledge of α (and $\bar{\alpha}$) does not help the adversary in decrypting any ciphertext that was honestly generated from the corresponding rerandomization. As this technique also appears in the preprocessing approach, we will elaborate on this formally in Section 6. We do not formalize the amortization approach further but instead focus on a preprocessing technique that allows us to get constant size ciphertext in Section 6.

6 Our OSTE Construction based on Preprocessing

In this section, we present our main OSTE construction, denoted by Π_{OSTEP}, in which we add a preprocessing phase to reduce the size of the ciphertext of the Π_{OSTE} construction. In the preprocessing phase, a set of encryption keys is generated and used by all parties, including the tracer. In OSTE with preprocessing, unlike for Π_{OSTE}, using keys other than the preprocessing output is prohibited. We summarize the model using the following definition.

Definition 6.1 (OSTE with Preprocessing (OSTEP)). We say that an OSTE scheme is an OSTE scheme with preprocessing, if the set of possible encryption keys, along with their corresponding aggregation keys, is computed and published in advance during the preprocessing phase after all parties have published their public keys and hints. Further, the encryption keys used in Enc and TrEnc are restricted to these pregenerated encryption keys only.

Let $\mathcal{L} = \{\emptyset, \{1\}, \{1,2\}, \ldots, [n]\}$ be the linear revocation order. Throughout this section, we imply $\mathcal{X} = \mathcal{L}$ when referring to membership hiding and message privacy. The following theorem summarizes the main result of this section.

Theorem 6.2 *The Π_{OSTEP} construction is an OSTEP scheme with $(\frac{\epsilon}{16})$-indistinguishability (per Definition 3.3) and $(\frac{\epsilon}{16n})$-membership hiding with $\mathcal{X} = \mathcal{L}$ (per Definition 3.4) such that the secret share size is 1 group element per party, the public key size is $O(n^4)$, and the ciphertext size is $O(1)$, where $n = \mathsf{poly}(\lambda)$ is the number of parties of the scheme, $\lambda \in \mathbb{N}$ is the security parameter, and $\epsilon \in [-1/2, 1/2]$ such that $\epsilon = 1/\mathsf{poly}(\lambda)$.*

As a corollary of Theorem 4.5 and Theorem 6.2, we achieve a traceable silent threshold encryption scheme (per Definition 4.1).

Remark 6.3 Note that our construction requires knowing the decoder's advantage ϵ in advance. This is referred to as *threshold traitor tracing* [36][3]. Zhandry [42] presented a compiler for eliminating the tracing threshold without affecting the dependence of the parameter sizes on n. We can get rid of fixing ϵ by using this compiler.

6.1 The Π_{OSTEP} Construction

Overview. We denote by $\mathcal{R}_j$ the set of revoked parties up to j ($\mathcal{R}_j = [j]$ and $\mathcal{R}_0 = \emptyset$). Further, we say that a ciphertext ct *revokes* a subset $\mathcal{R}$ if for every $i \in \mathcal{R}$ it holds that the party p_i cannot

[3]This is unrelated to the threshold encryption setting.

$\mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \epsilon)$

$N \leftarrow 1024 \cdot (n+1)\lambda\lambda_c^2 n^2/\epsilon^2$; $\quad d \leftarrow N/(n+1)$

for $j = 0, \ldots, n$ **do**

  for $\ell \in [d]$ **do**

    $([\alpha]_1, \mathsf{ek}, \mathsf{ak}, \pi) \xleftarrow{\$} \mathsf{RandAndProve}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{R}_j)$

    $K_{\text{TR}}[j, \ell] \leftarrow ([\alpha]_1, \mathsf{ek}, \mathsf{ak}, \pi)$

Denote by $K_{\text{TR}}[j]$ the set $\{K_{\text{TR}}[j, \ell]\}_{\ell\in[d]}$ of tracing keys that revoke the set $\mathcal{R}_j$.

for $\ell \in [N]$ **do**

  $([\alpha]_1, \mathsf{ek}, \mathsf{ak}, \pi) \xleftarrow{\$} \mathsf{RandAndProve}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{R}_0)$

  $K_{\text{NORM}}[\ell] \leftarrow ([\alpha]_1, \mathsf{ek}, \mathsf{ak}, \pi)$

$\mathcal{K} \xleftarrow{\$} \mathsf{Shuffle}(K_{\text{NORM}}, K_{\text{TR}})$

return $(\mathcal{K}, \mathsf{tk} \coloneqq (K_{\text{NORM}}, K_{\text{TR}}), \pi \coloneqq \{\pi_\ell\}_{\ell\in[|\mathcal{K}|]})$

Figure 9: The preprocessing of the Π_{OSTEP} construction. The $\mathsf{Shuffle}$ algorithm computes a random permutation. The sets K_{NORM}, K_{TR}, and $\mathcal{K}$ are interpreted as sequences and the $\mathsf{RandAndProve}$ algorithm is as defined above.

be part of the decryption committee of ct. That is, its decryption share for ct will be invalid. Similarly, we say that an encryption key ek revokes a subset $\mathcal{R}$ if every ciphertext ct, encrypted under ek, revokes the set $\mathcal{R}$.

Roughly speaking, we preprocess multiple rerandomizations for different revoked sets. For encryption, the encryption key ek is randomly selected from the preprocessed set of keys. Since ek may correspond to a key that revokes some parties, potentially causing decryption failure for certain coalitions, we encrypt the message under multiple keys to amplify decryption correctness. In tracing, the tracer utilizes dedicated tracing keys from the predefined set. The key challenge in our construction is designing Prep, Enc, and TrEnc that achieve the properties of indistinguishability, membership hiding, and message privacy.

The Π_{OSTEP} construction is depicted in Figures 9 and 10. We next describe the main components of the construction.

Preprocessing. The preprocessing algorithm Prep is shown in Figure 9. In Prep, an entity generates encryption keys using the rerandomization algorithm introduced in Section 5.2 (Figure 6). We generate N encryption keys (along with their corresponding aggregation keys) that are intended for tracing, denoted by K_{TR}, where for every $0 \le j \le n$, we generate $N/(n+1)$ keys that revoke the set $\mathcal{R}_j$. In addition, we generate N encryption keys that revoke no party and are not used for tracing (conceptually, these keys revoke the set $\mathcal{R}_0$), denoted by K_{NORM}. Note that there are keys that revoke $\mathcal{R}_0$ (i.e., revoke no party) in both K_{TR} and K_{NORM}. The keys in K_{TR} that revoke the set $\mathcal{R}_0$ are used by $\mathsf{TrEnc}(\cdot, \mathcal{R}_0)$ queries (see Figure 5). Finally, the set of encryption keys is $K_{\text{TR}} \cup K_{\text{NORM}}$. Crucially, we publish the encryption keys in a random order $\mathcal{K} \xleftarrow{\$} \mathsf{Shuffle}(K_{\text{Tr}} \cup K_{\text{NORM}})$ to hide the type of each key (i.e., which set it revokes) from the adversary. Note that K_{TR} and K_{NORM} are stored in their original order as a private tracing key. Here, we treat all sets as sequences, which means that we also store the index (in practice, the tracing key can be the random permutation applied by $\mathsf{Shuffle}$). We get $|K_{\text{NORM}}| = |K_{\text{TR}}| = N$ and $|\mathcal{K}| = 2N$.

Observe that the entity running the preprocessing phase could act maliciously and generate encryption keys such that a ciphertext can be decrypted using secret keys under its control. That

is, it can use malicious public keys, for which it knows the corresponding secret keys instead of the correct public keys of the parties. To mitigate this attack, it is required to attach a NIZK proof of knowledge to the rerandomization algorithm that proves knowledge of a witness $w = (\mathcal{R}, \alpha, \bar{\alpha} := \{\bar{\alpha}_\ell\}_{\ell \in \mathcal{R}})$ to the statement $\chi = (\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, [\alpha]_1, \mathsf{ek}, \mathsf{ak})$. The relation $\mathsf{R}_{\mathsf{Rand}}$ for corresponding proof system $\mathsf{PS}_{\mathsf{Rand}}$ (Definition A.1) is defined as

$$\mathsf{R}_{\mathsf{Rand}} = \left\{ (\chi, w) \colon ([\alpha]_1, \mathsf{ek}, \mathsf{ak}) = \mathsf{Rand}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}; \alpha, \bar{\alpha}) \right\}.$$

We define the algorithm $\mathsf{RandAndProve}$, extending Rand (Figure 6), in which we attach to the output of Rand a proof of knowledge in $\mathsf{PS}_{\mathsf{Rand}}$. Intuitively, the additional proof prevents the adversary from using its own secret keys, which it might use to break semantic security. We discuss the proofs of semantic security and message privacy in Appendix I.

$\mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m})$

for $\ell \in [\lambda_c]$ **do**
  $\mathsf{idx} \xleftarrow{\$} [|\mathcal{K}|]$ without repetitions
  $([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \leftarrow \mathcal{K}[\mathsf{idx}]$
  (c_1, c_2, c_3, c_4)
    $\leftarrow \mathsf{enc}(\mathsf{CRS}, [\alpha]_1, \mathsf{ek}, \mathsf{ak}, \mathsf{m})$
  // Replace $c_2 = \mathsf{ak}$, with idx.
  $\mathsf{ct}_\ell \leftarrow (c_1, \mathsf{idx}, c_3, c_4)$
return $\mathsf{ct} \leftarrow (\mathsf{ct}_1, \ldots, \mathsf{ct}_{\lambda_c})$

$\mathsf{TrEnc}(\mathsf{CRS}, \mathsf{tk} := (K_{\text{NORM}}, K_{\text{TR}}), \mathcal{K}, t, \mathsf{m}, \mathcal{R}_j)$

for $\ell \in [\lambda_c]$ **do**
  Sample as follows without repetitions:
  With prob. $1/2 + (j+1)/(2n+2)$ **do**
    $k := ([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \xleftarrow{\$} K_{\text{Tr}}[j]$
  With prob. $(n-j)/(2n+2)$ **do**
    $k := ([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \xleftarrow{\$} \bigcup_{v\in\{j+1,\ldots,n\}} K_{\text{Tr}}[v]$
  $\mathsf{idx} \leftarrow$ Index of k in $\mathcal{K}$
  $(c_1, c_2, c_3, c_4) \leftarrow \mathsf{enc}(\mathsf{CRS}, [\alpha]_1, \mathsf{ek}, \mathsf{ak}, \mathsf{m})$
  // Replace $c_2 = \mathsf{ak}$, with the index idx.
  $\mathsf{ct}_\ell \leftarrow (c_1, \mathsf{idx}, c_3, c_4)$
return $\mathsf{ct} \leftarrow (\mathsf{ct}_1, \ldots, \mathsf{ct}_{\lambda_c})$

Figure 10: The Enc and TrEnc algorithms of Π_{OSTEP} with respect to correctness parameter λ_c. The helper function enc is as defined for Π_{OSTE} in Figure 7.

Encryption. The encryption algorithms are presented in Figure 10. For normal encryption (Enc), we choose λ_c different keys at random from the preprocessing $\mathcal{K}$ and encrypt under them. We replace the ak part in the ciphertexts with the respective index such that the ciphertext size is now constant in the number of parties n. As some keys revoke certain parties from contributing to decryption, we need to encrypt under multiple keys to amplify decryption correctness.

In TrEnc, we encrypt the message m with respect to the revoked set $\mathcal{R}_j$. Note that the execution of TrEnc requires the tracing key $\mathsf{tk} = (K_{\text{NORM}}, K_{\text{TR}})$, i.e., the tracer needs to know which keys in $\mathcal{K}$ revoke which parties. Denote by $K_{\text{TR}}[j]$ the set of keys that revoke the set $\mathcal{R}_j$. As a first attempt, let us sample λ_c keys from $K_{\text{TR}}[j]$ and encrypt the message m under these keys. While this attempt satisfies message privacy, an adversary can distinguish a normal encryption from $\mathsf{TrEnc}(\mathcal{R}_0)$, breaking indistinguishability. In particular, a normal encryption would likely also include some tracing keys that revoke $\mathcal{R}_j$ for $j > 0$. Hence, an adversary, given the preprocessing $\mathcal{K}$, the indices in the λ_c ciphertexts, and the secret keys of some parties, could trivially check if any of the used keys revoke anyone. To overcome this, when executing $\mathsf{TrEnc}(\mathcal{R}_0)$, we also sample some keys from $K_{\text{Tr}}[j]$ for $j > 0$. Let $k_1, \ldots, k_{\lambda_c}$ be the preprocessing keys from $K_{\text{NORM}} \cup K_{\text{TR}}$ used to encrypt the ciphertext ct. We next consider two cases. If ct was encrypted using Enc, then $\Pr[k_i \text{ revokes } \mathcal{R}_0] = \frac{1}{2} + \frac{1}{2n+2}$, since this event happens if

$k_i \in K_{\text{NORM}} \cup K_{\text{TR}}[0]$. Furthermore, for every $1 \leq j \leq n$, it holds that $\Pr[k_i \text{ revokes } \mathcal{R}_j] = \frac{1}{2n+2}$, since this event happens if $k_i \in K_{\text{TR}}[j]$. If ct was encrypted as $\mathsf{TrEnc}(\mathcal{R}_0)$, then we simulate the first case by choosing k_i from $K_{\text{TR}}[0]$ with probability $\frac{1}{2} + \frac{1}{2n+2}$ and with probability $\frac{1}{2n+2}$ from $K_{\text{TR}}[j]$ for every $1 \leq j \leq n$. Observe that the distributions in both cases are indistinguishable to the adversary due to the indistinguishability property of the underlying OSTE scheme. Namely, if k_j revokes no party, the adversary cannot know if $k_j \in K_{\text{TR}}[0]$ or $k_j \in K_{\text{NORM}}$. We extrapolate this argument for membership hiding, i.e., when executing $\mathsf{TrEnc}(\mathcal{R}_j)$, we also sample some keys from $K_{\text{TR}}[j']$, where $j' > j$. To see why the distributions are identical (except for negligible difference), we rely on the membership hiding property of the underlying OSTE scheme, i.e., an adversary that does not know sk_j can not distinguish between keys in $K_{\text{TR}}[j-1]$ and keys in $K_{\text{TR}}[j]$.

