TSS-PV: Traceable Secret Sharing With Public Verifiability

Duc Anh Luong[1], Jong Hwan Park[2], Changmin Lee[1], and Hyoseung Kim[3]

[1] Korea University
[2] Sangmyung University
[3] Hallym University

Abstract. High-value custodial systems require both *public verifiability* to audit key distribution and *traceability* to identify insider leakage via black-box *reconstruction boxes*. Existing schemes achieve one property but not both, leaving practical systems exposed to either undetectable dealer misbehavior or untraceable share leakage. Combining these properties introduces the Provenance Paradox: public verifiability equips the reconstruction box with verification predicates and a committed transcript, yet known tracing mechanisms rely on inputs that lack such provenance, letting the box detect and reject tracing queries.

We present TSS-PV, the first publicly verifiable traceable secret sharing scheme that resolves this paradox. Our key insight is to inject indistinguishable *dummy shares* (which serve as future tracing inputs) during the sharing phase itself, ensuring they are committed to the public transcript before any reconstruction box is constructed. We formalize syntax and security under a modular adversarial model: public verifiability holds against fully malicious dealers and parties; traceability identifies leaking parties given honest distribution; and non-imputability prevents a malicious dealer from framing honest parties. Both tracing properties assume a verifiability-aware perfect reconstruction box.

We instantiate TSS-PV over cyclic groups using Schnorr-based NIZKs, including a new Σ-protocol for power-chain relations, and adopt the generic tracing framework of Boneh *et al.* (CRYPTO'24). Public verification scales linearly and tracing scales quadratically in the number of parties, an acceptable trade-off given that tracing is invoked only upon confirmed leakage. A Curve25519 prototype on commodity hardware demonstrates practicality: across 32–256 parties, distribution verification completes in $\approx$14–107 ms, tracing in $\approx$0.23–78 s, and trace verification in $\approx$0.13–26 s.

1 Introduction

The security of high-value digital assets, from blockchain bridges to root certificate authorities, rests on (t, n)-threshold schemes such as Shamir [36] or Blakley [7] secret sharing, which distribute trust across n custodians so that any t can reconstruct the key and any fewer than t learn nothing. In adversarial deployments, two practical threats persist. (i) *Dishonest Behavior:* a malicious

dealer or custodian submits malformed shares, causing reconstruction to fail. (ii) *Share Leakage:* a sub-threshold coalition ($f < t$) embeds valid shares into a *reconstruction box* [10], a black-box device that outputs the secret when fed $t-f$ additional valid shares, hiding the leakers' identities.

The Verification-Traceability Gap. None of existing work address both threats simultaneously. Verifiable Secret Sharing (VSS) [20,33] and its publicly verifiable extension (PVSS) [38,25,13] mitigate threat (i) by binding shares to dealer commitments via public transcripts, enabling verification of share distribution and submission.

Orthogonally, Traceable Secret Sharing (TSS) [29,10] addresses threat (ii) by enabling traceability, the ability to identify leakers by querying a reconstruction box built from any coalition of size $0 < f < t$, and *non-imputability*, the ability to protect honest parties from being falsely accused. Boneh et al. [10] refined TSS along two lines: scheme-specific constructions for Shamir and Blakley secret sharing (SSS/BSS) and a Generic TSS (GTSS) applicable to any symmetric sharing scheme.

Boneh et al. [10] define an ϵ-good reconstruction box R as one that returns the correct secret with probability at least ϵ over randomly sampled valid query sets, while a *perfect* R always succeeds on any $t-f$ valid shares disjoint from those embedded in it. SSS/BSS-specific tracing operates on ϵ-good boxes, preserves tracer secrecy, and injects malformed dummy shares (shares that cannot recover the secret); GTSS requires a perfect box and a fully trusted tracer whose tracing key contains the secret, and injects freshly sampled valid shares as tracing inputs. Both tracing schemes assume the reconstruction box cannot distinguish tracing inputs from party shares, i.e., shares generated for actual parties during the sharing phase.

Technical Challenges. Realizing TSS-PV requires resolving the following challenges.

Challenge 1: The Provenance Paradox. How can a tracer inject dummy shares when every party share is publicly committed? In the publicly verifiable setting, colluders can build a *verifiability-aware* perfect box that has oracle access to the public transcript and all verification predicates, letting it reject any query without transcript provenance. This defeats both the SSS/BSS-specific tracers of [10] and the tracer-crafted inputs of [29,28,19], whose dummies fail public verification. Dummy shares must therefore be indistinguishable from real shares under public verification, without weakening verifiability or altering the reconstruction threshold. Only GTSS [10] uses valid shares as tracing inputs, but these are generated after the sharing phase and so leave no transcript footprint for a verifiability-aware box to accept. Committing GTSS-style dummy shares during the sharing phase gives them transcript provenance, letting them pass the verification predicates of the reconstruction box. Adopting GTSS as the tracing foundation, however, introduces a new risk.

Challenge 2: The Privileged Tracer Risk. Because GTSS uses party shares as tracing inputs, a malicious tracer with access to all shares could recover the secret or fabricate evidence to frame honest parties. To mitigate this risk, TSS-

PV separates into two phases: a Standard Lifecycle (sharing and reconstruction), in which parties retain their shares to preserve secrecy, and an Accountability Mechanism (tracing), in which parties disclose their shares, trading secrecy for accountability. Crucially, parties disclose shares only after collusion evidence has been irrevocably committed; any evidence submitted after disclosure is rejected, ensuring that adversaries cannot exploit disclosed shares to frame honest parties. This two-phase separation, however, raises the question of *when* the system should transition to the tracing phase.

Challenge 3: The Tracing Trigger Problem. Without a verifiable precondition, an adversary can fabricate a leakage claim to force the system into the tracing phase and compel honest parties to disclose their shares, even when no collusion has actually occurred. A robust system must therefore initiate tracing only upon verified evidence of a functional reconstruction box, which we formalize as a testable *tracing trigger event*.

Contributions. Our contributions are threefold:

- *Formal TSS-PV Definition.* We propose the first formal syntax and security model for TSS-PV, unifying public verifiability, traceability, and non-imputability against reconstruction boxes in a single primitive. The model introduces two new definitional ingredients designed for the publicly verifiable setting: a verifiability-aware perfect reconstruction box, which extends Boneh et al.'s adversary [10] with oracle access to the public transcript and every verification predicate, and a testable *tracing trigger event*, which initiates tracing only upon verifiable evidence of a functional reconstruction box.
- *Specific Construction:* We propose an efficient (t, n) construction with linear public verification and quadratic dealer sharing. Adopting the GTSS mechanism, we sample $t-1$ dummy shares during the Sharing phase via an Anonymous Authenticated Channel (AAC) [4,22], ensuring they are embedded into the public transcript and indistinguishable from real shares even to a box with a full network view. For efficient public verifiability, we propose a new Σ-protocol for power-chain relations that enables share distribution without encryption or heavyweight zero-knowledge proofs.
- *Proof-of-Concept:* We implement a Curve25519 prototype (AVX2) and benchmark the end-to-end pipeline. In the standard lifecycle, public verification scales linearly in n and completes in $\approx$ 14–107 ms for 32–256 parties; share-submission check scales linearly in t. Tracing is a one-time forensic procedure invoked only when there is valid evidence of leakage; its dominant cost is reconstruction-box query (quadratic in t), completing in $\approx$ 0.2–78 s on commodity hardware.

We clarify our adversarial model as follows.

Standard Lifecycle (Sharing and Reconstruction). Both dealer and parties are modeled as fully malicious; our public-verifiability predicates detect any malformed distribution or submission on the public transcript, matching the adversarial strength of PVSS [38,25,13].

Accountability Mechanism (Tracing). R is modeled as a verifiability-aware perfect reconstruction box, i.e., a deterministic oracle that behaves correctly on all valid

inputs. Under this model, non-imputability holds against a malicious dealer while traceability assumes honest share distribution, consistent with all prior TSS work [10,29].

We also extend TSS-PV to pointwise ϵ-good boxes (success probability $\geq \epsilon$ on every valid input) at the cost of relaxing non-imputability to honest dealing.

Paper Structure. Section 2 surveys additional related work. Section 3 introduces notation and complexity assumptions. Section 4 defines our TSS-PV syntax. Section 5 defines security notions for TSS-PV. Section 6 presents the proof systems, system model, construction, and security analysis. Section 7 reports experimental results. Section 8 discusses limitations and future directions. Appendices A contains pointwise ϵ-good box extension.

2 Additional Related Work

Table 1 compares TSS-PV with closely related schemes; we discuss additional related work below.

Traceable Verifiable Secret Sharing. The recent TVSS [2] targets both verifiability and traceability but provides only participant-level verification (VSS) and addresses protocol-level misbehavior (compliant disclosures) rather than reconstruction-box leakage, leaving black-box threats unaddressed.

General Access Structure Traceable Secret Sharing. Farràs and Guiot [19] extended traceable secret sharing to general monotone access structures, though their tracing guarantee assumes the access structure is hidden from the adversary. Goyal et al. [28] revisited TSS foundations and identified shortcomings in prior formulations regarding reconstruction box capabilities. While they achieve stronger traceability for general access structures, their tracer synthesizes fresh inputs during the tracing phase. In contrast, we address tracing in the presence of a static public transcript, where such tracer-generated inputs lack provenance and can be rejected by the box.

Anonymous Secret Sharing. Anonymous secret sharing [39,8,31] requires that shares reveal no information about the identity of the parties holding them; Bishop et al. [6] recently introduced fully anonymous secret sharing (FASS), additionally requiring that unauthorized shares hide all identifiable information including party identities. TSS-PV's dummy shares must satisfy a similar indistinguishability, but over the full public transcript rather than share values alone; we therefore employ the AAC for protocol-level indistinguishability.

Post-Quantum PVSS. While early works established the foundations of verifiable secret sharing [20,33] and its publicly verifiable variant [38,13], recent research focuses on scalability and post-quantum security. Gentry et al. [25] and Minh et al. [32] propose constructions based on Learning With Errors (LWE) to achieve post-quantum security in the standard model. Unlike these works, which target the sharing phase or adopt post-quantum assumptions, our work integrates accountability into the PVSS framework.

Table 1: Scheme comparison. **PV**: public verifiability; **Box**: supports tracing under reconstruction-box; **Model**: box behavior (ϵ: ϵ-good, P: perfect); **NI**: non-imputability; **Assump.**: main cryptographic assumption;

Scheme	**PV**	**Box**	**Model**	**NI**	**Assump.**
GSS'21 [29]	✗	✓	ϵ	✓	OWF
BPR'24 [10]	✗	✓	ϵ / P[a]	✓	OWF
GJP'25 [28]	✗	✓	ϵ	✓	iO/OWF
BEMS'25 [2]	✗[b]	✗[c]	—	✓	DDH/DLOG
GHL'22 [25]	✓	✗	—	—	LWE
TSS-PV	✓	✓	ϵ / P[d]	✓	DDH/ℓ-DLOG

[a]SSS/BSS-specific tracing achieves ϵ-good; GTSS requires perfect box and trusted tracer.
[b]Participant-verifiable (VSS), not publicly verifiable.
[c]Traces protocol-level misbehavior (compliant disclosures), not reconstruction-box leakage.
[d]Main construction assumes perfect box with malicious-dealer NI; Appendix A extends to pointwise ϵ-good boxes via $\mathcal{O}(\lambda/\epsilon)$ query amplification, relaxing NI to honest dealing.

Other Tracing Primitives. Traitor tracing [14,24,27,16,40,18,26] identifies receivers whose decryption keys were used to produce a pirate decoder. While structurally related, TSS-PV differs fundamentally: the secret is shared rather than individually encrypted, and tracing targets the leaking coalition rather than individual key holders.

Boneh *et al.* [11] introduces traceable Verifiable Random Functions (VRF), in which a sub-threshold coalition leaks an *evaluation box* that computes the VRF output rather than a reconstruction box that returns a shared secret, and a *private* tracer, armed with a Paillier-based trapdoor that forges evaluation proofs, traces the box even when it demands proofs of correct partial evaluation. This setting is orthogonal to ours: the verifiability their box enforces is the correctness of VRF *outputs*, not the public auditability of *share distribution*, so the Provenance Paradox does not arise in their model, and is thus neither solved by, nor portable to, their construction.

3 Preliminary

This section introduces notation, complexity assumptions, and the standard NIZK proof system underlying our construction.

3.1 Notation

For $n \in \mathbb{N}$, $[n]$ denotes $\{1, \ldots, n\}$; $\{x_i\}_{i\in[n]}$ abbreviates $\{x_1, \ldots, x_n\}$, and $\{x_i\}_{i\in T}$ the subset indexed by $T \subseteq [n]$. We write $\boldsymbol{x} = (x_1, \ldots, x_n)$ for ordered tuples and $\boldsymbol{x} \in \mathcal{S}^n$ to indicate $\boldsymbol{x}$ has n components from a set $\mathcal{S}$. For operations: $x \leftarrow \mathsf{F}(y)$ denotes deterministic computation; $r \leftarrow\!\!\$\ \mathcal{S}$ denotes uniform sampling from $\mathcal{S}$;

$A \coloneqq B$ defines A to be B. $A^{\mathcal{O}}$ indicates algorithm A has oracle access to $\mathcal{O}$. $\mathsf{F}(\mathcal{P}(x) \leftrightarrow \mathcal{D}(y)) \to o$ describes an interactive protocol F between party $\mathcal{P}$ (input x) and dealer $\mathcal{D}$ (input y), producing output o.

3.2 Generic Traceable Secret Sharing

We adopt the GTSS algorithm of Boneh et al. [10] as a building block.

Let $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$ be the shares of a secret s, and let R be a perfect reconstruction box embedded with the leaked set $E = \{\mathsf{sh}_i : i \in \mathcal{C}\}$ for a coalition $\mathcal{C}$ of size $f < t$. To test party i, the tracer samples $t-f-1$ fresh dummy shares $\mathsf{dsh}_1, \ldots, \mathsf{dsh}_{t-f-1}$ of s and queries R on $Q_i = \{\mathsf{dsh}_1, \ldots, \mathsf{dsh}_{t-f-1}, \mathsf{sh}_i\}$, supplying exactly the $t-f$ additional shares the box requires. By the threshold property, R combines its f embedded shares with Q_i and recovers s if and only if the union $E \cup Q_i$ contains t distinct shares, i.e., if and only if $\mathsf{sh}_i \notin E$ (the dummies being fresh). Hence $\mathsf{R}(Q_i) = \perp$ implies $\mathsf{sh}_i \in E$, exposing party i as a leaker; otherwise party i is exonerated.

3.3 Complexity Assumptions

Our construction works under the ℓ-DLOG and DDH assumptions.

Definition 1 (ℓ-DLOG [23]). *Given an integer ℓ, a group $\mathbb{G}$ of prime order q, a generator g, and a set $\{g^x, \ldots, g^{x^\ell}\}$, we say that the ℓ-DLOG assumption holds in $\mathbb{G}$ if the probability that an adversary $\mathcal{A}$ outputs $x \in \mathbb{Z}_q$ is negligible. More formally:*

$$\Pr[x' \leftarrow \mathcal{A}(g, g^x, \ldots, g^{x^\ell}) : x' = x] = \mathsf{negl}(\lambda),$$

where $\mathsf{negl}(\cdot)$ is a negligible function and $\lambda \in \mathbb{N}$ is the security parameter.

Definition 2 (DDH [9]). *Given a group $\mathbb{G}$ of prime order q, a generator g, and three elements $g^a, g^b, T \in \mathbb{G}$, we say that the DDH assumption holds in $\mathbb{G}$ if it is infeasible for an adversary $\mathcal{A}$ to distinguish between $T = g^{ab}$ and $T = g^z$, where $z \leftarrow\!\!\$\, \mathbb{Z}_q$. More formally, let $\mathsf{negl}(\cdot)$ be a negligible function,*

$$\left|\Pr[\mathcal{A}(g,g^a,g^b,g^{ab}) = 1] - \Pr[\mathcal{A}(g,g^a,g^b,g^z) = 1]\right| = \mathsf{negl}(\lambda),$$

where $\lambda \in \mathbb{N}$ is the security parameter.

3.4 NIZK Proof

Our construction requires Schnorr-type Σ-protocols for proving linear relations among discrete logarithms DL-REP [12], rendered non-interactive via the Fiat–Shamir transform [21] in the random-oracle model (ROM).

Specifically, given a public statement $\boldsymbol{Y} = (Y_1, \ldots, Y_u) \in \mathbb{G}^u$ and public bases $G = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_u)$ where each $\boldsymbol{g}_i = (g_{i,1}, \ldots, g_{i,l}) \in \mathbb{G}^l$, we prove that a witness $\boldsymbol{w} = (w_1, \ldots, w_l) \in \mathbb{Z}_q^l$ is valid for the relation:

$$\mathcal{R}^{\mathsf{LIN}} = \left\{ (\boldsymbol{Y}, G; \boldsymbol{w}) \;\middle|\; \forall i \in [u] : Y_i = \prod_{j=1}^{l} g_{i,j}^{w_j} \right\}$$

Let $H : \{0,1\}^* \to \mathbb{Z}_q$ be a random oracle. The protocol $\Pi^{\mathsf{NIZK}} = (\mathsf{NIZK.Prf}, \mathsf{NIZK.Vr})$ for $\mathcal{R}^{\mathsf{LIN}}$ is defined as:

– $\mathsf{NIZK.Prf}(\boldsymbol{Y}, G; \boldsymbol{w}) \to \pi$: Sample $\boldsymbol{r} \leftarrow\!\!\$\ \mathbb{Z}_q^l$ and compute:

$$\forall i \in [u] : A_i \leftarrow \prod_{j=1}^{l} g_{i,j}^{r_j}, \quad c \leftarrow H(\boldsymbol{Y}, G, \boldsymbol{A}),$$
$$\forall j \in [l] : z_j \leftarrow r_j + c \cdot w_j,$$

Output $\pi \coloneqq (c, \boldsymbol{z})$.

– $\mathsf{NIZK.Vr}(\boldsymbol{Y}, G, \pi) \to \{0,1\}$: Parse $\pi = (c, \boldsymbol{z})$, compute $A'_i \leftarrow Y_i^{-c} \cdot \prod_{j=1}^{l} g_{i,j}^{z_j}$ for all $i \in [u]$, and return 1 iff $c = H(\boldsymbol{Y}, G, \boldsymbol{A}')$.

Informally, a secure NIZK proof system satisfies *completeness* (any honest prover holding a valid witness can convince the verifier), *zero-knowledge* (there exists a simulator that, given only the statement, can generate proofs indistinguishable from real ones), and *soundness* (no PPT adversary can produce an accepting proof for a false statement except with negligible probability).

Theorem 3.1 (NIZK Security). *The DL-REP NIZK proof system is perfectly complete, zero-knowledge, and sound in the ROM [12,34].*

These NIZK proof systems can also be shown to be proofs of knowledge (PoK), i.e., they satisfy *knowledge-soundness* in the ROM: any PPT prover that produces an accepting proof admits an efficient extractor that outputs a corresponding witness, except with negligible probability.

4 Syntax

This section defines the syntax of TSS-PV together with the reconstruction box R used in tracing. The core syntax combines elements of TSS [10] and PVSS [38,25].

TSS-PV Syntax. Given public parameters $\mathsf{pp} \coloneqq (1^\lambda, n, t, \rho)$, where $\rho \in \{0,1\}^\lambda$ denotes a correlation string (e.g., ρ specifies the basis elements for our exponent-based commitments), a t-out-of-n Traceable Secret Sharing with Public Verifiability (TSS-PV) scheme consists of seven PPT algorithms:

$$\mathsf{TSS\text{-}PV} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{VerSD}, \mathsf{VerSS}, \mathsf{VerRS}, \mathsf{Trace}^{\mathsf{R}}, \mathsf{TrVer}^{\mathsf{R}}).$$

As in Boneh et al. [10], we adopt a symmetric formulation in which the dealer interacts independently with each index $i \in [n]$ by running Share, rather than sampling all n shares in a single vector. All algorithms run relative to a fixed public environment determined by $(1^\lambda, n, t, \rho)$; construction-specific public elements (e.g., an algebraic domain and correlation material derived from ρ) are determined at setup and kept implicit below to reduce clutter. We define the algorithms below and omit pp from inputs when clear.