6.2 Proof of Theorem 6.2

The correctness of the scheme is straightforward given the correctness of the Π_{OSTE} construction. The correctness is lower bounded by $1 - \frac{1}{2^{\lambda_c}}$ since the probability to sample a key that does not revoke any party is at least $1/2$. We need to prove that our Π_{OSTEP} construction fulfills $\mathsf{IND\text{-}CPA}$, $(\frac{\epsilon}{16})$-indistinguishability (Definition 3.3), $(\frac{\epsilon}{16n})$-membership hiding (Definition 3.4), and message privacy (Definition 3.5). The full proofs for message privacy and $\mathsf{IND\text{-}CPA}$ security can be found in Appendix I. We next prove the indistinguishability property and membership hiding.

Remark 6.4 We note that the tracing authority is trusted for tracing but not semantic security. The tracing key tk does not give the adversary any advantage in the semantic security game. This holds since tk only reveals which keys revoke which parties. In fact, in our $\mathsf{IND\text{-}CPA}$ security game (Figure 11), the preprocessing is sampled by the adversary, which means security is required to hold, even if the adversary knows α and $\bar{\alpha}$.

Indistinguishability. As for Π_{OSTE}, observe that tracing keys that trace $\mathcal{R}_0$ (the keys in $K_{\text{TR}}[0]$) are identically distributed to normal encryption keys (the keys in K_{NORM}). Hence, a decoder can only distinguish normal ciphertexts from $\mathsf{TrEnc}(\mathcal{R}_0)$ ciphertexts through the difference of distributions, with respect to $\mathcal{K}$, that are generated by Enc and $\mathsf{TrEnc}(\mathcal{R}_0)$. In particular, we show that the adversary cannot choose a subset of encryption keys that enables him to construct a decoder that distinguishes between normal ciphertexts and TrEnc ciphertexts with significant probability, i.e., $1/2 + \epsilon/16$.

Lemma 6.5 *The Π_{OSTEP} construction, as presented in Section 6.1, is $(\epsilon/16)$-indistinguishable for $\epsilon = 1/\mathsf{poly}(\lambda)$, per Definition 3.3, for some $d = O(\lambda_c^2 \lambda n^2/\epsilon^2)$.*

Proof. Consider $\delta = \frac{\epsilon}{16}$ for an $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$. Let $n = \mathsf{poly}(\lambda)$ and let $\mathcal{A}$ be a PPT adversary. We show for construction Π_{OSTEP} that $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{GameInd}(1^\lambda, n, \delta)} \leq \mathsf{negl}(\lambda)$, where $\mathsf{GameInd}_{\mathcal{A}}$ is defined in Figure 2.

First, note that in order to win $\mathsf{GameInd}(1^\lambda, n, \delta)$, the adversary $\mathcal{A}$ is required to output a distinguishing decoder box $\mathcal{D}$ which is a δ-$\text{Dist}^{\mathsf{Enc},\mathsf{TrEnc}}$ for $\delta = \epsilon/16$, per Definition 3.2. As we consider only stateless decoders, $\mathcal{D}$ cannot rely on runtime information, such as how often a key has occurred. Furthermore, note that Π_{OSTEP} ensures that the same encryption key cannot be used twice in the same ciphertext. Consequently, $\mathcal{D}$ cannot assess the size of the keyspace from which keys are sampled. Also, recall that the keys in K_{NORM} are distributed identically to the keys in $K_{\text{TR}}[0]$. As the preprocessing $\mathcal{K}$ is randomly permuted according to tk, an adversary who does not know tk can only guess whether a given key that revokes $\mathcal{R}_0$ either belongs to K_{NORM} or $K_{\text{TR}}[0]$. Essentially, by construction, the only difference between

Enc ciphertexts and $\mathsf{TrEnc}(\mathcal{R}_0)$ ciphertexts is that the latter does not use keys from K_{NORM}. Therefore, to break indistinguishability, $\mathcal{D}$ can only try to use some strategy that utilizes the keys in $K_{\text{NORM}} \cup K_{\text{TR}}[0]$. I.e., its behavior would depend on the occurrence of one or more of these keys in the ciphertext ct. Let us assume that the adversary chooses a subset of keys $K \subseteq K_{\text{NORM}} \cup K_{\text{TR}}[0]$ for which $\mathcal{D}$ fixes a behavior. We show that the probabilities of drawing a key from K when, in the indistinguishability game, $b = 0$ or $b = 1$ fall apart by less than $\epsilon/16$ with overwhelming probability. As a result, we get that the probability that $\mathcal{D}$ behaves significantly differently in the two cases, and as such is $\frac{\epsilon}{16}$-Dist$^{\mathsf{Enc},\mathsf{TrEnc}}$, is negligible. That is, we show that $\Pr[\mathcal{D}(\mathsf{ct}_b) = b] < \frac{1}{2} + \frac{\epsilon}{16}$ with overwhelming probability.

To see this, consider the subset of keys K and let $L_0 = |K \cap K_{\text{NORM}}|$ and $L_1 = |K \cap K_{\text{TR}}[0]|$. Furtheremore, let ct be a ciphertext that is either an Enc ciphertext or a $\mathsf{TrEnc}(\mathcal{R}_0)$ ciphertext and let $k_1, \ldots, k_{\lambda_c}$ denote the keys used to encrypt ct, that is, the j-th component of ct is encrypted under k_j. By construction, when $b = 0$ (i.e., normal encryption), for every key $1 \leq j \leq \lambda_c$, it holds that $\Pr[k_j \in K_{\text{NORM}}] = 1/2$ and $\Pr[k_j \in K_{\text{TR}}[0]] = \frac{1}{2n+1}$. When $b = 1$ (i.e., $\mathsf{TrEnc}(\mathcal{R}_0)$), these probabilities are 0 and $\frac{1}{2} + \frac{1}{2(n+1)}$, respectively. Let y_0, y_1 be the expected size of $K \cap \{k_1, \ldots, k_{\lambda_c}\}$ when $b = 0$ and $b = 1$, respectively. For $\mathcal{D}$ to break indistinguishability, it must behave sufficiently differently for $b = 0$ and $b = 1$. Then, since the adversary cannot distinguish which keys are from L_0 and which are from L_1, it must hold that $|y_0 - y_1| \geq \epsilon/16$ with overwhelming probability. By construction, it holds that $y_0 = \lambda_c \cdot (\frac{1}{2}\frac{L_0}{d(n+1)} + \frac{1}{2(n+1)}\frac{L_1}{d})$ and $y_1 = \lambda_c \cdot (\frac{1}{2}\frac{L_1}{d} + \frac{1}{2(n+1)}\frac{L_1}{d})$. By the requirement that $|y_0 - y_1| \geq \epsilon/16$, we get

$$|L_0 - L_1(n+1)| \geq \frac{2d\epsilon(n+1)}{16\lambda_c}. \tag{2}$$

W.l.o.g, assume that $L_0 > L_1$ and let μ denote the expected size of L_0. Since $|K_{\text{NORM}}| = (n+1) \cdot |K_{\text{TR}}[0]|$ and $\mathcal{A}$ picks keys from $K_{\text{NORM}} \cup K_{\text{TR}}[0]$ at random, with the same distribution for K_{NORM} and $K_{\text{TR}}[0]$ (since he can not distinguish between the two sets), then the expected size of L_1 is $\frac{\mu}{n+1}$. Furtheremore, it holds that $L_0 - \mu = \frac{\mu}{n+1} - L_1$. Thus, for inequality 2 to hold, it must hold that:

$$|L_0 - \mu + (L_0 - \mu)(n+1)| \geq \frac{2d\epsilon(n+1)}{16\lambda_c}$$
$$|L_0 - \mu| \geq \frac{2d\epsilon(n+1)}{16\lambda_c(n+2)}.$$

By the Chernoff bound, for $\delta_c = \frac{2d\epsilon(n+1)}{16\lambda_c(n+2)\mu}$, we obtain

$$\Pr[|L_0 - \mu| > \delta_c\mu] \leq 2e^{-\mu\delta_c^2/3}. \tag{3}$$

As $\mu \leq |K_{\text{NORM}}| = d(n+1)$, it holds that for some $d = O(\frac{\lambda_c^2\lambda n^2}{\epsilon^2})$, the probability in Equation 3 is negligible. Therefore, inequality 2 could hold only with negligible probability. Hence, constructing a distinguishing decoder $\mathcal{D}$ with advantage $\frac{\epsilon}{16}$ could happen only with negligible probability. The probability of winning the indistinguishability game is negligible for any subset K chosen by the adversary. By union bound, the probability is also negligible if the adversary chooses a polynomial number of sets K.

■

Membership Hiding. The following lemma captures the membership hiding property of Π_{OSTEP}.

Lemma 6.6 *The Π_{OSTEP} construction, as presented in Section 6.1, is membership hiding for $\delta = \epsilon/16n$ and $\epsilon = 1/\mathsf{poly}(\lambda)$, per Definition 3.4, for some $d = O(\lambda\lambda_c^2 n^2/\epsilon^2)$.*

Proof. Let $n = \mathsf{poly}(\lambda)$, $\delta = \frac{\epsilon}{16n}$, and $\epsilon \in \frac{1}{\mathsf{poly}(\lambda)}$. We show that

$$\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A},\Pi_{\mathsf{OSTEP}}}(1^\lambda, n, \delta) \leq \mathsf{negl}(\lambda)$$

Our proof proceeds in two steps: First, we show that the keys in $K_{\text{TR}}[i-1]$ are indistinguishable from those in $K_{\text{TR}}[i]$ for any adversary that does not know sk_i. Intuitively, this follows from the membership hiding property of the underlying Π_{OSTE} scheme. Then, in the second step, relying on the fact that the only difference between $\mathsf{TrEnc}(\cdot, \mathcal{R}(i-1))$ and $\mathsf{TrEnc}(\cdot, \mathcal{R}(i))$ is that the former uses also keys from $K_{\text{TR}}[i-1]$ while the latter does not, we show that the adversary can not artificially choose a set of keys K from the set of $K_{\text{TR}}[i-1] \cup K_{\text{TR}}[i]$ that allows distinguishing between ciphertexts that use keys in $K_{\text{TR}}[i]$ from ciphertexts that do not. In combination, the indistinguishability of keys in $K_{\text{TR}}[i]$ and $K_{\text{TR}}[i-1]$ and the limited advantage an adversary can gain based on their occurrence imply membership hiding.

Proving that $K_{\mathbf{tr}}[i-1] \approx_c K_{\mathbf{tr}}[i]$. Ultimately, we aim to show that $\mathsf{GameMH}^{\Pi_{\mathsf{OSTEP}}}$ (Figure 2) is computationally close to a variant of the game, where in the preprocessing, we sample all keys in $K_{\text{TR}}[i-1]$ by revoking $\mathcal{R}_i$ instead of $\mathcal{R}_{i-1}$, i.e., equivalently to $K_{\text{TR}}[i]$. We do so through a series of game hops involving the zero-knowledge and simulation-extractability property of the proof system $\mathsf{PS}_{\mathsf{KGen}}$ and the zero-knowledge property of $\mathsf{PS}_{\mathsf{Rand}}$ (Definition A.1). We reduce the final step, where we substitute the keys for $K_{\text{TR}}[i-1]$, to the XDH assumption (Definition 2.2).

Before our reduction to XDH, we need to simulate the KGen proof on behalf of party i, extract the secret keys from the KGen proofs the adversary sends on behalf of all other parties, and simulate the Rand proofs for the preprocessing keys in $K_{\text{TR}}[i-1]$. These steps are necessary because, in our reduction to XDH, we embed the XDH challenge into the public key of party i. The reduction needs to know the secret keys of the honest parties to simulate the preprocessing keys with respect to the XDH challenge, which is why we need to extract them. Further, the reduction does not know sk_i, as well as α and possibly $\bar{\alpha}_{i-1}$ for $K_{\text{TR}}[i-1]$, so these proofs must be simulated.

Let Game_0 be the the membership hiding game $\mathsf{GameMH}^{\Pi_{\mathsf{OSTEP}}}$ for Π_{OSTEP}.

$\underline{\mathsf{Game}_1}$: In the first game hop, we replace the proof π_i in $\mathsf{PS}_{\mathsf{KGen}}$ that the challenger attaches to the public keys and hints of the honest party with a simulated one

$$\pi_i \xleftarrow{\$} \mathsf{PS}_{\mathsf{KGen}}.\mathsf{Sim}(\mathsf{td}_{\mathsf{KGen}}, \mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i).$$

Claim 6.7 If the proof system $\mathsf{PS}_{\mathsf{KGen}}$ is zero-knowledge (Definition A.1), then Game_0 and Game_1 are computationally indistinguishable.

Proof. The claim follows directly from the zero-knowledge property of $\mathsf{PS}_{\mathsf{KGen}}$. In particular, it holds that

$$(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \approx_c (\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \mathsf{PS}_{\mathsf{KGen}}.\mathsf{Sim}(\mathsf{td}_{\mathsf{KGen}}, \mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i))$$

for $(\mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda)$. ∎

$\underline{\mathsf{Game}_2}$: In Game_2, we use the extractor of $\mathsf{PS}_{\mathsf{KGen}}$ to extract $\{\mathsf{sk}_\ell\}_{\ell \in [n] \setminus \{i\}}$ from the proofs attached to the adversary's public keys and hints as $\mathsf{sk}_\ell \xleftarrow{\$} \mathsf{PS}_{\mathsf{KGen}}.\mathsf{Ext}(\mathsf{td}_{\mathsf{KGen}}, (\mathsf{CRS}, \mathsf{pk}_\ell, \mathsf{hint}_\ell), \pi_\ell)$. Additionally, Game_2 aborts if there exists an $\ell \in [n] \setminus \{i\}$ such that $\mathsf{pk}_\ell \neq [\mathsf{sk}_\ell]_1$ or $\mathsf{hint}_\ell \neq ([\mathsf{sk}_\ell \tau]_1, \ldots, [\mathsf{sk}_\ell \tau^n]_1)$.

Claim 6.8 For all $n = \mathsf{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ such that

$$\begin{aligned}&\left|\Pr[\mathsf{Game}_{1,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[\mathsf{Game}_{2,\mathcal{B}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\right| \\ &\leq (n-1)\mathbf{Adv}^{\text{sim-extract}}_{\mathcal{B},\mathsf{PS}_{\mathsf{KGen}}}(1^\lambda).\end{aligned}$$

The proof of the above claim is identical to that in the membership hiding proof for Π_{OSTE} (Lemma 5.3).

$\underline{\mathsf{Game}_3}$: In Game_3, we replace the proofs attached to re-randomizations in $K_{\text{TR}}[i-1]$ with simulated ones. In particular, we set

$$k.\pi \xleftarrow{\$} \mathsf{PS}_{\mathsf{Rand}}.\mathsf{Sim}(\mathsf{td}_{\mathsf{Rand}}, (\mathsf{CRS}, \{\mathsf{pk}_\ell, \mathsf{hint}_\ell\}_{\ell\in[n]}, k.[\alpha]_1, k.\mathsf{ek}, k.\mathsf{ak}))$$

for all $k \in K_{\text{TR}}[i-1]$.

Claim 6.9 For all PPT adversaries $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ such that

$$\begin{aligned}&\left|\Pr[\mathsf{Game}_{2,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[\mathsf{Game}_{3,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\right| \\ &\leq d \cdot \mathbf{Adv}^{\text{zero-knowledge}}_{\mathcal{B},\mathsf{PS}_{\mathsf{Rand}}}(1^\lambda),\end{aligned}$$

where $d = N/(n+1)$.

Proof. The claim follows from a simple hybrid argument over the $N/(n+1)$ proofs in $K_{\text{TR}}[i-1]$. ■

$\underline{\mathsf{Game}_4}$: In Game_4, we finally modify how the keys in $K_{\text{TR}}[i-1]$ are sampled. In particular, we sample each key in $k \in K_{\text{TR}}[i-1]$ by re-randomizing with respect to the revoked set $\mathcal{R}_i$ instead of $\mathcal{R}_{i-1}$, i.e., for all $k \in K_{\text{TR}}[i-1]$, we sample $k \xleftarrow{\$} \mathsf{Rand}(\mathsf{CRS}, \{\mathsf{pk}_j, \mathsf{hint}_j\}_{j\in[n]}, \mathcal{R}_i)$. Note that in Game_3, the corresponding proofs are already simulated.

Claim 6.10 For all PPT adversaries $\mathcal{A}$ there exists a PPT adversary $\mathcal{B}$ and a negligible function negl such that

$$\begin{aligned}&\left|\Pr[\mathsf{Game}_{3,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[\mathsf{Game}_{4,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\right| \\ &\leq d \cdot \mathbf{Adv}^{\mathsf{XDH}}_{\mathcal{B},\mathcal{G}}(1^\lambda) + \mathsf{negl}(\lambda),\end{aligned}$$

where $d = N/(n+1)$.

Proof. We prove the claim through a hybrid argument. Let H_0 be Game_3, H_d be Game_4 and H_u, for $0 < u \leq d$ be the game, where we replace the first u keys of $K_{\text{TR}}[i-1]$ with keys that revoke $\mathcal{R}_i$. We can upper-bound the success probability of any PPT adversary $\mathcal{A}$ to distinguish between Game_3 and Game_4 by that of an adversary $\mathcal{A}'$ who tries to distinguish H_{u-1} from H_u for arbitrary $0 < u \leq d$:

$$\begin{aligned}&\left|\Pr[\mathsf{Game}_{3,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[\mathsf{Game}_{4,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\right| \\ &\leq d \cdot \left|\Pr[H_{u-1,\mathcal{A}'}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[H_{u,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\right|\end{aligned}$$

To conclude the proof, we show that we can further upper bound the success probability of $\mathcal{A}'$ by reducing to XDH.

Let $\mathcal{A}'$ be a PPT adversary that distinguishes between H_{u-1} and H_u. We construct a PPT reduction $\mathcal{B}$ that internally uses $\mathcal{A}'$ to break the XDH assumption (Definition 2.2) for the pairing ensemble $\mathcal{G}$.

First, $\mathcal{B}$ receives an XDH challenge $[x]_1$, $[y]_1$, T, where either $T = [xy]_1$ or $T = [v]_1$ for a random $v \xleftarrow{\$} \mathbb{Z}_p^*$. The reduction $\mathcal{B}$ samples a random $\tau \xleftarrow{\$} \mathbb{Z}_p$ and generates the CRS. Then, it runs $\mathcal{A}'(1^\lambda, \mathsf{CRS})$ to receive the party's index i. It simulates a public key pk_i and hint hint_i for i by embedding the XDH challenge as follows:

$$\mathsf{pk}_i \leftarrow [x]_1, \text{ and } \mathsf{hint}_i \leftarrow (\tau[x]_1, \ldots, \tau^n[x]_1).$$

Observe that $\mathcal{B}$ therefore simulates $\mathsf{sk}_i = x$ without knowledge of x. $\mathcal{B}$ further simulates a proof $\pi_i^{\mathsf{PS}_{\mathsf{KGen}}}$ according to both H_{u-1} and H_u and runs $\mathcal{A}'(\mathsf{pk}_i, \mathsf{hint}_i, \pi_i^{\mathsf{PS}_{\mathsf{KGen}}})$. As a response, $\mathcal{B}$ receives a revoked set $\mathcal{R}$[4] as well as the corrupted public keys and hints with attached proof $\{\mathsf{pk}_\ell, \mathsf{hint}_\ell, \pi_\ell^{\mathsf{PS}_{\mathsf{KGen}}}\}_{\ell \in [n]\setminus\{i\}}$. Again, according to both games, $\mathcal{B}$ extracts the secret keys sk_ℓ from the proofs $\pi_\ell^{\mathsf{PS}_{\mathsf{KGen}}}$, aborting if the extraction is unsuccessful for any ℓ (in case either game aborts, $\mathcal{B}$ outputs 0).

Next, $\mathcal{B}$ simulates the preprocessing as follows. $\mathcal{B}$ generates all preprocessing keys in K_{NORM} and $K_{\text{TR}}[j]$ for $j \neq i-1$ honestly. To generate tracing keys for $K_{\text{TR}}[i-1, q]$ where $q \in [d]$, $\mathcal{B}$ does the following:

- **For $\mathbf{q} < \mathbf{u}$**, generate tracing keys that revoke the set $\mathcal{R}_i$.
- **For $\mathbf{q} > \mathbf{u}$**, generate tracing keys that revoke the set $\mathcal{R}_{i-1}$
- **For $\mathbf{q} = \mathbf{u}$**, embed the XDH challenge into the re-randomization by executing

$$([\alpha]_1, \mathsf{ek}, \mathsf{ak}) \leftarrow \mathsf{SimRand}([x]_1, [y]_1, [z]_1, T, \tau, \{\mathsf{sk}_\ell\}_{\ell \in [n]\setminus\{i\}}, \mathcal{R}_{i-1}, i),$$

  where $\mathsf{SimRand}$ is the same simulation algorithm that we use in the membership hiding proof of Π_{OSTE} (Figure 8). $\mathcal{B}$ can also simulate the corresponding proof in $\mathsf{PS}_{\mathsf{Rand}}$, as required by both hybrids, because the statement is part of $\mathcal{L}_{\mathsf{Rand}}$.

Let b_{XDH} be the internal bit of the XDH challenge. If $b_{\mathsf{XDH}} = 0$, then $T = [xy]_1 \equiv [\alpha \mathsf{sk}_i]_1$ (i.e., party i is unrevoked in $K_{\text{TR}}[i-1, u]$) and we simulate the preprocessing for H_{u-1}. If $b_{\mathsf{XDH}} = 1$, then $T = [v]_1 \equiv [\bar{\alpha}_i \mathsf{sk}_i]_1$ (i.e., party i is revoked in $K_{\text{TR}}[i-1, u]$) and we simulate the preprocessing for H_u.

The reduction assembles the preprocessing and sends it to $\mathcal{A}'$, receiving a decoder $\mathcal{D}$ and a message m. Finally, $\mathcal{B}$ checks if $\mathcal{A}'$ wins either game (both games have the same win condition) by checking if $\mathcal{D}$ is δ-$\text{Dist}^{\mathcal{R}_{i-1}, \mathcal{R}_i}$ according to Definition 3.2[5] and whether the remaining winning conditions apply. If $\mathcal{A}'$ wins, then $\mathcal{B}$ outputs 1. Otherwise, $\mathcal{B}$ outputs 0.

Further, recall that if $b_{\mathsf{XDH}} = 0$, then $\mathcal{B}$ simulates H_{u-1} as the key for $K_{\text{TR}}[i-1, u]$ does not revoke party i. If $b_{\mathsf{XDH}} = 1$, then $\mathcal{B}$ simulates H_u as party i is revoked in $K_{\text{TR}}[i-1, u]$. Hence, we can bound the probability of $\mathcal{A}'$ to distinguish as follows:

$$\begin{aligned}
&\Big|\Pr[H_{u-1,\mathcal{A}'}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[H_{u,\mathcal{A}'}(1^\lambda, n, \mathcal{L}, \delta) = 1]\Big| \\
&\leq \Big|\Pr[\mathcal{B}(1^\lambda, E, [x]_1, [y]_1, [xy]_1) = 1] - \Pr[\mathcal{B}(1^\lambda, E, [x]_1, [y]_1, [v]_1) = 1]\Big| + \mathsf{negl}(\lambda) \\
&\leq \mathbf{Adv}^{\mathsf{XDH}}_{\mathcal{B},\mathcal{G}}(1^\lambda) + \mathsf{negl}(\lambda),
\end{aligned}$$

[4] As Π_{OSTEP} only supports the linear revocation order, it must hold that $\mathcal{R} = \mathcal{R}_{i-1}$. Otherwise, either game will output 0.

[5] Note that while $\mathcal{B}$ can not compute the exact probability, it can approximate it up to negligible error $\mathsf{negl}(\lambda)$ using a polynomial number of queries by applying a Chernoff bound.

where $E \xleftarrow{\$} \mathcal{G}(1^\lambda)$ and $x, y, v \xleftarrow{\$} \mathbb{Z}_p^*$.

Overall, we can bound the success probability of any PPT adversary $\mathcal{A}$ to distinguish between Game_3 and Game_4 by

$$\begin{aligned}&\Big|\Pr[\mathsf{Game}_{3,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1] - \Pr[\mathsf{Game}_{4,\mathcal{A}}(1^\lambda, n, \mathcal{L}, \delta) = 1]\Big| \\ &\leq d \cdot \mathbf{Adv}^{\mathsf{XDH}}_{\mathcal{B},\mathcal{G}}(1^\lambda) + \mathsf{negl}(\lambda),\end{aligned}$$

for some negligible function negl. ■

Proving the negligibility of $\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A},\Pi_{\mathsf{OSTEP}}}$. First note that in order to win $\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A},\Pi_{\mathsf{OSTEP}}}(1^\lambda, n, \delta)$, the adversary $\mathcal{A}$ is required to return a distinguishing decoder $\mathcal{D}$ which is a δ-$\mathrm{Dist}^{\mathcal{R},\mathcal{R}\cup\{i\}}$ (per Definition 3.2) for $\delta = \epsilon/(16n)$. Similarly to our argument in Lemma 6.5, we need only to prove that the adversary cannot find a subset of keys K from $K_{\text{TR}}[i-1] \cup K_{\text{TR}}[i]$ by which the probability of drawing a key from K when $b = 0$ in the membership hiding game falls apart from the probability when $b = 1$ by more than $\frac{\epsilon}{16n}$. To see this, let $L_{i-1} = |K \cap K_{\text{TR}}[i-1]|$ and $L_i = |K \cap K_{\text{TR}}[i]|$. Further, let ct be the challenge ciphertext that is either $\mathsf{TrEnc}(\mathcal{R}_{i-1})$ or $\mathsf{TrEnc}(\mathcal{R}_i)$ and let $k_1, \ldots, k_{\lambda_c}$ be the encryption keys from $K_{\text{NORM}} \cup K_{\text{TR}}$ that are used in the encryption of ct. By construction, it holds that $K_{\text{TR}}[i-1] = K_{\text{TR}}[i] = d$ and when $b = 0$, i.e., ct is $\mathsf{TrEnc}(\mathcal{R}_{i-1})$, then it holds that $\Pr[k_j \in K_{\text{TR}}[i-1]] = \frac{1}{2} + \frac{i}{2(n+1)}$ and $\Pr[k_j \in K_{\text{TR}}[i]] = \frac{1}{2(n+1)}$. When $b = 1$, i.e., $\mathsf{ct} = \mathsf{TrEnc}(\mathcal{R}_i)$, then these probabilities are 0 and $\frac{1}{2} + \frac{i+1}{2(n+1)}$, respectively. Let y_0, y_1 be the expected size of $K \cap \{k_1, \ldots, k_{\lambda_c}\}$ when $b = 0$ and $b = 1$, respectively. Similarly to Lemma 6.5, we argue that the adversary can win the membership hiding game only if it holds that $|y_0 - y_1| > \epsilon/(16(n-t))$. This is required since, as we showed, the adversary can not distinguish between keys from L_{i-1} and keys from L_i. Thus, its strategy would depend only on the occurrence of the keys in $L_{i-1} \cup L_i$. We next show that

$$|y_0 - y_1| < \frac{\epsilon}{16n} \tag{4}$$

with overwhelming probability when we choose the appropriate d as in the lemma.