- $\mathsf{Share}(\mathcal{P}_i(\mathsf{ps}_i) \leftrightarrow \mathcal{D}(\mathsf{ds})) \to (\mathsf{op}_i, \mathsf{od}_i, T_i)$: an interactive protocol between the dealer $\mathcal{D}$, holding dealer state ds (which includes the secret S and dealer randomness ω), and party $\mathcal{P}_i$, holding party state ps_i. It produces:
  - Party output $\mathsf{op}_i = (\mathsf{sh}_i, \mathsf{ptk}_i, \mathsf{pvk}_i)$, where sh_i is the share, ptk_i the party tracing key, and pvk_i the party verification key.
  - Dealer output $\mathsf{od}_i = (\mathsf{dtk}_i, \mathsf{dvk}_i)$, where dtk_i is the dealer tracing key and dvk_i the dealer verification key.
  - T_i is the public transcript generated during sharing.

  We bundle the per-index keys as $\mathsf{tk}_i \coloneqq (\mathsf{ptk}_i, \mathsf{dtk}_i)$ and $\mathsf{vk}_i \coloneqq (\mathsf{pvk}_i, \mathsf{dvk}_i)$, and collect them into the global sets $\mathsf{TK} \coloneqq \{\mathsf{tk}_i\}_{i\in[n]}$ and $\mathsf{VK} \coloneqq \{\mathsf{vk}_i\}_{i\in[n]}$.
- $\mathsf{Rec}(\mathsf{SH}) \to S'/\perp$: on input a size-$t$ subset $\mathsf{SH} \subseteq \{\mathsf{sh}_i\}_{i\in[n]}$, outputs a candidate secret S'; otherwise returns $\perp$. Rec performs no validity check; share validity is handled by VerSS (below).
- $\mathsf{VerSD}(\mathsf{T}) \to \{0,1\}$: on input $\mathsf{T} = \{T_i\}_{i\in[n]}$, checks whether the dealer distributed shares correctly.
- $\mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T}) \to \{0,1\}$: on input sh_i and $\mathsf{T} = \{T_i\}_{i\in[n]}$, verifies that sh_i is the share committed at index i in T.
- $\mathsf{VerRS}(S', \mathsf{T}) \to \{0,1\}$: on input S' and $\mathsf{T} = \{T_i\}_{i\in[n]}$, verifies that S' is the secret shared in T.
- $\mathsf{Trace}^{\mathsf{R}}(\mathsf{TK}, \mathsf{T}) \to (\mathcal{C}', \pi_{\mathsf{trace}})$: a tracing algorithm with oracle access to the reconstruction box R. On input $\mathsf{TK} = \{\mathsf{tk}_i\}_{i\in[n]}$ and $\mathsf{T} = \{T_i\}_{i\in[n]}$, it outputs an accused index set $\mathcal{C}' \subseteq [n]$ and a tracing proof π_{trace}.
- $\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}}) \to \{0,1\}$: a public judging algorithm with black-box access to the same box R used by $\mathsf{Trace}^{\mathsf{R}}$. On input $\mathsf{VK} = \{\mathsf{vk}_i\}_{i\in[n]}$, $\mathsf{T} = \{T_i\}_{i\in[n]}$, and the pair $(\mathcal{C}', \pi_{\mathsf{trace}})$ produced by $\mathsf{Trace}^{\mathsf{R}}$, it outputs 1 iff the accusation is accepted.

Reconstruction Box Syntax. A reconstruction box R is produced by a coalition $\mathcal{C} \subset [n]$ with $|C| = f$ and $0 < f < t$. Internally, R holds the coalition's shares $\{\mathsf{sh}_i\}_{i\in\mathcal{C}}$ and may invoke Rec together with the public predicates; externally, it is accessed only through the following black-box query.

- $\mathsf{R}(\mathsf{SH}) \to S'/\perp$: on input a set of candidate shares SH, the box returns either a secret S' or $\perp$, any output not in the secret space.

5 Security Notion

Formalizing security for TSS-PV requires definitions that capture the interaction between verification predicates and the tracing mechanism, an interaction

absent from prior TSS and PVSS models. All definitions are stated for a fixed scheme $\mathsf{TSS\text{-}PV} = (\mathsf{Share}, \mathsf{Rec}, \mathsf{VerSD}, \mathsf{VerSS}, \mathsf{VerRS}, \mathsf{Trace}^{\mathsf{R}}, \mathsf{TrVer}^{\mathsf{R}})$ over party secret space $\mathcal{PS}_\lambda$, dealer secret space $\mathcal{S}_\lambda$, dealer randomness space Ω_λ, and share space $\mathcal{SH}_\lambda$.

We adopt the static corruption model: $\mathcal{A}$ declares all corrupted parties at the onset of each experiment and may deviate arbitrarily from the protocol. TSS-PV is a symmetric secret sharing scheme in which all shares are identically distributed, so observing protocol executions adaptively before selecting corrupted parties confers no additional advantage [10].

Our correctness, dealer secrecy, and the verifiability notions adapt the standard VSS/PVSS framework [20,38,10]. Party secrecy (Def. 8), verifiability-aware reconstuction box (Def. 10), the tracing trigger event $\mathcal{T}$ (Def. 11), and the non-imputability definition under a verifiability-aware adversary (Def. 13) are introduced in this work.

5.1 Correctness

Informally, correctness requires that if all parties are honest then all proofs and predicates pass and any subset of t shares recovers the secret.

Definition 3 (Correctness). *Let* $\big(\mathsf{pp}, \Sigma \coloneqq \{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{T} \coloneqq \{T_i\}_{i\in[n]}\big)$ *be the output of experiment* $\mathbf{ExpCor}^{n,t}_{\mathsf{TSS\text{-}PV}}(\lambda)$ *as per Fig. 1. Let* $\epsilon = \epsilon(\lambda) \in [0,1]$ *be a function of the security parameter* λ*. We say the scheme is* ϵ-correct *if for every* $n \in \mathbb{N}$*, every* $0 < t \le n$*, the following probability is at least* $1-\epsilon$*, where the probability is taken over the coins of* Share *and* $\mathbf{ExpCor}^{n,t}_{\mathsf{TSS\text{-}PV}}(\lambda)$ *internal randomness.*

$$\Pr\left[\begin{array}{l} \big(\mathsf{VerSD}(\mathsf{T})=1\big) \ \wedge \ \big(\forall \mathsf{sh}\in\Sigma : \mathsf{VerSS}(\mathsf{sh},\mathsf{T})=1\big) \\ \wedge\ \exists S'\in\mathcal{S}_\lambda \ \textit{s.t.}\ \forall \mathsf{SH}\subseteq\Sigma, |\mathsf{SH}|=t : \\ \qquad\qquad \mathsf{Rec}(\mathsf{SH})=S' \wedge \mathsf{VerRS}(S',\mathsf{T})=1 \\ \wedge\ \big(\forall \mathsf{SH}'\subseteq\Sigma, |\mathsf{SH}'|<t : \mathsf{Rec}(\mathsf{SH}')=\perp\big) \end{array}\right]$$

Experiment $\mathbf{ExpCor}^{n,t}_{\mathsf{TSS\text{-}PV}}(\lambda)$

---

1. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda, \mathsf{pp} \coloneqq (1^\lambda, n, t, \rho)$;
2. $\{\mathsf{ps}_i\}_{i\in[n]} \leftarrow\!\!\$\ \mathcal{PS}_\lambda$, $S \leftarrow\!\!\$\ \mathcal{S}_\lambda$, $\omega \leftarrow\!\!\$\ \Omega_\lambda$, $\mathsf{ds} \coloneqq (S,\omega)$;
3. $\forall i \in [n] : (\mathsf{op}_i, \mathsf{od}_i, T_i) \leftarrow \mathsf{Share}\,(\mathcal{P}_i(\mathsf{ps}_i) \leftrightarrow \mathcal{D}(\mathsf{ds}))$,
   where $\mathsf{op}_i = (\mathsf{sh}_i, \mathsf{ptk}_i, \mathsf{pvk}_i), \mathsf{od}_i = (\mathsf{dtk}_i, \mathsf{dvk}_i)$;
4. **return** $\big(\mathsf{pp}, \Sigma \coloneqq \{\mathsf{sh}_i\}_{i\in[n]}, \mathsf{T} \coloneqq \{T_i\}_{i\in[n]}\big)$.

Fig. 1: The correctness experiment for TSS-PV. Dealer and all parties are honest.

5.2 Verifiability

We define three verification properties.

Distribution verifiability (VerSD). Informally, VerSD defends against a *malicious dealer*: whenever $\mathsf{VerSD}(\mathsf{T}) = 1$, every pair of t-subsets of VerSS-valid shares yields the same secret under Rec, so a cheating dealer cannot produce a transcript that VerSD accepts yet admits two different secrets.

Definition 4 (VerSD). *Let* $\mathbf{ExpVSD}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment in Fig. 2. We say* $\mathsf{TSS\text{-}PV}$ *has* verifiable distribution *if for every PPT adversary* $\mathcal{A}$*, the following advantage is negligible for any security parameter* $\lambda \in \mathbb{N}$*.*

$$\mathsf{Adv}^{\mathsf{vsd}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \Pr\left[\mathbf{ExpVSD}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right],$$

where the probability is taken over the random coins of $\mathcal{A}$ *and* $\mathbf{ExpVSD}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *internal randomness.*

Experiment $\mathbf{ExpVSD}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

---

1. $(n, t, \mathsf{state}) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$;
2. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda,\ \mathsf{pp} := (1^\lambda, n, t, \rho)$;
3. $(\mathsf{T} := \{T_i\}_{i\in[n]}, \Sigma := \{\mathsf{sh}_i\}_{i\in[n]}, I_0, I_1) \leftarrow \mathcal{A}(\mathsf{state}, \mathsf{pp})$;
4. $\mathcal{V}(\mathsf{T}, \Sigma) := \{\mathsf{sh}_i \in \Sigma : \mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T}) = 1\}$;
5. **if** $\big(|\mathcal{V}(\mathsf{T}, \Sigma)| < t\big) \vee \big(|I_0| \neq t\big) \vee \big(|I_1| \neq t\big) \vee \big(I_0 = I_1\big)$
   $\vee\ \big(I_0 \not\subseteq \{i \in [n] : \mathsf{sh}_i \in \mathcal{V}(\mathsf{T}, \Sigma)\}\big)$
   $\vee\ \big(I_1 \not\subseteq \{i \in [n] : \mathsf{sh}_i \in \mathcal{V}(\mathsf{T}, \Sigma)\}\big)$
   then return 0;
6. $S_b \leftarrow \mathsf{Rec}(\mathsf{SH}_b)$, where $\mathsf{SH}_b := \{\mathsf{sh}_i : i \in I_b\}$ for $b \in \{0,1\}$;
7. **if** $(\mathsf{VerSD}(\mathsf{T}) = 1) \wedge (S_0 \neq S_1)$ **then return** 1;
8. **return** 0.