First, note that:

$$y_0 = \lambda_c \left(\left(\frac{1}{2} + \frac{i}{2(n+1)} \right) \frac{L_{i-1}}{d} + \frac{1}{2(n+1)} \frac{L_i}{d} \right)$$

and

$$y_1 = \lambda_c \left(\frac{1}{2} + \frac{i+1}{2(n+1)} \right) \frac{L_i}{d}.$$

Hence, Equation 4 would not hold if and only if

$$|L_{i-1} - L_i| \geq \frac{\epsilon d}{16 n \lambda_c} \left(\frac{2(n+1)}{n+i+1} \right). \tag{5}$$

W.l.g, assume that $L_{i-1} > L_i$ and let μ denote the expected size of L_{i-1}. Since $|K_{\text{TR}}[i-1]| = |K_{\text{TR}}[i]|$ and $\mathcal{A}$ picks keys from $K_{\text{TR}}[i-1] \cup K_{\text{TR}}[i]$ at random, with the same distribution for $K_{\text{TR}}[i-1]$ and $K_{\text{TR}}[i]$, then it holds that the expected size of L_i is also μ. Therefore, $L_{i-1} - \mu = \mu - L_i$. Hence, from Equation 5, we get it must hold that

$$|L_{i-1} - \mu| > \frac{\epsilon d}{32 n \lambda_c}.$$

By the Chernoff bound, for $\delta_c = \frac{\epsilon d}{32 n \lambda_c \mu}$, it holds that:

$$\Pr\left[|L_{i-1} - \mu| > \frac{\epsilon d}{32 n \lambda_c}\right] \leq 2e^{-\mu \delta_C^2/3},$$

which is negligible if

$$\lambda \leq \mu \delta_C^2/3 = \frac{\epsilon^2 d^2}{(32)^2 3 n^2 \lambda_c^2 \mu}.$$

Thus, since $\mu \leq d/2$, it must hold that $d = O(\frac{\lambda_c^2 \lambda n^2}{\epsilon^2})$. Given this number of keys, the probability that a subset K can influence the success probability of $\mathcal{D}$ by $\delta = \frac{\epsilon}{16n}$ is only negligible. As the adversary can try only a polynomial number of sets K, then by union bound, the probability of winning the game will be negligible. ■

References

[1] Baird, L., Garg, S., Jain, A., Mukherjee, P., Sinha, R., Wang, M., Zhang, Y.: Threshold signatures in the multiverse. In: 44th IEEE Symposium on Security and Privacy, SP (2023) (Cited on page 6.)

[2] Ballard, L., Green, M., de Medeiros, B., Monrose, F.: Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive, Paper 2005/417 (2005), https://eprint.iacr.org/2005/417 (Cited on page 9.)

[3] Bebel, J., Ojha, D.: Ferveo: Threshold decryption for mempool privacy in bft networks. Cryptology ePrint Archive (2022) (Cited on page 2.)

[4] Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for r1cs. In: Advances in Cryptology - EUROCRYPT 2019. Springer (2019) (Cited on page 8.)

[5] Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M.K. (ed.) Advances in Cryptology - CRYPTO 2004. Springer (2004) (Cited on page 9.)

[6] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Annual international cryptology conference. Springer (2001) (Cited on page 16.)

[7] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: International conference on the theory and application of cryptology and information security. Springer (2001) (Cited on page 16.)

[8] Boneh, D., Naor, M.: Traitor tracing with constant size ciphertext. In: Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS (2008) (Cited on page 2.)

[9] Boneh, D., Partap, A., Rotem, L.: Accountability for misbehavior in threshold decryption via threshold traitor tracing. In: Advances in Cryptology - CRYPTO. Lecture Notes in Computer Science, vol. 14926. Springer (2024) (Cited on page 2, 3, 4, 7, 23.)

[10] Boneh, D., Partap, A., Rotem, L.: Traceable secret sharing: Strong security and efficient constructions. In: Annual International Cryptology Conference. pp. 221–256. Springer (2024) (Cited on page 7.)

[11] Boneh, D., Sahai, A., Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer (2006) (Cited on page 3, 10, 13, 39.)

[12] Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Algorithmica (2017) (Cited on page 7.)

[13] Bormet, J., Faust, S., Othman, H., Qu, Z.: Beat-mev: Epochless approach to batched threshold encryption for mev prevention. Cryptology ePrint Archive (2024) (Cited on page 23.)

[14] Branco, P., Lai, R.W., Maitra, M., Malavolta, G., Rahimi, A., Woo, I.K.: Traitor tracing without trusted authority from registered functional encryption. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer (2024) (Cited on page 7, 8.)

[15] Campanelli, M., Nitulescu, A., Ràfols, C., Zacharakis, A., Zapico, A.: Linear-map vector commitments and their practical applications. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer (2022) (Cited on page 8.)

[16] Chen, Y., Vaikuntanathan, V., Waters, B., Wee, H., Wichs, D.: Traitor-tracing from LWE made simple and attribute-based. In: Theory of Cryptography - 16th International Conference, TCC (2018) (Cited on page 7.)

[17] Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Advances in Cryptology - CRYPTO'94. Springer (1994) (Cited on page 7.)

[18] Choudhuri, A.R., Garg, S., Piet, J., Policharla, G.V.: Mempool privacy via batched threshold encryption: Attacks and defenses. Cryptology ePrint Archive (2024) (Cited on page 2, 23.)

[19] Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded cca2-secure encryption. In: Advances in Cryptology - ASIACRYPT 2007 (2007) (Cited on page 7.)

[20] Das, S., Camacho, P., Xiang, Z., Nieto, J., Bünz, B., Ren, L.: Threshold signatures from inner product argument: Succinct, weighted, and multi-threshold. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (2023) (Cited on page 2.)

[21] Datta, P., Pal, T., Yamada, S.: Registered FE beyond predicates: (attribute-based) linear functions and more. In: Advances in Cryptology - ASIACRYPT 2024 (2024) (Cited on page 8.)

[22] Desmedt, Y.: Society and group oriented cryptography: A new concept. In: Pomerance, C. (ed.) Advances in Cryptology - CRYPTO '87 (1987) (Cited on page 1.)

[23] Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) Advances in Cryptology - CRYPTO '89 (1989) (Cited on page 1.)

[24] Dziembowski, S., Faust, S., Lizurej, T., Mielniczuk, M.: Secret sharing with snitching. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. pp. 840–853 (2024) (Cited on page 7.)

[25] Dziembowski, S., Faust, S., Luhn, J.: Shutter network: Private transactions from threshold cryptography. Cryptology ePrint Archive, Paper 2024/1981 (2024), https://eprint.iacr.org/2024/1981 (Cited on page 2, 23.)

[26] Francati, D., Friolo, D., Maitra, M., Malavolta, G., Rahimi, A., Venturi, D.: Registered (inner-product) functional encryption. In: Advances in Cryptology - ASIACRYPT 2023 (2023) (Cited on page 8.)

[27] Garg, S., Jain, A., Mukherjee, P., Sinha, R., Wang, M., Zhang, Y.: hints: Threshold signatures with silent setup. In: 2024 IEEE Symposium on Security and Privacy (SP). IEEE (2024) (Cited on page 2, 18.)

[28] Garg, S., Kolonelos, D., Policharla, G.V., Wang, M.: Threshold encryption with silent setup. In: Advances in Cryptology - CRYPTO 2024. pp. 352–386. Springer (2024) (Cited on page 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 17, 18, 37, 42, 43, 44, 46, 49.)

[29] Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure distributed key generation for discrete-log based cryptosystems. Journal of Cryptology **20**, 51–83 (2007) (Cited on page 2.)

[30] Gong, J., Luo, J., Wee, H.: Traitor tracing with $n^{1/3}$-size ciphertexts and o(1)-size keys from k-lin. In: Advances in Cryptology - EUROCRYPT 2023 (2023) (Cited on page 2.)

[31] Goyal, R., Koppula, V., Waters, B.: Collusion resistant traitor tracing from learning with errors. In: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (2018) (Cited on page 7, 10, 12.)

[32] Goyal, R., Vusirikala, S., Waters, B.: Collusion resistant broadcast and trace from positional witness encryption. In: Public-Key Cryptography - PKC 2019 (2019) (Cited on page 7.)

[33] Goyal, V., Song, Y., Srinivasan, A.: Traceable secret sharing and applications. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, Virtual Event, August 16–20, 2021, Proceedings, Part III 41. pp. 718–747. Springer (2021) (Cited on page 7.)

[34] Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Advances in Cryptology - ASIACRYPT. pp. 177–194. Springer (2010) (Cited on page 7.)

[35] Luo, J.: Ad hoc broadcast, trace, and revoke. IACR Communications in Cryptology (2024) (Cited on page 8.)

[36] Naor, M., Pinkas, B.: Threshold traitor tracing. In: Advances in Cryptology - CRYPTO '98 (1998) (Cited on page 24.)

[37] Pedersen, T.P.: A threshold cryptosystem without a trusted party. In: EUROCRYPT '91 (1991) (Cited on page 2.)

[38] Ràfols, C., Zapico, A.: An algebraic framework for universal and updatable snarks. In: Annual International Cryptology Conference. Springer (2021) (Cited on page 8.)

[39] Rotem, L., Segev, G.: Algebraic distinguishers: From discrete logarithms to decisional uber assumptions. In: Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part III 18. pp. 366–389. Springer (2020) (Cited on page 37.)

[40] Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Advances in Cryptology—EUROCRYPT'97: International Conference on the Theory and Application of Cryptographic Techniques Konstanz, Germany, May 11–15, 1997 Proceedings 16. pp. 256–266. Springer (1997) (Cited on page 4, 9.)

[41] Tomescu, A., Chen, R., Zheng, Y., Abraham, I., Pinkas, B., Golan-Gueta, G., Devadas, S.: Towards scalable threshold cryptosystems. In: 2020 IEEE Symposium on Security and Privacy (2020) (Cited on page 2.)

[42] Zhandry, M.: New techniques for traitor tracing: Size and more from pairings. In: Annual International Cryptology Conference. Springer (2020) (Cited on page 7, 24.)

[43] Zhandry, M.: Optimal traitor tracing from pairings. Cryptology ePrint Archive (2024) (Cited on page 7.)

[44] Zhu, Z., Li, J., Zhang, K., Gong, J., Qian, H.: Registered functional encryptions from pairings. In: Advances in Cryptology - EUROCRYPT 2024 (2024) (Cited on page 8.)

A Preliminary Definitions

Definition A.1 (NIZK-PoK). Let R be an NP-relation of statement-witness pairs $(\chi, w) \in \mathsf{R}$ a non-interactive zero-knowledge proof system of knowledge PS_R for R is a tuple of algorithms $\mathsf{PS}_\mathsf{R} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Sim}, \mathsf{Ext})$ with the following syntax:

- $(\mathsf{CRS}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$. Given the security parameter λ, the Setup algorithm outputs a common reference string CRS and a simulation and extraction trapdoor td. The CRS is implicit input to all subsequent algorithms.
- $\pi \xleftarrow{\$} \mathsf{Prove}(\chi, w)$. Given a statement-witness pair (χ, w), the Prove algorithm outputs a proof π.
- $1/0 \leftarrow \mathsf{Verify}(\chi, \pi)$. Given a statement χ and a proof π, the Verify algorithm outputs 1, iff π is valid for χ.
- $\pi \xleftarrow{\$} \mathsf{Sim}(\mathsf{td}, \chi)$. Given the trapdoor td and a statement χ, the simulation algorithm Sim outputs a simulated proof π.
- $w \xleftarrow{\$} \mathsf{Ext}(\mathsf{td}, \chi, \pi)$. Given the trapdoor td, a statement χ and a proof π, the extractor Ext returns a witness w.

A proof system PS_R must fulfill the following properties.

- Correctness: For all $(\chi, w) \in \mathsf{R}$ it holds that $\mathsf{Verify}(\chi, \mathsf{Prove}(\chi, w)) = 1$.
- Simulation-Extractability: For all PPT adversaries $\mathcal{A}$, negligible function negl such that

$$\Pr\left[\begin{array}{cc} \mathsf{Verify}(\chi,\pi) = 1 \wedge & (\mathsf{CRS}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda) \\ (\chi, w) \notin \mathsf{R} \wedge & : (\chi, \pi) \xleftarrow{\$} \mathcal{A}^{\mathsf{Sim}(\mathsf{td},\cdot)}(1^\lambda, \mathsf{CRS}) \\ \chi \notin Q & w \xleftarrow{\$} \mathsf{Ext}(\mathsf{td}, \chi, \pi) \end{array}\right] \leq \mathsf{negl}(\lambda),$$

  where Q is the set of $\mathcal{A}$'s queries to the $\mathsf{Sim}(\mathsf{td}, \cdot)$ oracle.

- Zero-Knowledge: For all $(\chi, w) \in \mathsf{R}$ it holds that

$$(\mathsf{CRS}, \chi, \mathsf{Prove}(\chi, w)) \approx_c (\mathsf{CRS}, \chi, \mathsf{Sim}(\mathsf{td}, \chi)),$$

  where $(\mathsf{CRS}, \mathsf{td}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$.

Definition A.2 (Independence [39]). Let $p \in \mathbb{N}$ be a prime, let $m, t_1, t_2, t_t \in \mathbb{N}$, let $\mathbf{l_1} \in \mathbb{F}_p[X_1, ..., X_m]^{t_1}, \mathbf{l_2} \in \mathbb{F}_p[X_1, ..., X_m]^{t_2}, \mathbf{l_t} \in \mathbb{F}_p[X_1, ..., X_m]^{t_t}$ be tuples of polynomials such that $l_1[0] = l_2[0] = l_t[0] = 1$, and let $f \in \mathbb{F}[X_1, ..., X_m]$ be a polynomial. We say that f is dependent on $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_t})$ if there exist integers $\{k_i\}_{i\in[t]}$, $\{k'_i\}_{i\in[t]}$, $\{k''_i\}_{i\in[t]}$ such that

$$f = \sum_{i,j\in[t]} k_i \cdot l_1[i] \cdot k'_j \cdot l_2[j] + \sum_{i\in[t]} k''_i \cdot t[i]$$

If f is not dependent on $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_t})$, we say that it is independent of $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_t})$.