Fig. 2: Distribution verifiability experiment. The adversary simulates the dealer and all parties.

Share Submission Verifiability (VerSS). Informally, VerSS defends an honest dealer against *malicious parties* and provides two guarantees: *binding*, no adversary can produce two distinct shares at the same index that both pass $\mathsf{VerSS}(\cdot, \mathsf{T})$; and *consistency*, any t shares that pass $\mathsf{VerSS}(\cdot, \mathsf{T})$ reconstruct via Rec to the dealer's secret S.

Definition 5 (VerSS). *Let* $\mathbf{ExpVSS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment in Fig. 3. We say* $\mathsf{TSS\text{-}PV}$ *has* verifiable submission *if for every PPT adversary* $\mathcal{A}$*, the following advantage is negligible for any security parameter* $\lambda \in \mathbb{N}$*.*

$$\mathsf{Adv}^{\mathsf{vss}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \Pr\left[\mathbf{ExpVSS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right],$$

where the probability is taken over the random coins of $\mathcal{A}$, Share, *and* $\mathbf{ExpVSS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *internal randomness.*

Experiment $\mathbf{ExpVSS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

---

1. $(n, t, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$;
2. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda, \mathsf{pp} := (1^\lambda, n, t, \rho)$;
3. $\mathsf{state}_2 \leftarrow \mathcal{A}(\mathsf{state}_1, \mathsf{pp})$;
4. $S \leftarrow\!\!\$\ \mathcal{S}_\lambda,\ \omega \leftarrow\!\!\$\ \Omega_\lambda,\ \mathsf{ds} := (S, \omega)$;
5. $\forall i \in [n] : ((\cdot), (\mathsf{dtk}_i, \mathsf{dvk}_i), T_i) \leftarrow \mathsf{Share}\,(\mathcal{A} \leftrightarrow \mathcal{D}(\mathsf{ds}))$;
6. $\mathsf{T} := \{T_i\}_{i\in[n]}$;
7. $(\mathsf{SH}, i^*, \mathsf{sh}_{i^*}, \mathsf{sh}'_{i^*}) \leftarrow \mathcal{A}(\mathsf{state}_2, \mathsf{T})$, where $|\mathsf{SH}| = t, \mathsf{SH} \subseteq \mathcal{SH}_\lambda$;
8. $S' \leftarrow \mathsf{Rec}(\mathsf{SH})$;
9. $b_0 \leftarrow \big(\mathsf{VerSS}(\mathsf{sh}_{i^*}, \mathsf{T}) = \mathsf{VerSS}(\mathsf{sh}'_{i^*}, \mathsf{T}) = 1\big) \wedge \big(\mathsf{sh}_{i^*} \neq \mathsf{sh}'_{i^*}\big)$;
10. $b_1 \leftarrow \big(\forall \mathsf{sh} \in \mathsf{SH} : \mathsf{VerSS}(\mathsf{sh}, \mathsf{T}) = 1\big) \wedge \big(S' \neq S\big)$;
11. **if** $b_0 \vee b_1$ **then return** 1;
12. **return** 0;

Fig. 3: The experiment of share-submission verifiability. The adversary simulates malicious parties.

Reconstructed Secret Verifiability (VerRS). Informally, VerRS provides *public binding* for the reconstructed secret: for any transcript T, even one produced by a malicious dealer, at most one $S' \in \mathcal{S}_\lambda$ satisfies $\mathsf{VerRS}(S', \mathsf{T}) = 1$. An ill-formed T may admit no such S'; we treat this as denial-of-service rather than a security violation.

Definition 6 (VerRS). *Let* $\mathbf{ExpVRS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment in Fig. 4. We say* $\mathsf{TSS\text{-}PV}$ *has* verifiable secret reconstruction *if for every PPT adversary* $\mathcal{A}$*, the following advantage is negligible for any security parameter* $\lambda \in \mathbb{N}$*.*

$$\mathsf{Adv}^{\mathsf{vrec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \Pr\left[\mathbf{ExpVRS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right],$$

where the probability is taken over random coins of $\mathcal{A}$ *and internal randomness of* $\mathbf{ExpVRS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$*.*

All three predicates are standalone: each re-checks global consistency internally, so no predicate assumes prior execution of another. The algorithmic realization of this design appears in Remark 7.

5.3 Secrecy

Informally, secrecy has two notions. *Dealer secrecy* requires that an adversary controlling up to $t-1$ parties learns nothing about the dealer's secret. *Party*

Experiment $\mathbf{ExpVRS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

1. $(n, t, \mathsf{state}) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$;
2. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda,\ \mathsf{pp} := (1^\lambda, n, t, \rho)$;
3. $(\mathsf{T} := \{T_i\}_{i\in[n]}, S_0, S_1) \leftarrow \mathcal{A}(\mathsf{state}, \mathsf{pp})$, where $S_0, S_1 \in \mathcal{S}_\lambda$ and $S_0 \neq S_1$;
4. **if** $\big(\mathsf{VerRS}(S_0, \mathsf{T}) = 1\big) \wedge \big(\mathsf{VerRS}(S_1, \mathsf{T}) = 1\big)$ **then return** 1;
5. **return** 0;

Fig. 4: VerRS experiments. The adversary plays the role of the dealer and all parties.

secrecy requires that an honest party's share stays hidden from a malicious dealer and the other $n-1$ parties.

Definition 7 (Dealer Secrecy). *Let* $\mathbf{ExpDSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment defined in Fig. 5, we say* TSS-PV *satisfies dealer secrecy if for every PPT adversary* $\mathcal{A}$ *and every security parameter* $\lambda \in \mathbb{N}$*, it holds that the following advantage is negligible in* λ*.*

$$\mathsf{Adv}^{\mathsf{dsec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \left| \Pr\left[\mathbf{ExpDSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right] - \frac{1}{|\mathcal{S}_\lambda|} \right|,$$

where the probability is taken over random coins of $\mathcal{A}$ *and internal randomness of* $\mathbf{ExpDSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$*.*

Definition 8 (Party secrecy). *Let* $\mathbf{ExpPSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment defined in Fig. 6, we say* TSS-PV *satisfies party secrecy if for every PPT adversary* $\mathcal{A}$ *and every security parameter* $\lambda \in \mathbb{N}$*, the following advantage is negligible in* λ*.*

$$\mathsf{Adv}^{\mathsf{psec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \left| \Pr\left[\mathbf{ExpPSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right] - \frac{1}{|\mathcal{SH}_\lambda|} \right|,$$

where the probability is taken over random coins of $\mathcal{A}$ *and internal randomness of* $\mathbf{ExpPSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$*.*

Remark 1 (Remarks on Secrecy). We make the following remarks on the secrecy experiments.

- **The Advantage Term.** We subtract $1/|\mathcal{S}_\lambda|$ and $1/|\mathcal{SH}_\lambda|$ from the adversary advantages $\mathsf{Adv}^{\mathsf{dsec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ and $\mathsf{Adv}^{\mathsf{psec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$, respectively, to discount the trivial attack where $\mathcal{A}$ guesses the secret $S \in \mathcal{S}_\lambda$ and $\mathsf{sh} \in \mathcal{SH}_\lambda$ (which naturally succeeds with probability $1/|\mathcal{S}_\lambda|$ and $1/|\mathcal{SH}_\lambda|$, respectively).
- **Computational Secrecy.** Our scheme achieves computational dealer secrecy rather than perfect (information-theoretic) dealer secrecy. This is an inherent consequence of public reconstruction verifiability: the predicate VerRS binds

Experiment $\mathbf{ExpDSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

1. $(n, t, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$;
2. $\{\mathsf{ps}_i\}_{i=t}^{n} \leftarrow\!\!\$\ \mathcal{PS}_\lambda$;
3. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda$, $\mathsf{pp} \coloneqq (1^\lambda, n, t, \rho)$;
4. $\mathsf{state}_2 \leftarrow \mathcal{A}(\mathsf{state}_1, \mathsf{pp})$;
5. $S \leftarrow\!\!\$\ \mathcal{S}_\lambda$, $\omega \leftarrow\!\!\$\ \Omega_\lambda$, $\mathsf{ds} \coloneqq (S, \omega)$;
6. **for** $i = 1, \ldots, t-1$: $((\cdot), (\mathsf{dtk}_i, \mathsf{dvk}_i), T_i) \leftarrow \mathsf{Share}(\mathcal{A} \leftrightarrow \mathcal{D}(\mathsf{ds}))$;
7. **for** $i = t, \ldots, n$: $((\mathsf{sh}_i, \mathsf{ptk}_i, \mathsf{pvk}_i), (\mathsf{dtk}_i, \mathsf{dvk}_i), T_i) \leftarrow \mathsf{Share}(\mathcal{P}_i(\mathsf{ps}_i) \leftrightarrow \mathcal{D}(\mathsf{ds}))$;
8. $S^* \leftarrow \mathcal{A}(\mathsf{state}_2, \{T_i\}_{i\in[n]})$;
9. **if** $S = S^*$ **then return** 1;
10. **return** 0.

Fig. 5: The dealer secrecy experiment. The adversary $\mathcal{A}$ controls $t-1$ parties during the sharing phase.

Experiment $\mathbf{ExpPSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

1. $(n, t, i^*, \{\mathsf{ps}_j\}_{j\in[n]\setminus\{i^*\}}, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$ and $i^* \in [n]$;
2. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda$, $\mathsf{pp} \coloneqq (1^\lambda, n, t, \rho)$, $\mathsf{ps}_{i^*} \leftarrow\!\!\$\ \mathcal{PS}_\lambda$;
3. $\mathsf{state}_2 \leftarrow \mathcal{A}(\mathsf{state}_1, \mathsf{pp})$;
4. $\big((\mathsf{sh}_{i^*}, \mathsf{ptk}_{i^*}, \mathsf{pvk}_{i^*}), (\cdot), T_{i^*}\big) \leftarrow \mathsf{Share}\big(\mathcal{P}_{i^*}(\mathsf{ps}_{i^*}) \leftrightarrow \mathcal{A}\big)$;
5. $\hat{\mathsf{sh}}_{i^*} \leftarrow \mathcal{A}(\mathsf{state}_2, T_{i^*})$;
6. **if** $\hat{\mathsf{sh}}_{i^*} = \mathsf{sh}_{i^*}$ **then return** 1 **else return** 0;

Fig. 6: The party secrecy experiment for TSS-PV. The adversary $\mathcal{A}$ simulates the dealer and $n-1$ parties.

the secret S to the public transcript T via a commitment that is computationally hiding under DDH but not information-theoretically hiding. An unbounded adversary could enumerate all candidates $S' \in \mathcal{S}_\lambda$ and check $\mathsf{VerRS}(S', \mathsf{T})$ to recover S.