We require the Master Theorem to prove the security of our construction(s).

Theorem A.3 (Master Theorem [28]). *Let $p \in \mathbb{N}$ be a prime, let $m, t_1, t_2, t_3 \in \mathbb{N}$, let $\mathbf{l_1} \in \mathbb{F}_p[X_1, ..., X_m]^{t_1}, \mathbf{l_2} \in \mathbb{F}_p[X_1, ..., X_m]^{t_2}, \mathbf{l_t} \in \mathbb{F}_p[X_1, ..., X_m]^{t_3}$ be tuples of polynomials of maximum degree d_r, d_s and d_t respectively. Let $f \in \mathbb{F}[X_1, ..., X_m]$ be a polynomial of degree at most d_f. Denote $d = \max\{d_r, d_s, d_t, d_f\}$ and $t = t_1 + t_2 + t_3$. If f is independent of $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_t})$, then for any generic ppt adversary $\mathcal{A}$ that makes at most q group oracle queries*

$$\left| Pr\left[\mathcal{A}\begin{pmatrix} p, h_1[\mathbf{l_1}(x)] \\ h_2[\mathbf{l_2}(x)], \\ h_T[f(x)] \end{pmatrix} = 1\right] - Pr\left[\mathcal{A}\begin{pmatrix} p, h_1[\mathbf{l_1}(x)] \\ h_2[\mathbf{l_2}(x)], \\ h_T[r] \end{pmatrix} = 1\right] \right| \leq \frac{(q+t+2)^2 \cdot d}{2p}$$

where h_1, h_2 and h_T denote handles for groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ and the probabilities are taken over the choices of $x \xleftarrow{\$} \mathbb{Z}_p^n$ and $r \xleftarrow{\$} \mathbb{Z}_p$.

B Remaining Definitions for OSTE schemes

Definition B.1 (Correctness for OSTE). We call an OSTE protocol correct if for all $n = \mathsf{poly}(\lambda)$, $t < n$, all $S \subseteq [n]$ with $|S| \geq t$ it holds that

$$\Pr\left[\begin{array}{c} \mathsf{DecAggr}(\mathsf{CRS}, \mathcal{K}, \mathsf{ct}, \{\sigma_i\}_{i\in S}) = \mathsf{m} \\ \wedge\ \forall i \in [n] \colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) = 1 \\ \wedge\ \mathsf{PreVfy}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{K}, \pi) = 1 \end{array}\right] \geq 1 - \mathsf{negl}(\lambda),$$

where $\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$, $(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda)$ for all $i \in [n]$, $(\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]})$, $\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m})$, as well as $\sigma_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$ for each $i \in [n]$.

Definition B.2 (IND-CPA security for OSTE). An OSTE scheme $\mathcal{E}$ is IND-CPA secure if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n) \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

where $\mathsf{IND\text{-}CPA}_{\mathcal{A}}$ is defined in Figure 11.

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

$\mathsf{CRS} \xleftarrow{\$} \mathsf{Setup}(1^\lambda, n)$
$(\mathcal{T}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(1^\lambda, \mathsf{CRS})$
for $i \in [n], i \notin \mathcal{T}$ **do**
  $(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, \mathsf{CRS})$
$(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\in\mathcal{T}}, t, \mathcal{K}, \pi, \mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \xleftarrow{\$} \mathcal{A}(\{\mathsf{pk}_i, \mathsf{hint}_i, \pi_i\}_{i\notin\mathcal{T}}, \mathsf{st})$
$b \xleftarrow{\$} \{0,1\}$
$\mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m}_b)$
$b' \xleftarrow{\$} \mathcal{A}(\mathsf{ct}, \mathsf{st})$
return $b = b' \wedge \mathcal{T} \subseteq [n] \wedge |\mathcal{T}| < t$
  $\wedge\, \forall i \in \mathcal{T}\colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i)$
  $\wedge\, \forall i, j \in [n], i \neq j\colon \mathsf{pk}_i \neq \mathsf{pk}_j$
  $\wedge\, \mathsf{PreVfy}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \mathcal{K}, \pi)$

Figure 11: A game-based IND-CPA security definition for OSTE and TSTE schemes.

C TSTE Correctness and Security Definitions

Definition C.1 (Correctness for TSTE). We call a TSTE scheme correct with respect to parameter λ_c if for all $n = \mathsf{poly}(\lambda)$, $t < n$, all $S \subseteq [n]$ with $|S| \geq t$ and for

$$\begin{gathered}\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda, n);\\ \forall i\ (\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) \xleftarrow{\$} \mathsf{KGen}(1^\lambda);\\ (\mathcal{K}, \pi, \mathsf{tk}) \xleftarrow{\$} \mathsf{Prep}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}, \epsilon);\\ \mathsf{ct} \xleftarrow{\$} \mathsf{Enc}(\mathsf{CRS}, \mathcal{K}, t, \mathsf{m});\\ \forall i\ \sigma_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})\end{gathered}$$

it holds that

$$\Pr\left[\begin{array}{c}\mathsf{DecAggr}(\mathsf{CRS}, \mathcal{K}, \mathsf{ct}, \{\sigma_i\}_{i\in S}) = \mathsf{m}\\ \wedge\, \forall i \in [n]\colon \mathsf{IsValid}(\mathsf{CRS}, \mathsf{pk}_i, \mathsf{hint}_i, \pi_i) = 1\\ \wedge\, \mathsf{PreVfy}(\mathsf{CRS}, \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in n]}, \mathcal{K}, \pi) = 1\\ \wedge\, \forall i \in S\colon \mathsf{PartVfy}(\mathsf{ct}, \sigma_i, \mathsf{pk}_i) = 1\end{array}\right] \geq 1 - \mathsf{negl}(\lambda_c)$$

Security. Like OSTE schemes, TSTE must fulfill a IND-CPA security definition. The security definitions for OSTE and TSTE definitions both concern the Enc function. A definition concerning Enc is sensible in the context of honest encryptions, while a consistent definition for both schemes simplifies our transformation between the two (Section 4.1).

Definition C.2 (IND-CPA security for TSTE). We call a TSTEP scheme $\mathcal{E}$ IND-CPA secure, if for all PPT adversaries $\mathcal{A}$ and $n = \mathsf{poly}(\lambda)$

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\mathcal{E}}(1^\lambda, n) \leq \mathsf{negl}(\lambda)$$

where $\mathsf{IND\text{-}CPA}_{\mathcal{A}}$ is defined in Figure 11.

D TSTE Correctness and Security Proofs

Lemma D.1 *Consider scheme Π_{TSTE} defined in Section 4.1 for $M = \mathsf{poly}(\lambda)$. Let $\mathcal{E}$ denote the underlying OSTE scheme. If $\mathcal{E}$ fulfills security, Π_{TSTE} is secure.*

Proof. The security of Π_{TSTE} is directly implied by the security of $\mathcal{E}$. Recall that both OSTE and TSTE schemes employ the same security definition. Further, our construction instantiates all functions of Π_{TSTE} with their counterpart in $\mathcal{E}$ if such a counterpart exists. Thus, Π_{TSTE} must be secure. ∎

E Full Traceability Proof for Π_{TSTE}

In the following, we prove Lemma 4.6.

Proof. Our proof takes inspiration from [11] but considers the decoder-based definitions of indistinguishability and membership-hiding. Compared to tracing schemes building on classical PLBE, adversaries against TSTE schemes can use the preprocessing output. This allows them to build decoders with a small but non-negligible advantage in the indistinguishability and membership hiding games. Consequently, our proof does not require the indistinguishability advantage δ_1 or the membership hiding advantage δ_2 to be negligible. This is a major difference to classical PLBE constructions and a requirement when building a TSTE scheme from our construction in the preprocessing model. We show that it is possible to achieve traceability nonetheless by imposing upper bounds of $\delta_1 \leq \frac{\epsilon}{16}$ and $\delta_2 \leq \frac{\epsilon}{16n}$.

Consider a decoder box $\mathcal{D}$ in the traceability game GameTR (Figure 4) and a fixed order of revocation of parties. Let p_i denote the advantage that $\mathcal{D}$ has in distinguishing between the encryption of two messages m_0 and m_1 for $\mathcal{R}_i$, meaning that the first i parties of the revocation order have been revoked. Let $\hat{p}_i$ denote the approximation of p_i measured by the Trace algorithm using W queries and let $\mathcal{T}'$ be the set returned by the algorithm. In the following, denote the number of iterations of the tracing algorithm by W. For Π_{TSTE}, we set $W = 2^7 \cdot 3n^2 \frac{\lambda}{\epsilon}$.

In any GameTR execution that is won by the adversary, the Trace algorithm either does not trace any party ($\mathcal{T}' = \emptyset$) or traces an honest party ($\mathcal{T}' \not\subseteq \mathcal{T}$). In the first case, by the conditions of the Trace algorithm (Figure 5) presented in Figure 5, it must hold that $\hat{p}_{i-1} - \hat{p}_i < \frac{\epsilon}{4(n-t)}$ for all $i \in [n-t]$ (Condition 1) and $\hat{p}_{n-t} < \frac{\epsilon}{4}$ (Condition 2). As a result of the small success probability remaining after $n-t$ revocation steps and the bounded difference between success probabilities in each step, the initial advantage $\hat{p}_0$ must be small. For the quality of the decoder ϵ, it holds that

$$\begin{aligned}\epsilon - \hat{p}_0 =& \epsilon - \hat{p}_{n-t} - \sum_{i=1}^{n-t} (\hat{p}_{i-1} - \hat{p}_i) \\ >& \epsilon - (n-t)\frac{\epsilon}{4(n-t)} - \frac{\epsilon}{4} > \frac{\epsilon}{4}\end{aligned}$$

If $\mathcal{D}$ wins the traceability game by framing an honest party, it either holds that $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$ and one of the last t parties to be revoked is honest, or $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ for an honest i. If $\mathcal{D}$ is successful, at least one of the following statements must hold:

1. $\mathcal{T}' = \emptyset$ and $\epsilon - \hat{p}_0 > \frac{\epsilon}{4}$.
2. $\mathcal{T}' \not\subseteq T$ and $\hat{p}_{n-t} \geq \frac{\epsilon}{4}$.
3. $\mathcal{T}' \not\subseteq T$ and $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ for an honest i.

In the rest of the proof, we treat cases (1), (2), and (3) separately and show that they impose a contradiction to the indistinguishability, message privacy, and membership hiding properties of $\mathcal{E}$. For each case, we use the traceability adversary $\mathcal{A}$ to construct an adversary that breaks one of the properties if $\mathcal{A}$ has non-negligible advantage.

Consider case (1). Given an adversary $\mathcal{A}$ in GameTR, we build an adversary $\mathcal{A}_1$ playing the indistinguishability game $\mathsf{GameInd}^{\mathsf{OSTE}}_{\mathcal{A}_1}(1^\lambda, n, \delta_1)$ as defined in Figure 2. We then show that $\mathcal{A}_1$ breaks indistinguishability if $\mathcal{A}$ has non-negligible advantage. Given a decoder $\mathcal{D}$ output by $\mathcal{A}$, $\mathcal{A}_1$ builds a decoder box $\mathcal{D}'$ that distinguishes between Enc encryptions and TrEnc encryptions for $\mathcal{R} = \emptyset$ with advantage at least δ_1. $\mathcal{A}_1$ proceeds as follows:

- Obtain CRS from the challenger, generate hints, relay the necessary information to $\mathcal{A}$ and obtain hints, and t in return, obtain $\mathcal{K}$ and relay it to $\mathcal{A}$.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Obtain traceability decoder $\mathcal{D}$ and messages $\mathsf{m}_0, \mathsf{m}_1$ from $\mathcal{A}$.
- Output $(\mathcal{D}', \mathsf{m}_b)$ where $\mathcal{D}'(\mathsf{ct}) = \begin{cases} 0 & \text{if } \mathcal{D}(\mathsf{ct}) = b \\ 1 & \text{else} \end{cases}$.

To analyze the success probability of $\mathcal{A}_1$, consider that $\mathcal{A}$ outputs an ϵ-$\mathrm{Dist}^{\mathsf{m}_0,\mathsf{m}_1}$ decoder with probability $\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}(1^\lambda, n, \epsilon)$. In case (1), it holds that $\epsilon - \hat{p}_0 > \frac{\epsilon}{4}$. Using the Chernoff bound, we find that

$$\Pr\left[p_0 - \hat{p}_0 > \frac{\epsilon}{8}\right] \leq 2e^{-W\frac{\epsilon^2}{8^2 \cdot 3 \cdot p_0}} \leq 2e^{-\lambda}$$

which is negligible. It holds that $\epsilon - p_0 \geq \frac{\epsilon}{8}$ with overwhelming probability. Given an Enc ciphertext, $\mathcal{D}'$ outputs the correct result with probability ϵ. For ciphertexts computed using TrEnc this probability is $(1-p_0)$, as $\mathcal{D}'$ only outputs 1 if the result of $\mathcal{D}$ is incorrect. This results in an overall success probability of $\frac{1}{2}\epsilon + \frac{1}{2}(1-p_0) = \frac{1}{2} + \frac{\epsilon - p_0}{2}$. With overwhelming probability, it holds that $\frac{1}{2} + \frac{\epsilon - p_0}{2} - \frac{1}{2} \geq \frac{\epsilon}{16} \geq \delta_1$. Overall, we find that:

$$\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}(1^\lambda, n, \epsilon) \leq \mathbf{Adv}^{\mathsf{GameInd}}_{\mathcal{A}_1,\mathcal{E}}(1^\lambda, n, \delta_1) + \mathsf{negl}(\lambda)$$

We have shown by reduction that the first case contradicts the fact that $\mathcal{E}$ fulfills indistinguishability with parameter δ_1.

Consider now case (2). Using $\mathcal{A}$ we build an adversary $\mathcal{A}_2$ playing the message privacy game $\mathsf{GameMP}_{\mathcal{A}_2}(1^\lambda, n)$ as defined in Figure 3. We show that $\mathcal{A}_2$ breaks the message privacy of $\mathcal{E}$ if $\mathcal{A}$ has non-negligible probability. $\mathcal{A}_2$ proceeds as follows:

- Obtain CRS from the challenger, relay information to $\mathcal{A}$, and obtain $\mathcal{T}$ and hints for compromised parties in return. Obtain $\mathcal{K}$ from the challenger and relay it to $\mathcal{A}$.
- Obtain traceability decoder $\mathcal{D}$ and messages $\mathsf{m}_0, \mathsf{m}_1$ from $\mathcal{A}$.
- Sample a bit β.
- Obtain the encryption ct' of m_β from the TrEnc oracle. Obtain the challenge ciphertext ct.
- Return $b' \leftarrow \begin{cases} \beta & \text{if } \mathcal{D}(\mathsf{ct}) = \mathcal{D}(\mathsf{ct}') \\ 1-\beta & \text{else} \end{cases}$.

In case (2), it holds that $\hat{p}_{n-t} > \frac{\epsilon}{4}$. $\mathcal{A}_2$ wins GameMP if it has a non-negligible advantage in distinguishing between encryptions of m_0 and m_1. By the Chernoff bound, it holds that

$$\Pr\left[p_{n-t} - \hat{p}_{n-t} > \frac{\epsilon}{8}\right] \leq 2e^{-W\frac{\epsilon^2}{8^2 \cdot 3 \cdot p_0}} \leq 2e^{-\lambda}$$

whereby $p_{n-t} > \frac{\epsilon}{8}$ with overwhelming probability. If $\mathcal{D}$ is a good decoder, $\mathcal{A}_2$ wins the message privacy game if both calls to $\mathcal{D}$ yield the correct result or if both are wrong. This occurs with a probability of $\left(\frac{1}{2} + \frac{\epsilon}{8}\right)^2 + \left(\frac{1}{2} - \frac{\epsilon}{8}\right)^2 = \frac{1}{2} + 2\left(\frac{\epsilon}{8}\right)^2$. As $\frac{\epsilon}{8}$ is non-negligible, this results in a non-negligible advantage of breaking message privacy. $\mathcal{A}_2$ produces a good decoder with probability $\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}(1^\lambda, n, \epsilon)$, which in turn is non-negligible. Note that in case (2), we assume that one of the last t parties to be revoked is honest and is framed by the adversary. Because of this, $\mathcal{A}_2$ wins GameMP if it sets $\mathcal{R} = [n-t]$ as less than t of its shares are used for decryption. We find that

$$\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}(1^\lambda, n, \epsilon) \leq \mathbf{Adv}^{\mathsf{GameMP}}_{\mathcal{A}_2,\mathcal{E}}(1^\lambda, n) + \mathsf{negl}(\lambda)$$

We have shown a contradiction to the message privacy of $\mathcal{E}$ in the second case.

Finally, consider case (3). We build an adversary $\mathcal{A}_3$ from $\mathcal{A}$ that plays the membership hiding game $\mathsf{GameMH}^{\mathsf{OSTE}}_{\mathcal{A}_3}(1^\lambda, n, \delta_2)$, depicted in Figure 2. We then show that $\mathcal{A}_3$ breaks membership hiding if $\mathcal{A}$ has a non-negligible advantage. To do so, $\mathcal{A}_3$ produces a decoder box $\mathcal{D}'$ that is a δ_2-Dist$^{\mathcal{R},\mathcal{R}\cup\{i\}}$ as follows

- Obtain CRS from the challenger, relay information to $\mathcal{A}$, obtain hints for compromised parties from $\mathcal{A}$, choose i as one of the honest indices, and sample the remaining hints.
- Relay $\mathcal{K}$ to $\mathcal{A}$ and obtain $\mathcal{D}$, m_0, m_1 in return.
- Sample $b \xleftarrow{\$} \{0,1\}$.
- Return $(\mathcal{D}', \mathsf{m}_b)$ where $\mathcal{D}'(\mathsf{ct}) = \begin{cases} 0 & \text{if } \mathcal{D}(\mathsf{ct}) = b \\ 1 & \text{else} \end{cases}$.

In case (3) it holds that $\hat{p}_{i-1} - \hat{p}_i \geq \frac{\epsilon}{4(n-t)}$ for an honest i. Using the Chernoff bound, we find that $p_{i-1} - p_i \geq \frac{\epsilon}{8(n-t)}$ with overwhelming probability, as

$$\begin{aligned}\Pr\left[p_{i-1} - p_i < \frac{\epsilon}{8(n-t)}\right] \leq & \Pr\left[p_i - \hat{p}_i \geq \frac{\epsilon}{16(n-t)}\right] \\ \leq & e^{-W\frac{2\epsilon^2}{(16(n-t))^2 \cdot 3 \cdot p_i}} \leq \mathsf{negl}(\lambda)\end{aligned}$$

The probability that this statement holds for the same i that $\mathcal{A}_3$ chooses in GameMH is $\frac{1}{n-t}$. Then, $\mathcal{D}'$ returns 0 if i is not revoked with probability p_{i-1} and 1 for revoked i with probability $1 - p_i$, since it only outputs i if $\mathcal{D}$'s output is incorrect. Overall, $\mathcal{D}'$ returns the correct result with probability $\frac{1}{2}p_{i-1} + \frac{1}{2}(1 - p_i) = \frac{1}{2} + \frac{p_{i-1} - p_i}{2}$. With overwhelming probability, it holds that $\frac{1}{2} + \frac{p_{i-1} - p_i}{2} - \frac{1}{2} \geq \frac{\epsilon}{16(n-t)} \geq \delta_2$. In consequence:

$$\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}(1^\lambda, n, \epsilon) \leq (n-t)\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A}_3,\mathcal{E}}(1^\lambda, n, \delta_2) + \mathsf{negl}(\lambda)$$

We have shown a contradiction to the membership hiding property of $\mathcal{E}$ in case (3).

Combining all three results, we find that any one of the three statements that are implied by the existence of a traceability adversary $\mathcal{A}$ with non-negligible success probability implies a

contradiction to the indistinguishability, message privacy or membership hiding properties of the underlying OSTE scheme $\mathcal{E}$. It holds that

$$\begin{aligned}\mathbf{Adv}^{\mathsf{GameTR}}_{\mathcal{A},\Pi_{\mathsf{TSTE}}}\left(1^\lambda, n, \epsilon\right) \leq & \mathbf{Adv}^{\mathsf{GameInd}}_{\mathcal{A}_1,\mathcal{E}}\left(1^\lambda, n, \delta_1\right) + \mathbf{Adv}^{\mathsf{GameMP}}_{\mathcal{A}_2,\mathcal{E}}\left(1^\lambda, n\right) \\ & + (n-t)\mathbf{Adv}^{\mathsf{GameMH}}_{\mathcal{A}_3,\mathcal{E}}\left(1^\lambda, n, \delta_2\right) + \mathsf{negl}(\lambda).\end{aligned}$$

■

F Remaining linear constraints of STE [28]

We briefly elaborate on the remaining constraints and how Garg et al. linearly verify them.

Verifying that $\deg(Q_x) \leq |\mathbb{H}| - 2$. To comply with the sumcheck lemma (Lemma 2.1), one must ensure that Q_x has degree at most $|\mathbb{H}| - 2$ To prove that Q_x has degree at most $|\mathbb{H}| - 2$ for $|\mathbb{H}| = n + 1$ linearly, we require the aggregator to evaluate a polynomial $\hat{Q}_x(X) = Q_x(X) \cdot X$ at position τ. The corresponding equation is

$$[Q_x(\tau)]_1 \circ [\tau]_2 = [\hat{Q}_x(\tau)]_1 \circ [1]_2.$$

This ensures that Q_x is of maximum degree $|\mathbb{H}| - 2 = n - 1$ because if Q_x were of higher degree, say n, then $\hat{Q}_x$ would be of degree $n + 1$, which the aggregator can not interpolate at position τ since the CRS only reaches up to $[\tau^n]$. Before we can understand the check that the set B is authorized, we mention that we always include a dummy party at position 0 and assume that its secret key is also 0.

Forcing non-trivial witnesses. Furthermore, we force the aggregator to interpolate $B(\omega^0) = 1$ (i.e., a dummy party must be included) to ensure that B is non-zero. This requirement can be linearly verified through a KZG opening proof, i.e., we force the verifier to compute $[Q_0(\tau)]_1$ for a polynomial Q_0 such that $B(X) - 1 = Q_0(X)(X - \omega^0)$. The verification is the linear KZG verification:

$$[1]_1 \circ [B(\tau)]_2 = [Q_0(\tau)]_1 \circ [\tau - \omega^0]_2 + [1]_T$$

Ensuring that B is authorized. To ensure that B is authorized (i.e., the threshold is met), Garg et al. again use a degree check. In particular, they force the aggregator to compute $[\hat{B}(\tau)]_1$ such that $\hat{B}(X) = B(X) \cdot X^t$. The reasoning behind this is that the aggregator has to set $B(\omega^i) = 0$ for every party i that it does not know a partial decryption of, as the sumcheck equation enforces to aggregate the public keys correctly, and the signature check ensures that the aggregated signature is valid under aPK. Further, the aggregator has to set $B(\omega^0) = 1$ to compute the opening proof for the dummy party. Assuming that the aggregator has an insufficient amount of partial signatures, say $t - 1$, then B would have to be 0 in $n - (t - 1)$ positions and 1 at position ω^0, which means that the aggregator has to interpolate B with respect to $n - t + 2$ positions. The degree of B is therefore $n - t + 1$, which means the degree of $\hat{B}$ is $n + 1$. Hence, the aggregator with insufficient partial signatures is unable to evaluate $[\hat{B}(\tau)]$, as the CRS only contains powers-of-τ up to $[\tau^n]_{1,2}$. An aggregator with t partial signatures, however, would have a polynomial B of degree $n - t$, which means it is able to evaluate $[\hat{B}(\tau)]_1$, as $\hat{B}$ is of degree n. The corresponding linear verification equation is

$$[B(\tau)]_2 \circ [\tau^t]_1 = [\hat{B}(\tau)]_1 \circ [1]_2.$$

G Remaining Algorithms of Π_{OSTE}

In Section 5.2, we already introduced the most important algorithms, i.e., the re-randomization Rand, encryptions Enc and TrEnc, the partial decryption $\mathsf{PartDec}$, and the proof system $\mathsf{PS}_{\mathsf{KGen}}$.

For completeness, we present the remaining algorithms Setup, KGen, $\mathsf{IsValid}$, Prep, PreVfy, and $\mathsf{DecAggr}$. Note that they are largely similar to those of STE in [28].

Setup. The setup algorithm samples a random $\tau \xleftarrow{\$} \mathbb{Z}_p$ and generate the CRS as

$$([1]_{1,2}, [\tau]_{1,2}, [\tau^2]_{1,2}, \ldots, [\tau^n]_{1,2}).$$

Key Generation. In KGen, a party samples a random $\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_p^*$ and computes the public key as $\mathsf{pk} \leftarrow [\mathsf{sk}]_1$ and $\mathsf{hint} \leftarrow ([\mathsf{sk}\tau]_1, [\mathsf{sk}\tau^2]_1, [\mathsf{sk}\tau^n]_1))$. Additionally, it generates a proof in $\mathsf{PS}_{\mathsf{KGen}}$ as $\pi \xleftarrow{\$} \mathsf{PS}_{\mathsf{KGen}}.\mathsf{Prove}((\mathsf{CRS}, \mathsf{pk}, \mathsf{hint}), \mathsf{sk})$. To validate the correctness of the public key and hint, the $\mathsf{IsValid}$ algorithm runs $\mathsf{PS}_{\mathsf{KGen}}.\mathsf{Verify}((\mathsf{CRS}, \mathsf{pk}, \mathsf{hint}), \pi)$.

Preprocessing. Our Π_{OSTE} construction has no preprocessing and supports public tracing. Hence we just set $\mathcal{K} := \{\mathsf{pk}_i, \mathsf{hint}_i\}_{i\in[n]}$ and the tracing key $\mathsf{tk} := \bot$. Accordingly, the PreVfy algorithm always outputs 1.

$\mathsf{DecAggr}(\mathsf{CRS}, \mathsf{ct}, \{\sigma_i\}_{i\in S})$

$([\gamma]_2, \mathsf{ak}, \mathsf{ct}_3, \mathsf{ct}_4) \leftarrow \mathsf{ct}$

Interpolate B on $\{(\omega^0, 1), (\omega^i, 0)_{i[n]\setminus S}\}$

$$[B(\tau)]_2 \leftarrow \left[\sum_{j=0}^{n} B(\omega^j) L_j(\tau)\right]_2$$

$$[\hat{B}(\tau)]_1 \leftarrow [\tau^t B(\tau)]_1$$

$$\mathsf{aPK} \leftarrow \frac{1}{n+1}\left(\sum_{i\in S} B(\omega^i)\mathsf{ak}_{i,0} + [1]_1\right)$$

$$\sigma^* \leftarrow \frac{1}{n+1}\left(\sum_{i\in S} B(\omega^i)\sigma_i + [\gamma]_2\right)$$

$$[Q_x(\tau)]_1 \leftarrow \sum_{i=0}^{n} B(\omega^i)\mathsf{ak}_{i,2}$$

$$[\hat{Q}_x(\tau)]_1 \leftarrow \sum_{i=0}^{n} B(\omega^i)\mathsf{ak}_{i,1}$$

$$[Q_0(\tau)]_1 \leftarrow \left[\frac{B(\tau)-1}{\tau-\omega^0}\right]_1$$

$$[Q_Z(\tau)]_1 \leftarrow \sum_{i=0}^{n} B(\omega^i)\mathsf{ak}_{i,3} + \sum_{i=0}^{n} B(\omega^i)\mathsf{ak}_{i,4}$$

$$w \leftarrow ([B(\tau)]_2, -\mathsf{aPK}, -[Q_Z(\tau)]_1, -[Q_x(\tau)]_1, [\hat{Q}_x(\tau)]_1, \sigma^*, -[\hat{B}(\tau)]_1, -[Q_0(\tau)]_1)^\intercal$$

return $\mathsf{m} \leftarrow \mathsf{ct}_3 - \mathsf{ct}_2 \circ w$

Figure 12: The aggregation algorithm of Π_{OSTE}.