5.4 Traceability

Informally, traceability requires that if a coalition $\mathcal{C}$ of $f < t$ parties builds a reconstruction box R that correctly reconstructs the secret from additional valid shares, then the tracer can identify every member of $\mathcal{C}$ and produce a publicly verifiable proof, using only black-box access to R. Because this guarantee is well-defined only when the dealer distributes shares correctly, we first formalize correct distribution as a predicate.

Definition 9 (Distribution Predicate). *Let* $\big(\Sigma \coloneqq \{\mathsf{sh}_i\}_{i\in[n]},\ \mathsf{T} \coloneqq \{T_i\}_{i\in[n]}\big)$ *be the output of running* n *instances of* Share. *Let* $\mathcal{V}(\mathsf{T}, \Sigma) \coloneqq \{\mathsf{sh} \in \Sigma : \mathsf{VerSS}(\mathsf{sh}, \mathsf{T}) =$

1} *be the set of valid shares. We define a correct distribution* $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma)$ *as*

$$\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) := \begin{bmatrix} |\mathcal{V}(\mathsf{T}, \Sigma)| \geq t \\ \wedge \ \exists S' \in \mathcal{S}_\lambda \ s.t. \ \forall\, \mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma),\ |\mathsf{SH}| = t : \\ \qquad\qquad \mathsf{Rec}(\mathsf{SH}) = S' \wedge \ \mathsf{VerRS}(S', \mathsf{T}) = 1 \end{bmatrix}.$$

Remark 2 (Remark on Def. 9.). By the binding property of VerRS (Def. 6), the secret S' in Φ_{dist} is unique whenever $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$. Conversely, $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 0$ indicates at least one of the three conjuncts in the definition failed.

Then, under the correct distribution ($\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$), we define a perfect reconstruction box that is verifiability-aware.

Definition 10 (Verifiability-aware Perfect Reconstruction Box). *Fix* (T, Σ) *with* $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$ *and unique* S'*. An oracle* R *is a verifiability-aware perfect reconstruction box for a coalition* $\mathcal{C} \subseteq [n]$ *with leakage* $f = |\mathcal{C}|$ *if its internal state contains exactly the shares* $\mathsf{E} = \{\mathsf{sh}_i : i \in \mathcal{C}\} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$ *and the following holds for all* $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$*:*

- ***(Valid inputs)*** *If* $|\mathsf{SH}| = t - f$ *and* $\mathsf{SH} \cap \mathsf{E} = \emptyset$*, then*

$$\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1] = 1$$

  and moreover $\mathsf{R}(\mathsf{SH}) = S'$ *with probability* 1*.*
- ***(Invalid inputs)*** *Otherwise,*

$$\Pr[\mathsf{R}(\mathsf{SH}) = \perp \vee \mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 0] = 1.$$

The probability is over the internal randomness of R*.*

Remark 3 (Remark on Def. 10). In this model, the oracle R has access to the public transcript T and all verification predicates of TSS-PV (VerSD, VerSS, and VerRS); it may invoke these predicates internally to validate input shares before attempting reconstruction. Appendix A relaxes the valid-input clause to $\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1] \geq \epsilon$ for $\epsilon \in (0, 1]$, formalizing pointwise ϵ-good boxes.

Given a verifiability-aware perfect reconstruction box as per Def. 10, we define the tracing trigger event $\mathcal{T}$ as follows.

Definition 11 (Tracing Trigger Event). *Let* $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$ *as per Def. 9 and let* R *be a verifiability-aware perfect reconstruction box as in Def. 10. For a coalition* $\mathcal{C} \subseteq [n]$ *of size* f*, we say the tracing trigger event* $\mathcal{T}(\mathsf{T}, \Sigma, \mathsf{R}; \mathcal{C})$ *occurs if and only if there exists* $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$ *with* $|\mathsf{SH}| = t - f$ *and* $\mathsf{SH} \cap \{\mathsf{sh}_i : i \in \mathcal{C}\} = \emptyset$ *such that* $\mathsf{VerRS}\big(\mathsf{R}(\mathsf{SH}), \mathsf{T}\big) = 1$*.*

Remark 4 (Leakage Evidence). Def. 11 defines the trigger event $\mathcal{T}$ based on the existence of a verifiability-aware perfect reconstruction box. In the concrete protocol, the dealer verifies $\mathcal{T}$ by probing the box with dummy shares. The probe demonstrates functionality on the test set; our formal model (which assumes honest sharing) treats this as sufficient evidence of leakage.

We now define traceability as follows.

Definition 12 (Traceability). *Let* $\mathbf{ExpTrace}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment of Fig. 7. Let* Φ_{dist} *be the distribution predicate as per Def. 9,* R *be the verifiability-aware perfect reconstruction box as per Def. 10, and* $\mathcal{T}$ *be the tracing trigger event as per Def. 11. We say that* $\mathsf{TSS\text{-}PV}$ *satisfies* traceability *if the following advantage is negligible in* λ *for every PPT adversary* $\mathcal{A}$*.*

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \left| \Pr\left[\mathbf{ExpTrace}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right] - \frac{1}{|\mathcal{S}_\lambda|} \right|,$$

where the probability is over $\mathbf{ExpTrace}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *internal randomness,* Share, R*, and* $\mathcal{A}$*.*

Remark 5 (Remark on Def. 12). We clarify our definition of Traceability as follows.

- *Adversarial Model.* The traceability experiment (Def. 12) assumes honest share generation (thus, TK and VK), and honest trigger activation by the dealer. This scopes the definition to capturing *correctness* of tracing (leakers are identified) rather than *liveness* (tracing will occur). Our construction achieves additional verifiability properties beyond what the definition requires.
- *Adversarial Advantage.* As in [10], we discount the baseline $1/|\mathcal{S}_\lambda|$ to remove the trivial strategy where a box R ignores its inputs and returns a uniformly random secret; such a box passes VerRS with probability exactly $1/|\mathcal{S}_\lambda|$, independent of the shares.

Experiment $\mathbf{ExpTrace}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

---

1. $(n, t, f, \mathsf{state}) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < f < t \leq n$;
2. $\rho \leftarrow\!\$\ \{0,1\}^\lambda$, $\quad \mathsf{pp} := (1^\lambda, n, t, \rho)$, $\quad S \leftarrow\!\$\ \mathcal{S}_\lambda$, $\quad \omega \leftarrow\!\$\ \Omega_\lambda$,
   $\mathsf{ds} := (S, \omega)$, $\quad \{\mathsf{ps}_i\}_{i\in[n]} \leftarrow\!\$\ \mathcal{PS}_\lambda$;
3. $\forall i \in [n] : (\mathsf{op}_i, \mathsf{od}_i, T_i) \leftarrow \mathsf{Share}\,(\mathcal{P}_i(\mathsf{ps}_i) \leftrightarrow \mathcal{D}(\mathsf{ds}))$,
   where $\mathsf{op}_i = (\mathsf{sh}_i, \mathsf{ptk}_i, \mathsf{pvk}_i), \mathsf{od}_i = (\mathsf{dtk}_i, \mathsf{dvk}_i)$;
4. set $\mathsf{TK} := \{\mathsf{ptk}_i, \mathsf{dtk}_i\}_{i\in[n]}$, $\quad \mathsf{VK} := \{(\mathsf{pvk}_i, \mathsf{dvk}_i)\}_{i\in[n]}$,
   $\Sigma := \{\mathsf{sh}_i\}_{i\in[n]}$, and $\mathsf{T} := \{T_i\}_{i\in[n]}$;
5. $(\mathcal{C}, \mathsf{R}) \leftarrow \mathcal{A}(\mathsf{state}, \mathsf{pp}, \{\mathsf{op}_i\}_{i\in[n]}, \mathsf{T})$, where $|\mathcal{C}| = f$ and $\mathcal{C} \subset [n]$;
6. **if** $\neg\mathcal{T}(\mathsf{T}, \Sigma, \mathsf{R}; \mathcal{C})$ **then return** 0;
7. $(\mathcal{C}', \pi_{\mathsf{trace}}) \leftarrow \mathsf{Trace}^{\mathsf{R}}(\mathsf{TK}, \mathsf{T})$;
8. **if** $(\mathcal{C}' = \mathcal{C}) \wedge (\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}}) = 1)$ **then return** 0;
9. **return** 1;

Fig. 7: Traceability experiment. $\mathcal{A}$ simulates f leakers.

5.5 Non-imputability

Informally, non-imputability requires that no adversary, even one controlling the dealer and the other $n-1$ parties, can produce an accepted accusation $\mathcal{C}$ that includes an honest party $\mathcal{P}_{i^*}$.

Definition 13 (Non-imputability). *Let* $\mathbf{ExpNI}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *be the experiment in Fig. 8. Let* R *be the verifiability-aware perfect reconstruction box as per Def. 10, and* $\mathcal{T}$ *be the tracing trigger event as per Def. 11. We say* TSS-PV *satisfies* non-imputability *if for every PPT adversary* $\mathcal{A}$*, and every* $\lambda \in \mathbb{N}$*, the following advantage is negligible.*

$$\mathsf{Adv}^{\mathsf{ni}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) := \Pr\left[\mathbf{ExpNI}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right],$$

where the probability is over $\mathbf{ExpNI}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ *internal randomness,* Share*,* R*, and* $\mathcal{A}$*..*

Experiment $\mathbf{ExpNI}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$

---

1. $(n, t, \mathsf{state}_1) \leftarrow \mathcal{A}(1^\lambda)$, where $0 < t \leq n$;
2. $\rho \leftarrow\!\!\$\ \{0,1\}^\lambda, \mathsf{pp} := (1^\lambda, n, t, \rho)$;
3. $(i^*, \mathsf{state}_2) \leftarrow \mathcal{A}(\mathsf{state}_1, \mathsf{pp})$, where $i^* \in [n]$;
4. $\mathsf{ps}_{i^*} \leftarrow\!\!\$\ \mathcal{PS}_\lambda$, $\mathsf{L} := [n] \setminus \{i^*\}$;
5. $((\mathsf{sh}_{i^*}, \mathsf{ptk}_{i^*}, \mathsf{pvk}_{i^*}), (\cdot), T_{i^*}) \leftarrow \mathsf{Share}\left(\mathcal{P}_{i^*}(\mathsf{ps}_{i^*}) \leftrightarrow \mathcal{A}\right)$;
6. $(\mathcal{C}, \pi_{\mathsf{trace}}, \{(\mathsf{sh}_\ell, \mathsf{pvk}_\ell, T_\ell)\}_{\ell \in \mathsf{L}}, \{\mathsf{dvk}_\ell\}_{\ell \in [n]}, \mathsf{R}) \leftarrow \mathcal{A}\left(\mathsf{state}_2, T_{i^*}\right)$;
7. set $\mathsf{T} := \{T_\ell\}_{\ell \in [n]}, \mathsf{VK} := \{(\mathsf{pvk}_\ell, \mathsf{dvk}_\ell)\}_{\ell \in [n]}$, and $\Sigma := \{\mathsf{sh}_\ell\}_{\ell \in [n]}$;
8. **if** $\neg\mathcal{T}(\mathsf{T}, \Sigma, \mathsf{R}; \mathcal{C})$ **then return** 0;
9. **if** $\left(i^* \in \mathcal{C}\right) \wedge \left(\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}, \pi_{\mathsf{trace}}) = 1\right)$ **then return** 1;
10. **return** 0;

Fig. 8: The Non-imputability experiment. The adversary $\mathcal{A}$ simulates the dealer $\mathcal{D}$ and $n-1$ parties.

6 Construction

In this section, we propose our proof systems, the TSS-PV construction, and their seucurity analysis.

6.1 Proof Systems

We propose NIZK instances and the PC-PoK as the fundamental building blocks for our TSS-PV construction.

NIZK Instances We instantiate the NIZK from Section 3 for the following relations:

Per-party share proof π_i. Given $\boldsymbol{Y} = (\mathsf{p}_i, \mathsf{cr}, \mathsf{cs})$, bases $G = \big((\mathsf{f}_i, g), (h_1, 1_{\mathbb{G}}), (1_{\mathbb{G}}, h_2)\big)$, witness $\boldsymbol{w} = (r, s)$, the proving relation is $\mathsf{p}_i = \mathsf{f}_i^r g^s \ \wedge\ \mathsf{cr} = h_1^r \ \wedge\ \mathsf{cs} = h_2^s$.

Reconstruction consistency proof ψ. Given $\boldsymbol{Y} = (S', \mathsf{cs})$, bases $G = \big((g), (h_2)\big)$, witness $\boldsymbol{w} = (s)$, the proving relation is $S' = g^s \ \wedge\ \mathsf{cs} = h_2^s$. Note, the proof ψ is generated during the sharing phase (when $S' = S$ is known to the dealer) but becomes publicly verifiable only after S' is revealed (e.g., during reconstruction or tracing).

We denote by NIZK_π and NIZK_ψ the instantiations of Π^{NIZK} for the per-party proof π_i and the reconstruction proof ψ, respectively.

PC-PoK In our construction, each party must prove knowledge of their evaluation point x_i such that $\mathsf{f}_i = \prod_{j=1}^{t-1} \alpha_j^{x_i^j}$ for a public vector $\boldsymbol{\alpha} \in \mathbb{G}^{t-1}$. Standard DL-REP proofs [12] do not directly apply since the exponents $(x_i, x_i^2, \ldots, x_i^{t-1})$ are powers of a single witness rather than independent values. We construct a specialized Σ-protocol (so-called PC-PoK) for this power-chain relation.

Formally, given the public parameters $\mathsf{pp}^{\mathsf{PC}} = (g, \boldsymbol{\alpha} \in \mathbb{G}^{t-1})$, we prove knowledge of a witness $x \in \mathbb{Z}_q$ for the power-chain relation:

$$\mathcal{R}^{\mathsf{PC}} = \left\{ \big((\mathsf{f}, \boldsymbol{V}); x\big) \;\middle|\; \begin{array}{l} V_0 = g \ \wedge\ \forall k \in [t-1] : V_k = (V_{k-1})^x \\ \wedge\ \mathsf{f} = \displaystyle\prod_{k=1}^{t-1} \alpha_k^{x^k} \end{array} \right\}$$

We employ a strategy decomposing $\mathcal{R}^{\mathsf{PC}}$ into two parallel Σ-protocols: *Track A* (Structure) uses chained DLEQ proofs [15] to establish $V_{j+1} = V_j^x$, and *Track B* (Consistency) uses a Schnorr-type multi-exponentiation proof [34] to link f with $\boldsymbol{V}$.

Let $H : \{0,1\}^* \to \mathbb{Z}_q$ be a random oracle. The protocol $\Pi^{\mathsf{PC}} = (\mathsf{PC.Prf}, \mathsf{PC.Vr})$ is defined as:

– $\mathsf{PC.Prf}(\mathsf{pp}^{\mathsf{PC}}, x) \to (\mathsf{f}, \boldsymbol{V}, \pi^{\mathsf{PC}})$: The prover computes

$$\begin{aligned}
&V_0 = g, \quad \forall k \in [t{-}1] : V_k = (V_{k-1})^x, \\
&\boldsymbol{V} \coloneqq (V_1, \ldots, V_{t-1}), \quad \mathsf{f} \leftarrow \prod_{k=1}^{t-1} \alpha_k^{x^k}, \\
&\boldsymbol{r} = (r_1, \ldots, r_{t-1}) \leftarrow\!\$\ \mathbb{Z}_q^{t-1}, \quad \forall k \in [t{-}2] : A_k \leftarrow V_k^{r_1}, \\
&\forall k \in [t{-}1] : B_k \leftarrow g^{r_k}, \quad \widehat{B} \leftarrow \prod_{k=1}^{t-1} \alpha_k^{r_k}, \\
&c \leftarrow H(\mathsf{pp}^{\mathsf{PC}}, \mathsf{f}, \boldsymbol{V}, \{A_k\}_{k\in[t-2]}, \{B_k\}_{k\in[t-1]}, \widehat{B}), \\
&\forall k \in [t{-}1] : z_k \leftarrow r_k + c \cdot x^k, \quad \boldsymbol{z} \coloneqq (z_1, \ldots, z_{t-1}).
\end{aligned}$$

Output $(\mathsf{f}, \boldsymbol{V}, \pi^{\mathsf{PC}} := (c, \boldsymbol{z}))$.

– $\mathsf{PC.Vr}(\mathsf{pp}^{\mathsf{PC}}, \mathsf{f}, \boldsymbol{V}, \pi^{\mathsf{PC}}) \to \{0,1\}$: The verifier parses $\pi^{\mathsf{PC}} = (c, \boldsymbol{z})$ and computes

$$\forall k \in [t{-}2] : A'_k \leftarrow V_k^{z_1} V_{k+1}^{-c},$$
$$\forall k \in [t{-}1] : B'_k \leftarrow g^{z_k} V_k^{-c}, \quad \widehat{B}' \leftarrow \left(\prod_{k=1}^{t-1} \alpha_k^{z_k}\right) \cdot \mathsf{f}^{-c}.$$

The verifier accepts if

$$c{=}H(\mathsf{pp}^{\mathsf{PC}}, \mathsf{f}, \boldsymbol{V}, \{A'_k\}_{k\in[t-2]}, \{B'_k\}_{k\in[t-1]}, \widehat{B}').$$

Theorem 6.2 (Security of PC-PoK). *In the ROM and under the* 1*-DLOG in* $\mathbb{G}$*, the PC-PoK proof system is perfectly complete, zero-knowledge, and knowledge-sound.*

Proof. Completeness follows since, for all $k \in [t-2]$,

$$A'_k = V_k^{z_1} V_{k+1}^{-c} = V_k^{r_1 + cx} (V_k^x)^{-c} = V_k^{r_1} = A_k$$

and for all $k \in [t-1]$,

$$B'_k = g^{z_k} V_k^{-c} = g^{r_k + cx^k} (g^{x^k})^{-c} = g^{r_k} = B_k.$$

The proof for $\widehat{B}'$ is analogous.

For zero-knowledge, the simulator $\mathcal{S}$ samples random c and $\boldsymbol{z}$, computes commitments backwards via the verification equations, and programs

$$H(\mathsf{pp}^{\mathsf{PC}}, \mathsf{f}, \boldsymbol{V}, \{A_k\}, \{B_k\}, \widehat{B}) = c.$$

The transcript is identically distributed since $(c, \boldsymbol{z})$ are uniform in both cases.

For knowledge soundness, let x be the solution of 1-DLOG problem. By the Forking Lemma [5], two valid $(c, \boldsymbol{z})$ and $(c', \boldsymbol{z}')$, where $c \neq c'$ for the same commitments, can be obtained with non-negligible probability. This implies the existence of extracted values $x^k = (z_k - z'_k)/(c - c')$ for all $k \in [t-1]$. Hence, the solution x can be easily derived by $x = x^k \cdot (x^{k-1})^{-1}$.

6.2 System Model

Our model follows the standard threshold secret-sharing setting used in traceable/verifiable schemes, with three modifications to support accountability: (i) the sharing phase requires an Anonymous Authenticated Channel (AAC) to preserve index indistinguishability, (ii) the participant set includes $t-1$ dummy indices used for tracing, and (iii) tracing is executed by the dealer using dummy shares, after parties publicly release their verification keys upon trigger activation.

Reconstruction remains standard: any t valid shares recover the secret. The effective operational threshold is t out of $m = |\mathcal{I}|$ real parties, as dummy shares are held by the dealer and utilized only during the tracing phase.

Public Parameters. The construction is parameterized by $\mathsf{pp} = (1^\lambda, n, t, \rho)$ and domain $\mathcal{G}(\lambda) = (\mathbb{G}, q, g, h_1, h_2)$ (an implicit global parameter fixed for security parameter λ and shared across all algorithms). ρ is a public random string (e.g., from a random beacon) used to derive a basis vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{t-1}) \in \mathbb{G}^{t-1}$ (e.g., via a hash-to-group procedure). The discrete logarithm relations among generators and basis elements are assumed unknown. For PC-PoK, the public parameter $\mathsf{pp}^{\mathsf{PC}} := (g, \boldsymbol{\alpha})$ is derived directly from pp.