DecAggr. The aggregation algorithm is identical to that of STE but uses the re-randomized aggregation key ak that is part of the ciphertext instead of a global one. The full algorithm is shown in Figure 12.

H Full Message Privacy Proof for Π_{OSTE}

In the following, we present the detailed proof of Lemma 5.2.

Proof. This proof resembles the proof presented in the original paper by Garg et al. [28]. For completeness, we give the full details of the proof, accounting for changes in the encryption scheme, the definition, and the additional group elements the adversary is presented with. We show the lemma via a reduction to the master theorem as presented in Theorem A.3. The master theorem holds in the generic group model [28], and we present a reduction to it for generic adversaries. We present a series of game hops, simplifying the message privacy game before finally reducing it to the master theorem. In particular, we consider the following games:

Game_0: We define $\mathsf{Game}_0 \leftarrow \mathsf{GameMP}$.

Game_1: Game_1 is equivalent to Game_0, but the adversary does not get access to a TrEnc oracle.

Game_2: Game_2 is equivalent to Game_1, but we replace the proofs π attached to the hints by simulated proofs as $\mathsf{Sim}(\mathsf{td}, \mathsf{pk}_i, \mathsf{hint}_i)$ where Sim is the simulator of the proof system PS, executed on a trapdoor td.

We show

$$\begin{aligned}\mathbf{Adv}^{\mathsf{GameMP}}_{\mathcal{A}}(1^\lambda, n) &= \mathbf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(1^\lambda, n)\\ &\overset{H.1}{\leq} \mathbf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda, n) + \mathsf{negl}(\lambda)\\ &\overset{H.2}{\leq} \mathbf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda, n) + \mathsf{negl}(\lambda)\\ &\overset{H.3,H.4}{\leq} \mathsf{negl}(\lambda)\end{aligned}$$

In the final step of our proof, we present our reduction to the master theorem (Lemma H.3) and prove that our parameters for the master theorem are linearly independent as per Definition A.2 (Lemma H.4). ∎

Lemma H.1 *For all PPT adversaries $\mathcal{A}$, for $n = \mathsf{poly}(\lambda)$, and considering scheme Π_{OSTE}, there exists a PPT adversary $\mathcal{B}$ such that*

$$\boldsymbol{Adv}^{\mathsf{Game}_0}_{\mathcal{A},\Pi_{\mathsf{OSTE}}}(1^\lambda, n) = \boldsymbol{Adv}^{\mathsf{Game}_1}_{\mathcal{B},\Pi_{\mathsf{OSTE}}}(1^\lambda, n)$$

Note that there is no private tracing key in the context of Π_{OSTE}. Therefore, any adversary can compute TrEnc on their own. More formally, for any adversary $\mathcal{A}$ in Game_0, there exists an adversary $\mathcal{B}$ in Game_1 which performs the same steps as $\mathcal{A}$ but evaluates the TrEnc function on its own instead of calling the oracle. As both obtain the same result, their success probability must be the same.

Lemma H.2 *For a PPT adversary $\mathcal{A}$, for $n = \mathsf{poly}(\lambda)$, and considering scheme Π_{OSTE} instantiated with a zero-knowledge proof system $\mathsf{PS}_{\mathsf{Rand}}$, it holds that Game_1 and Game_2 are computationally indistinguishable.*

The computational indistinguishability follows directly from the computational indistinguishability of a real and a simulated proof as stated by the zero-knowledge property of $\mathsf{PS}_{\mathsf{Rand}}$.

Lemma H.3 *Consider a generic PPT adversary $\mathcal{A}$ that makes up to q queries to the group oracle. Let $n = \mathsf{poly}(\lambda)$ and consider construction Π_{OSTE} defined in Section 5. Let $\mathcal{R} \in \mathcal{X}$ be a set of revoked parties, and let $\mathcal{T}$ be the set of parties compromised by the adversary. We show that for each choice of $\mathcal{R}$ and $\mathcal{T}$, the following statement holds:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{\mathsf{Game}_2}(1^\lambda, n) \leq \mathsf{negl}(\lambda)$$

Proof. Consider an adversary $\mathcal{A}$ in Game_2. Recall the master theorem as defined in Theorem A.3 for the following choice of polynomial lists:

$$\begin{aligned}
\mathbf{l_1} &= \Big(\alpha, s_1\mathsf{ek} + s_4\tau^t + s_5, s_3\alpha, \big[\tau^i, [\mathsf{sk}_j\tau^i]_{j\in[n]}, [\alpha\mathsf{sk}_j\tau^i]_{j\in\mathcal{V}}, [\bar{\alpha}_j\mathsf{sk}_j\tau^i]_{j\in\mathcal{R}}\big]_{i\in[0,n]}\Big)\\
\mathbf{l_2} &= \big(s_1 + s_3\gamma, s_1Z(\tau), s_1\tau + s_2\tau, s_2, s_4, s_5Z_0, \gamma, [\tau^i]_{i\in[0,n]}\big)\\
\mathbf{l_T} &= (1)\\
f &= s_5
\end{aligned}$$

where $\mathbf{l_1}, \mathbf{l_2}$ and $\mathbf{l_T}$ are lists of polynomials in $y = (\tau, \gamma, \alpha, \bar{\alpha}_1, \ldots, \bar{\alpha}_n, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, s_1, \ldots, s_5)$, $\mathcal{V} = [n] \setminus \mathcal{R}$ is the set of non-revoked (valid) parties and $\mathcal{H} = [n] \setminus \mathcal{T}$ is the set of honest parties. Note that the polynomial lists implicitly also depend on $\mathcal{R}$ and $\mathcal{T}$, yet these values are known to the adversary. We consider ek with respect to specific $\mathcal{R}$ and $\mathcal{T}$ in our proof. We show, by reduction, that an adversary $\mathcal{A}$ in Game_2 can be used to distinguish between f and random input in the master theorem, thereby upper-bounding $\mathcal{A}$'s success probability by a negligible function. We construct a distinguisher $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine to achieve that.

First, we motivate our choices of $\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T}$, and f for our instantiation of the master theorem. In the context of our reduction, $\mathcal{B}$ wants to use $\mathcal{A}$ to gain significant advantage in distinguishing $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T}, f)$ from $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T}, r)$ for the instantiations given above. To do so, $\mathcal{B}$ obtains a challenge and uses it to simulate Game_2 to $\mathcal{A}$ perfectly. All information that is required to do so is included in the polynomial lists. While the first polynomials in lists $\mathbf{l_1}$ and $\mathbf{l_2}$ correspond to the components of the ciphertext itself, the latter polynomials correspond to the information in the CRS, hints, and rerandomized hints.

Adversary $\mathcal{B}$ simulates Game_2 as follows: Given the challenge $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T})$, $\mathcal{B}$ relays labels for elements $[\tau^i]_1$ for $i \in [0, n]$ to $\mathcal{A}$ as the CRS. $\mathcal{B}$ also gives a list of labels for the public keys and hints of honest parties, $[\mathsf{sk}_i\tau^j]_{i\in\mathcal{H}, j\in[0,n]}$ to $\mathcal{A}$. As defined in Game_2, $\mathcal{B}$ simulates the honest parties' hints, using the simulator of proof system $\mathsf{PS}_{\mathsf{KGen}}$. $\mathcal{B}$ then outputs a list of labels for the hints of corrupted parties. As $\mathcal{A}$ can control the group oracle, they can assign the labels of the adversary to the specific hints given in $\mathbf{l_1}$. Then, $\mathcal{A}$ computes the preprocessing output, including the encryption key in $\mathbf{l_1}$ and its corresponding aggregation key. Note that both these keys can be computed knowing $\mathcal{R}$, $\mathcal{T}$, and the elements of $\mathbf{l_1}$, as described in the Rand function (Figure 6).

Finally, after obtaining the preprocessing output, $\mathcal{A}$ returns two messages $\mathsf{m}_0, \mathsf{m}_1$ to be encrypted. $\mathcal{B}$ samples a random bit β and encrypts message m_β as $([\gamma]_2, \mathsf{ak}, s^\intercal \cdot A, s_5 + \mathsf{m}_\beta)$. The components of $s^\intercal A$ are taken from $\mathbf{l_1}$ and $\mathbf{l_2}$, $[\gamma]_2$ is taken from $\mathbf{l_2}$ and ak refers to the corresponding ak computed during preprocessing.

Finally, the reduction outputs 0 if $\mathcal{A}$ produces the correct result and 1 otherwise. As $\mathcal{B}$ perfectly simulates Game_2 to $\mathcal{A}$, $\mathcal{B}$ has at least the same success probability as $\mathcal{A}$ when distinguishing f from random. However, the master theorem states that for independent polynomial lists, a polynomial number of group oracle queries, polynomial size, and order of the polynomial lists, the advantage of any adversary in distinguishing the two cases should only be negligible. We show that the polynomial lists $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T})$ are independent in Lemma H.4. This implies that $\mathbf{Adv}_{\mathcal{A},\Pi_{\mathsf{OSTE}}}^{\mathsf{Game}_2}(1^\lambda, n)$ can only be negligible. ∎

Lemma H.4 *Let*

$$\begin{aligned}
\mathbf{l_1} &= \Big(\alpha, s_1\mathsf{ek} + s_4\tau^t + s_5, s_3\alpha, \big[\tau^i, [\mathsf{sk}_j\tau^i]_{j\in[n]}, [\alpha\mathsf{sk}_j\tau^i]_{j\in\mathcal{V}}, [\bar{\alpha}_j\mathsf{sk}_j\tau^i]_{j\in\mathcal{R}}\big]_{i\in[0,n]}\Big)\\
\mathbf{l_2} &= \big(s_1 + s_3\gamma, s_1Z(\tau), s_1\tau + s_2\tau, s_2, s_4, s_5Z_0, \gamma, [\tau^i]_{i\in[0,n]}\big)\\
\mathbf{l_T} &= (1)\\
f &= s_5
\end{aligned}$$

be lists of polynomials in $\mathbb{Z}_p$. It holds that $\mathbf{l_1}, \mathbf{l_2}$ and $\mathbf{l_T}$ are independent of f, according to Definition A.2.

Proof. This proof proceeds similarly to the independence proof for the original silent setup scheme in [28]. Towards a contradiction, we assume that a linear combination of group elements exists that allows the adversary to compute $f = s_5$. We then extract requirements on the values of some of the combinations' coefficients to reach a contradiction to the univariate sumcheck, as detailed in Lemma 2.1. In detail, this involves using partial derivatives to extract the five equations detailed in Section 5.1 from the linear combination that need to be fulfilled for a correct decryption.

Intuitively, these equations imply an upper bound on the degree of polynomial B that an adversary multiplies with the first component of the ciphertext. This limitation expresses the lower bound of t parties required for decryption. At the same time, the univariate sumcheck, in combination with the five equations, imposes a lower bound on the degree of B, expressing the constraint of the number of valid shares the adversary actually has. The lower bound is larger than the upper bound, which yields a contradiction.

More formally, let $\mathtt{1}^n$ denote the n-dimensional unit vector and let $\mathbf{1} = \mathtt{1}^{3+(2n+1)(n+1)}$ and $\mathbf{1'} = \mathtt{1}^{7+(n+1)}$. Furthermore, let $\odot$ denote the Hadamard product (point-wise multiplication) for matrices and let $a_{i,*}$ and $a_{*,j}$ denote the row i and column j of a matrix A, respectively. Assume towards contradaction that the polynomial lists $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T}, \mathbf{f})$ are linearly dependent. Then there exist coefficients $K = (k_{i,j}) \in \left(\mathbb{Z}_p^{3+(2n+1)(n+1)} \times \mathbb{Z}_p^{7+(n+1)}\right), k_T \in \mathbb{Z}_p$, such that

$$s_5 = \mathbf{1}(K \odot \mathbf{l_1}^\intercal \cdot \mathbf{l_2})\mathbf{1'}^\intercal + k_T \cdot \mathbf{l_T}$$

This linear combination includes all possible combinations of group elements in $\mathbf{l_1}$ and $\mathbf{l_2}$ that might be used to compute s_5. Consider now the differential:

$$\frac{\partial}{\partial s_i} s_5 = \frac{\partial}{\partial s_i} \mathbf{1}(K \odot \mathbf{l_1}^\intercal \cdot \mathbf{l_2})\mathbf{1'} + k_T \cdot \mathbf{l_T}$$

for $i \in [1, ..., 5]$. Differentiating with respect to $s_1, \ldots, s_5$ eliminates all terms that do not contain the variable, simplifying the expression to the following five equations (compare Section 5.1):

$$0 = k_{2,*}\mathbf{l_2}^\intercal\mathsf{ek} + k_{*,1}\mathbf{l_1}^\intercal + k_{*,2}\mathbf{l_1}^\intercal Z(\tau) + k_{*,3}\mathbf{l_1}^\intercal\tau \tag{1}$$
$$0 = k_{*,3}\mathbf{l_1}^\intercal\tau + k_{*,4}\mathbf{l_1}^\intercal \tag{2}$$
$$0 = k_{*,1}\mathbf{l_1}^\intercal\gamma + k_{3,*}\mathbf{l_2}^\intercal\alpha \tag{3}$$
$$0 = k_{2,*}\mathbf{l_2}^\intercal\tau^t + k_{*,5}\mathbf{l_1}^\intercal \tag{4}$$
$$1 = k_{2,*}\mathbf{l_2}^\intercal + k_{*,6}\mathbf{l_1}^\intercal Z_0 \tag{5}$$

The adversary can compute s_5 only if there are coefficients such that all the above equations are fulfilled. Observe that $k_{i,j} = 0$ for all summands that consist of a product of variables unique in their equation. For instance, $k_{i,j} = 0$ for a term $k_{i,j}\alpha\gamma\tau$ if $\alpha\gamma\tau$ appears nowhere else in the

equation. Such unique combinations of variables cannot be expressed as a linear combination of other summands. The same might be observed by considering the derivative of the equation in all combination variables.