Entities. The indexed universe $\mathcal{U} = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ is partitioned into m real indices $\mathcal{I}$ and $t-1$ dummy indices $\mathcal{J}$. We enforce indistinguishability: given transcript T and pp, determining whether index $\ell \in [n]$ belongs to $\mathcal{I}$ or $\mathcal{J}$ is computationally hard. Dummy indices are logical participants whose network credentials are controlled by the dealer.

In addition to these participants, the system defines specific functional roles:

- *Combiner*: Aggregates shares to reconstruct the secret; this role can be assigned to any entity.
- *Trigger Authority*: Validates leakage evidence to activate tracing. In TSS-PV, this role is fixed as the Dealer.
- *Tracer*: Executes the tracing algorithm. This role is also fixed as the Dealer in TSS-PV.
- *Judge*: Verifies the tracing output (proof of guilt); can be any entity holding the verification key.
- *External Verifier*: Verifies the validity of the reconstructed secret without access to individual shares.

Communication Channels. The sharing phase requires an AAC [4,22] to preserve index indistinguishability: the AAC provides origin authenticity, sender anonymity, unlinkability, and uniform message volume across all n indices, preventing a network observer from distinguishing real parties $\mathcal{I}$ from dummy indices $\mathcal{J}$. Reconstruction and tracing operate over standard authenticated channels, as index identities are revealed once the tracing trigger $\mathcal{T}$ fires.

Protocol Lifecycle. TSS-PV operates in two modes, depicted in Fig. 9: a standard lifecycle for sharing and reconstruction, and an accountability mechanism activated upon confirmed leakage evidence.

- *Standard Lifecycle.* Includes the Sharing and Reconstruction phases.
  - *Sharing.* The dealer $\mathcal{D}$ runs Share interactively with each party $\mathcal{P}_i$ for $i \in [n]$ over the AAC. Any entity can verify share distribution using VerSD.
  - *Reconstruction.* All parties submit shares to a combiner. It checks validity using VerSS and reconstructs S. External verifiers (who do not have access to shares) may optionally verify the output using VerRS.
- *Tracing (Accountability Mechanism).* The tracing phase comprises two subphases:
  - *Trigger:* The dealer probes the reconstruction box R with subsets of dummy shares. If some subset of size $t - f$ consistently yields a valid secret (via VerRS), the dealer determines the leakage size f and declares the trigger

$\mathcal{T}$ active. Upon activation, all parties release tracing and verification keys. Once $\mathcal{T}$ fires, R is fixed; no new reconstruction box is accepted as leakage evidence.

- *Execution:* The tracer uses tracing keys to identify leakers via $\mathsf{Trace}^{\mathsf{R}}$ and the judge uses verification keys to confirm the tracing output via $\mathsf{TrVer}^{\mathsf{R}}$.

Design Choice Justification. We explicitly motivate three architectural decisions. Firstly, we assign custody of the $t-1$ dummy shares to the dealer. Since the dealer already knows s (and thus $S = g^s$), holding additional dummy shares yields no information gain. Conversely, delegating this to a TTP would reduce the effective secrecy threshold because a TTP holding $t-1$ dummy shares needs only one cooperating real party to recover the secret. Secondly, we assign trigger authority to the dealer, positioning it as the gatekeeper of the forensic process. This aligns incentives: tracing protects the dealer's secret, so the dealer is most motivated to initiate it upon leakage. Delegating to a TTP introduces principal-agent risk, where a curious or colluding TTP might suppress the trigger to exploit the reconstruction box.

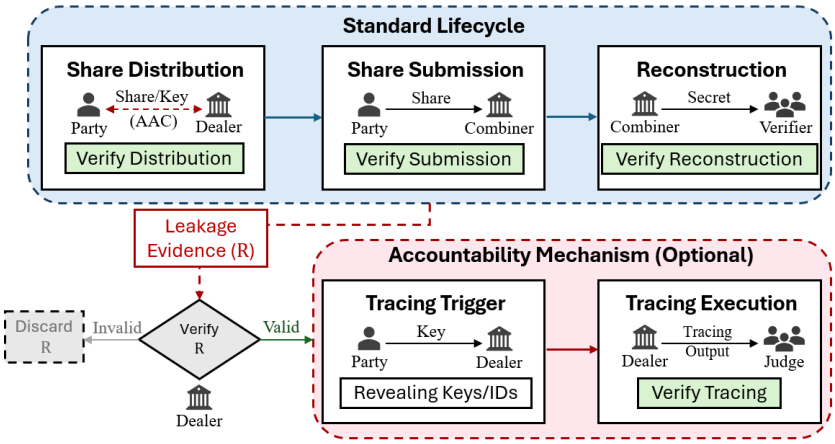


Fig. 9: Operational workflow of TSS-PV. Leakage evidence (the reconstruction box R) can surface during the standard lifecycle. R is verified; if valid, the dealer activates the accountability mechanism to identify leakers.

6.3 Construction

We introduce our construction in detail as follows.

Sharing Algorithm. The sharing algorithm (Figure 10) is an interactive protocol between the dealer $\mathcal{D}$ and each participant $\mathcal{P}_i$ for $i \in [n]$, where $n = m+(t-1)$ includes both real parties ($i \in \mathcal{I}$) and dummy indices ($i \in \mathcal{J}$). Note that ds is

pre-sampled by the dealer, ps_i is pre-sampled by each real party $\mathcal{P}_i$ for $i \in \mathcal{I}$, and by the dealer for dummy indices $i \in \mathcal{J}$. The core design principle is that each public share component $\mathsf{p}_i = \mathsf{f}_i^r \cdot S$ evaluates a hidden degree-$(t-1)$ polynomial at point x_i in the exponent, enabling threshold reconstruction via Lagrange interpolation on $\mathbb{G}$. The global commitment C_i and proofs (π_i, ψ_i) are generated identically across all n indices, binding the dealer to a single $\omega = (r, s)$ and making real and dummy indices publicly indistinguishable. Note, the PC-PoK proof π_i^{PC} allows the dealer to verify that $\mathcal{P}_i$ knows x_i consistent with $\mathsf{f_i}$, while π_i and ψ_i allow any external party to verify share computation and secret commitment, respectively.

At the end of the protocol, each real party $\mathcal{P}_i$ for $i \in \mathcal{I}$ outputs $\mathsf{ptk}_i = \mathsf{pvk}_i = \mathsf{sh}_i = (x_i, \mathsf{p}_i)$, while the dealer outputs $\mathsf{dtk}_i = \mathsf{dvk}_i = (x_i, \mathsf{p}_i)$ for each dummy index $i \in \mathcal{J}$.

$\underline{\mathsf{Share}(\mathcal{P}_i(\mathsf{ps}_i) \leftrightarrow \mathcal{D}(\mathsf{ds})) \to ((\mathsf{sh}_i, \mathsf{ptk}_i, \mathsf{pvk}_i), (\mathsf{dtk}_i, \mathsf{dvk}_i), T_i)\text{:}}$

//Run for each $i \in [n]$. $\mathcal{I}$ indexes real parties; $\mathcal{J}$ indexes dummies.

Party steps ($\mathcal{P}_i$): //for $i \in \mathcal{J}$: $\mathcal{D}$ executes these steps

1: **parse** $\rho = \boldsymbol{\alpha} \in \mathbb{G}^{t-1}$ **and** $\mathsf{ps}_i = (x_i)$, where $x_i \leftarrow\!\!\$\ \mathbb{Z}_q$;
2: $(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}) \leftarrow \mathsf{PC.Prf}(\mathsf{pp}^{\mathsf{PC}}, x_i)$; //note: $\mathsf{pp}^{\mathsf{PC}} := (g, \boldsymbol{\alpha})$
3: **send** $(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}})$ to $\mathcal{D}$;

Dealer steps ($\mathcal{D}$):

4: **parse** $\rho = \boldsymbol{\alpha} \in \mathbb{G}^{t-1}$, $\mathsf{ds} = (S, \omega)$, $\omega = (r, s)$, where $r, s \leftarrow\!\!\$\ \mathbb{Z}_q$ and $S = g^s$;
5: **if** $\mathsf{PC.Vr}(\mathsf{pp}^{\mathsf{PC}}, \mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}) = 0$ or $\mathsf{f}_i = \mathbf{1}_{\mathbb{G}}$ **then abort**;
6: $\mathsf{p}_i \leftarrow \mathsf{f}_i^r \cdot S$; //note: $\mathsf{f}_i^r \cdot S = \prod_{k=1}^{t-1} \alpha_k^{x_i^k r} \cdot S$
7: $\mathsf{C}_i := (\mathsf{cr}_i, \mathsf{cs}_i)$ **where** $\mathsf{cr}_i \leftarrow h_1^r$, $\mathsf{cs}_i \leftarrow h_2^s$; //same for all i
8: $\pi_i \leftarrow \mathsf{NIZK.Prf}((\mathsf{p}_i, \mathsf{cr}_i, \mathsf{cs}_i), ((\mathsf{f}_i, g), (h_1, \mathbf{1}_{\mathbb{G}}), (\mathbf{1}_{\mathbb{G}}, h_2)); \omega)$;
9: $\psi_i \leftarrow \mathsf{NIZK.Prf}((S, \mathsf{cs}_i), (g, h_2); s)$; //same for all i
10: **send** $(\mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)$ to $\mathcal{P}_i$;

Output assignment (both roles):

11: **for** $i \in \mathcal{I}$:
    $\mathcal{P}_i$ **sets** $\mathsf{ptk}_i = \mathsf{pvk}_i = \mathsf{sh}_i = (x_i, \mathsf{p}_i)$;
    $\mathcal{D}$ **sets** $\mathsf{dtk}_i = \mathsf{dvk}_i = \perp$;
12: **for** $i \in \mathcal{J}$: //$\mathcal{D}$ plays both roles
    set $\mathsf{ptk}_i = \mathsf{pvk}_i = \mathsf{sh}_i = \perp$ **and** $\mathsf{dtk}_i = \mathsf{dvk}_i = (x_i, \mathsf{p}_i)$;

Fig. 10: Sharing protocol of TSS-PV. For dummy indices $i \in \mathcal{J}$, $\mathcal{D}$ generates ps_i and executes the party steps locally; all n protocol instances are routed through the AAC channel and are externally indistinguishable. Each instance produces a transcript $T_i = (\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)$. The proof ψ_i is verifiable only after S is recovered via Rec.