With this observation, we impose constraints on the coefficient vectors, eliminating terms from the equations one after the other. For each equation, we mark summands that are eliminated by constraints of the observed equation in blue and eliminations by constraints of previously observed equations in grey. To simplify notation, we abbreviate the hint terms as follows:

$$H_i = \left[\left[\mathsf{sk}_j\tau^i\right]_{j\in[n]}, \left[\alpha\mathsf{sk}_j\tau^i\right]_{j\in\mathcal{V}}, \left[\bar{\alpha}_j\mathsf{sk}_j\tau^i\right]_{j\in\mathcal{R}}\right]$$

Expanding equation (2), we find:

$$\begin{aligned}0 =& k_{*,3}\tau\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,\not{n}]}^{n-1}\right)^\intercal\\ &+k_{*,4}\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n]}\right)^\intercal,\end{aligned}$$

where $k_{*,3}\mathbf{l_1}^\intercal$ can only contain factors of τ^i where $i<n$. Summands that contain τ^{n+1} are unique in this equation and must have a coefficient of 0. This captures the degree-check property this equation aims to enforce. Expanding equation (3) yields:

$$\begin{aligned}0 =& k_{*,1}\gamma\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n]}\right)^\intercal\\ &+k_{3,*}\alpha\left(s_1+s_3\gamma, s_1Z(\tau), s_1\tau+s_2\tau, s_2, s_4, s_5Z_0, \gamma, \left[\tau^i\right]_{i\in[0,n]}\right)^\intercal.\end{aligned}$$

An adversary can only use terms containing $\gamma\alpha$ and constants to solve this equation. This captures the validity of the aggregated signature. Notably, they cannot validate an aggregated signature using the public keys and hints of parties they do not control. From equation (4), it follows that:

$$\begin{aligned}0 =& k_{2,*}\tau^t\left(s_1+s_3\gamma, s_1Z(\tau), s_1\tau+s_2\tau, s_2, s_4, s_5Z_0, \gamma, \left[\tau^i\right]_{i\in[0,\not{n}]}^{n-t}\right)^\intercal\\ &+k_{*,5}\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n]}\right),\end{aligned}$$

again limiting the maximum degree of τ^i terms that may be used to fulfill this equation. In particular, $k_{2,*}\mathbf{l_2}^\intercal$ cannot contain τ^i terms for $i>n-t$, enforcing a degree check on $k_{2,*}\mathbf{l_2}^\intercal$. This ensures that an adversary must use at least t shares to decrypt correctly.

By equation (5) it holds that $1 = k_{2,*}\mathbf{l_2}^\intercal + k_{*,6}(\tau-1)\mathbf{l_1}^\intercal$. This implies that $k_{2,*}\mathbf{l_2}^\intercal = 1$ if $\tau = 1$. Therefore, $k_{2,*}\mathbf{l_2}^\intercal$ cannot be the zero polynomial. The same may be observed by expanding the terms of the equation. Finally, it holds for equation (1), that:

$$\begin{aligned}0 =& k_{2,*}\mathsf{ek}\left(s_1+s_3\gamma, s_1Z(\tau), s_1\tau+s_2\tau, s_2, s_4, s_5Z_0, \gamma, \left[\tau^i\right]_{i\in[0,n-t]}\right)^\intercal\\ &+k_{*,1}\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n]}\right)^\intercal\\ &+k_{*,2}Z(\tau)\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n]}\right)^\intercal\\ &+k_{*,3}\tau\left(\alpha, s_1\mathsf{ek}+s_4\tau^t+s_5, s_3\alpha, \left[\tau^i, H_i\right]_{i\in[0,n-1]}\right)^\intercal.\end{aligned}$$

In the following, let $g_1 = k_{2,*}\mathbf{l_2}^\intercal$, $g_2 = k_{*,1}\mathbf{l_2}^\intercal$, $g_3 = k_{*,2}\mathbf{l_1}^\intercal$, and $g_4(\tau) = k_{*,3}\mathbf{l_1}^\intercal$. Note that our constraints imply, as discussed above, that $\deg(g_1) \leq n - t$, $\deg(g_2) \leq 0$, $\deg(g_3) \leq n$, and $\deg(g_4) \leq n - 1$. Consider that by definition

$$\begin{aligned}\mathsf{ek} = &\sum_{i\in[n]\setminus(T\cup\mathcal{R})} \alpha\mathsf{sk}_i L_i(\tau) + \sum_{i\in\mathcal{R}\setminus\mathcal{T}} \bar{\alpha}_i\mathsf{sk}_i L_i(\tau)\\ &+ \sum_{i\in\mathcal{T}\cup\{0\}\setminus\mathcal{R}} \alpha\kappa_i L_i(\tau) + \sum_{i\in\mathcal{T}\cap\mathcal{R}} \bar{\alpha}_i\kappa_i L_i(\tau),\end{aligned}$$

where sk_i, α, and $\bar{\alpha}_j$ for $j \in [n]$ are variables in $\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T}$ and f and κ_i for $i \in \mathcal{T}$ are keys known to the adversary (constants in $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T})$. Now, consider the Lagrange basis form of polynomial $g_1(y) = \sum_{i\in[n]} b_i L_i(\tau)$, where b_i denotes the function value of g_1 at positions ω^i. The univariate sumcheck in the context of equation (1) equates to:

$$\begin{aligned}\mathsf{ek}\cdot g_1 \overset{2.1}{=} &\sum_{i\in[n]\setminus(T\cup\mathcal{R})} \alpha\mathsf{sk}_i b_i + \sum_{i\in\mathcal{R}\setminus\mathcal{T}} \bar{\alpha}_i\mathsf{sk}_i b_i + \sum_{i\in\mathcal{T}\cup\{0\}\setminus\mathcal{R}} \alpha\kappa_i b_i + \sum_{i\in\mathcal{T}\cap\mathcal{R}} \bar{\alpha}_i\kappa_i b_i\\ &+Q_x(\tau)\cdot\tau + Q_Z(\tau)\cdot Z(\tau).\end{aligned}$$

By equation (1), it also holds that:

$$\overset{(1)}{=} -(g_2 + g_3 Z(\tau) + g_4\tau)$$

Recall that all unique combinations of variables must have a coefficient of 0 and $Z(\tau) = \tau^{n+1} - 1$. As g_4 is of degree at most $n - 1$ and $Q_x(\tau)$ is of the same degree by definition, the only polynomials of degree τ^{n+1} or larger are $Q_Z(\tau)Z(\tau)$ and $g_3 Z(\tau)$. Further, note that g_3 must be a polynomial computed using the CRS. Assume that polynomial $(g_3 - Q_Z(\tau))(\tau^{n+1} - 1)$ contains a term involving sk_i for some $i \in [n]$, and a factor of τ^j where $j \leq n$. Then the equation must include a term involving the same sk_i with a factor of τ^{j+n+1} and inverted coefficient. As $(g_3 - Q_Z(\tau))(\tau^{n+1} - 1)$ is the only polynomial with τ coefficients of degree larger than n this leads to a contradiction. Thereby, $(g_3 - Q_Z(\tau)(\tau^{n+1} - 1)$ cannot contain terms involving sk_i.

Similarly, $(g_4 - Q_X(\tau))\tau$ cannot contain terms involving sk_i that are constant in τ, as $g_4 - Q_X(\tau)$ must be also be a valid polynomial.

As a result, terms $\alpha\mathsf{sk}_i$, $\bar{\alpha}_i\mathsf{sk}_i$, and $\bar{\alpha}_i\kappa_i$ can appear only in the first two as well as the fourth term originating from the sumcheck. It follows that $b_i = 0$ for each $i \in [n] \setminus (\mathcal{T} \setminus \mathcal{R})$. Given its Lagrange basis form, g_1 must have at least $n - |\mathcal{T} \setminus \mathcal{R}|$ roots. As g_1 cannot be the zero polynomial, this implies that $\deg(g_1) \geq n - |\mathcal{T} \setminus \mathcal{R}|$. Since $|\mathcal{T} \setminus \mathcal{R}| < t$ by virtue of the message privacy game, it holds that $\deg(g_1) > n - t$. However, by the constraints imposed by the equation system, it holds that $n - t \geq \deg(g_1)$. We arrive at a contradiction. Therefore, $(f, \mathbf{l_1}, \mathbf{l_2}, \mathbf{l_T})$ must be linearly independent. ■

I Message Privacy and Security for Π_{OSTEP}

Lemma I.1 *Consider an arbitrary PPT adversary $\mathcal{A}$ and let $n = \mathsf{poly}(\lambda)$. If $\mathsf{PS}_{\mathsf{Rand}}$ is zero-knowledge, Π_{OSTEP} as presented in Section 6 fulfills message privacy.*

Observe that the differences between Π_{OSTE} and Π_{OSTEP} are the added preprocessing and the restriction of the keyspace to the pre-generated keys. However, the challenger is responsible for preprocessing in GameMP. Message privacy can, therefore, be proven in exactly the same fashion as for Π_{OSTE}. The lemma follows immediately from Lemma 5.2.

Lemma I.2 *Consider an arbitrary PPT adversary $\mathcal{A}$ and let $n = \mathsf{poly}(\lambda)$. If $\mathsf{PS}_{\mathsf{Rand}}$ is sound and zero-knowledge, Π_{OSTEP} as presented in Section 6 is IND-CPA secure.*

In contrast to Π_{OSTE}, IND-CPA security of Π_{OSTEP} does not follow immediately from its message privacy. In IND-CPA, the adversary is trusted with computing the preprocessing output. For Π_{OSTEP}, this implies that the rerandomization values α and $\bar{\alpha}_j$ are known to the adversary. In the following, we present a proof sketch for IND-CPA security of construction Π_{OSTEP}.

IND-CPA follows similarly to the message privacy proof by a reduction to the master theorem. Since the values of α and $\bar{\alpha}_j$ are now known to the adversary, the polynomial lists of the master theorem simplify to

$$\begin{aligned}
\mathbf{l_1} &= \left(s_1\mathsf{ek} + s_4\tau^t + s_5, s_3, \left[\tau^i, \left[\mathsf{sk}_j\tau^i\right]_{j\in[n]}\right]_{i\in[0,n]}\right)\\
\mathbf{l_2} &= \left(s_1 + s_3\gamma, s_1Z(\tau), s_1\tau + s_2\tau, s_2, s_4, s_5Z_0, \gamma, [\tau^i]_{i\in[0,n]}\right)\\
\mathbf{l_T} &= (1)\\
f &= s_5.
\end{aligned}$$

Similarly to the message privacy proof, we present two game hops before reducing the final game hop to the master theorem.

$\underline{\mathsf{Game}_0}$: We define $\mathsf{Game}_0 \leftarrow \mathsf{IND\text{-}CPA}$.

$\underline{\mathsf{Game}_1}$: Game_1 is equivalent to Game_0, but the adversary merely chooses α, $\{\bar{\alpha}_j\}_{j\in\mathcal{R}}$ and sends them to the challenger who instead computes the preprocessing output.

$\underline{\mathsf{Game}_2}$: Game_2 is equivalent to Game_1, but we replace the proofs π attached to the hints by simulated proofs as $\mathsf{PS}_{\mathsf{KGen}}.\mathsf{Sim}(\mathsf{td}, \mathsf{pk}_i, \mathsf{hint}_i)$ where Sim is the simulator of the proof system $\mathsf{PS}_{\mathsf{KGen}}$, executed on a trapdoor td.

We then show that

$$\mathbf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(1^\lambda, n) \leq \mathbf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda, n) + \mathsf{negl}(\lambda)$$

by a reduction to the soundness of proof system $\mathsf{PS}_{\mathsf{Rand}}$ (soundness is implied by simulation-extractability). As long as $\mathsf{PS}_{\mathsf{Rand}}$ is sound, an adversary is forced to compute the preprocessing honestly, only learning $\alpha, \{\bar{\alpha}_j\}_{j\in\mathcal{R}}$ in the process.

Further, we prove that

$$\mathbf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(1^\lambda, n) \leq \mathbf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda, n) + \mathsf{negl}(\lambda)$$

by a reduction to the zero-knowledge property of $\mathsf{PS}_{\mathsf{Rand}}$. This reduction is equivalent to the proof of Lemma H.2.

Finally, we prove that

$$\mathbf{Adv}^{\mathsf{Game}_2}_{\mathcal{A}}(1^\lambda, n) \leq \mathsf{negl}(\lambda)$$

by a reduction to the master theorem. The reduction is built in exactly the same way as for the proof of Lemma H.3, but using the modified polynomial lists above. Since the reduction has knowledge of α and $\{\bar{\alpha}_j\}_{j\in\mathcal{R}}$, it can simulate Game_2 to $\mathcal{A}$ in the same way as for message privacy. Using the master theorem requires the polynomial lists $(\mathbf{l_1}, \mathbf{l_2}, \mathbf{l_t}, f)$ to be independent. As the polynomial lists used in this proof consider the rerandomization to be known, they are the same used in the security proof of the original STE construction of [28]. The independence of these lists of polynomials has already been proven in [28], concluding our proof.