Remark 6 (Uniform Secret). Our construction requires the dealer to sample $s \leftarrow\!\!\$\ \mathbb{Z}_q$ and set $S = g^s$, rather than choosing $S \in \mathbb{G}$ arbitrarily. This is a structural necessity: the dealer uses s as a witness in three places, to compute the commitment $\mathsf{cs} = h_2^s$, to form the per-index share proof π_i, and to form the reconstruction proof ψ. An arbitrary group element $S \in \mathbb{G}$ without a known discrete logarithm would make all three operations infeasible, requiring a fundamentally different public verification mechanism.

Reconstruction Algorithm. The algorithm Rec in Figure 11 recovers the secret $S = g^s$ from t arbitrary valid shares using Lagrange interpolation on the exponents. Note, in the reconstruction phase, the combiner runs VerSS on shares before running Rec.

$\underline{\mathsf{Rec}(\mathsf{SH}) \to S/\perp :}$

1: **if** $|\mathsf{SH}| < t$ **then return** $\perp$;
2: **parse** $\mathsf{SH} = \{\mathsf{sh}_i\}_{i\in[t]}$;
3: $\forall i \in [t]$: **parse** $\mathsf{sh}_i = (x_i, \mathsf{p}_i)$;
4: **return** $S \leftarrow \prod_{j=1}^{t} \mathsf{p}_j^{\Delta_j}$, **where** $\Delta_j \leftarrow \prod_{i=1, i\neq j}^{t} \frac{x_i}{x_i - x_j}$;

Fig. 11: The reconstruction algorithm Rec of TSS-PV.

Share Distribution Verification. The public algorithm VerSD (Figure 12) enables external parties to confirm that the dealer used a single global $\omega = (r, s)$ across all n indices: the consistency check (Line 2) enforces one commitment, and the per-index NIZK check (Line 3) ensures each p_i was formed under that same commitment.

Share Submission Verification. Given a submitted share $\mathsf{sh}_i = (x_i, \mathsf{p}'_i)$ and the public transcript T, algorithm VerSS (Figure 12) checks that only the legitimate holder of evaluation point x_i could have produced this share: it recomputes $\tilde{\mathsf{f}}_i$ from x_i (Line 4) and verifies that p'_i matches the dealer's committed value and passes the NIZK (Line 5).

Secret Verification Algorithm. The public algorithm VerRS (Figure 12) lets any party verify a candidate secret S' against the dealer's commitment without access to any shares: if S' is the secret, the global proof ψ_1 (as ψ_i is identical for all $i \in [n]$) passes the verification as $S' = g^s \wedge \mathsf{cs}_1 = h_2^s$ for some witness s.

Remark 7 (Verification Algorithms Relationship). Algorithm VerSS re-runs the NIZK verification on index i, even though VerSD also verifies π_i. This redundancy is intentional: VerSS is designed to be sound even for verifiers who do not execute VerSD beforehand. Similarly, both VerSS and VerRS re-check global consistency (Line 2), ensuring each algorithm is self-contained. In settings where $\mathsf{VerSD}(\mathsf{T}) = 1$ is already guaranteed, the consistency check and the call to

$\mathsf{NIZK.Vr}$ inside VerSS are logically implied by the global check together with the pre-share value- and share-binding tests ($\tilde{\mathsf{f}}_i = \mathsf{f}_i$ and $\mathsf{p}'_i = \mathsf{p}_i$), and may be omitted as an implementation-level optimization. Our security definitions and proofs, however, treat each verification algorithm as a standalone predicate that does not assume prior execution of the others.

$\underline{\mathsf{VerSD}(\mathsf{T}) \to \{0,1\}}$:

parse $\mathsf{T} = \{(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)\}_{i\in[n]}$, where $\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)\ \forall i \in [n]$;
if $\exists i, j \in [n] : \psi_i \neq \psi_j \vee \mathsf{C}_i \neq \mathsf{C}_j$ **then return** 0;
if $\exists i \in [n]$:
  $\mathsf{NIZK.Vr}((\mathsf{p}_i, \mathsf{cr}_i, \mathsf{cs}_i), ((\mathsf{f}_i, g), (h_1, \mathbf{1}_{\mathbb{G}}), (\mathbf{1}_{\mathbb{G}}, h_2)), \pi_i) \neq 1$
  then return 0;
return 1;

$\underline{\mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T}) \to \{0,1\}}$:

parse $\mathsf{T} = \{(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)\}_{i\in[n]}$, where $\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)\ \forall i \in [n]$;
parse $\mathsf{sh}_i = (x_i, \mathsf{p}'_i)$;
if $\exists i, j \in [n] : \psi_i \neq \psi_j \vee \mathsf{C}_i \neq \mathsf{C}_j$ **then return** 0;
$\tilde{\mathsf{f}}_i \leftarrow \prod_{j=1}^{t-1} \alpha_j^{x_i^j}$;
if $(\tilde{\mathsf{f}}_i \neq \mathsf{f}_i) \vee (\mathsf{p}'_i \neq \mathsf{p}_i) \vee (\mathsf{NIZK.Vr}((\mathsf{p}'_i, \mathsf{cr}_i, \mathsf{cs}_i), ((\tilde{\mathsf{f}}_i, g), (h_1, \mathbf{1}_{\mathbb{G}}), (\mathbf{1}_{\mathbb{G}}, h_2)), \pi_i) \neq 1)$
  then return 0
return 1;

$\underline{\mathsf{VerRS}(S', \mathsf{T}) \to \{0,1\}}$:

parse $\mathsf{T} = \{(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)\}_{i\in[n]}$, where $\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)\ \forall i \in [n]$;
if $\exists i, j \in [n] : \psi_i \neq \psi_j \vee \mathsf{C}_i \neq \mathsf{C}_j$ **then return** 0;
if $\mathsf{NIZK.Vr}((S', \mathsf{cs}_1), (g, h_2), \psi_1) \neq 1$ **then return** 0;
return 1;

Fig. 12: The public verification algorithms. VerSD does not verify π_i^{PC} as π_i^{PC} attests party honesty which VerSS captures.

Tracing Algorithm. Given tracing keys TK and the public transcript T, the tracer identifies which parties have their shares embedded in the reconstruction box R. The key insight is that a query $\mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})$ succeeds (i.e., $\mathsf{VerRS} = 1$) if and only if sh_i is *not* in the box's internal leakage set E; by Def. 10, queries containing a leaked share fail. The tracer thus tests each real index and removes those whose queries succeed, leaving only indices whose shares are embedded in R. The output consists of the accused set $\mathcal{C}'$ together with evidence π_{trace} containing the corresponding shares. The dummy set size $t-f-1$ is chosen so that DSH always falls short of threshold alone, forcing R to depend on the tested share sh_i to reconstruct.

Tracing Verification Algorithm. We adopt a replay-based design so that any auditor can independently exercise the reconstruction box without trusting the tracer's logs. Given $(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}})$ and black-box access to R, the judge validates provenance (Lines 3–6) then replays R on each accused index (Lines 7–8): a successful query proves that index innocent, so any such result causes the entire accusation set $\mathcal{C}'$ to be rejected, binding the accusation to the observable behavior of R under publicly verifiable inputs.

Note, in our construction, the information required to trace a share is identical to the information required to verify it; hence tk_i and vk_i contain the same payload (the cleartext share). However, we maintain the distinction in the syntax to adhere to the generic TSS framework [10], supporting future instantiations where tracing and verification capabilities may rely on distinct cryptographic material.

$\underline{\mathsf{Trace}^{\mathsf{R}}(\mathsf{TK}, \mathsf{T}) \to (\mathcal{C}', \pi_{\mathsf{trace}})\text{:}}$

1: **parse** $\mathsf{TK} = \{(\mathsf{ptk}_i, \mathsf{dtk}_i)\}_{i\in[n]}$;
2: $\mathcal{I} \coloneqq \{i \in [n] : \mathsf{ptk}_i \neq \bot\}$, $\mathcal{J} \coloneqq \{i \in [n] : \mathsf{dtk}_i \neq \bot\}$
3: $\forall i \in \mathcal{I}$: **parse** $\mathsf{ptk}_i = \mathsf{sh}_i$, $\forall j \in \mathcal{J}$: **parse** $\mathsf{dtk}_j = \mathsf{sh}_j$;
4: $\mathsf{DSH} \leftarrow\!\!\$\ \{\mathsf{sh}_j\}_{j\in\mathcal{J}},\ |\mathsf{DSH}| = t - f - 1$;
5: $\mathcal{C}' \coloneqq \mathcal{I}$;
6: **for** $i \in \mathcal{I}$:
  $S_i \leftarrow \mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})$;
  if $\mathsf{VerRS}(S_i, \mathsf{T}) = 1$ **then set** $\mathcal{C}' \coloneqq \mathcal{C}' \setminus \{i\}$;
7: **set** $\pi_{\mathsf{trace}} \coloneqq \{\mathsf{sh}_i\}_{i\in\mathcal{C}'}$;
8: **return** $(\mathcal{C}', \pi_{\mathsf{trace}})$;

$\underline{\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}}) \to \{0, 1\}\text{:}}$

1: **parse** $\mathsf{T} = \{(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)\}_{i\in[n]}$, where $\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)\ \forall i \in [n]$;
2: **if** $\exists i, j \in [n] : \psi_i \neq \psi_j \vee \mathsf{C}_i \neq \mathsf{C}_j$ **then return** 0;
3: **parse** $\mathsf{VK} = \{(\mathsf{pvk}_i, \mathsf{dvk}_i)\}_{i\in[n]}$;
4: $\mathcal{I} \coloneqq \{i \in [n] : \mathsf{pvk}_i \neq \bot\}$, $\mathcal{J} \coloneqq \{i \in [n] : \mathsf{dvk}_i \neq \bot\}$
5: $\forall i \in \mathcal{I}$: **parse** $\mathsf{pvk}_i = \mathsf{sh}_i$, $\forall j \in \mathcal{J}$: **parse** $\mathsf{dvk}_j = \mathsf{sh}_j$;
6: **if** $\big(\mathcal{C}' \not\subseteq \mathcal{I}\big) \vee \big(\pi_{\mathsf{trace}} \neq \{\mathsf{sh}_i\}_{i\in\mathcal{C}'}\big) \vee (\mathsf{VerSD}(\mathsf{T}) \neq 1)$
  $\vee(\exists i \in \mathcal{I} : \mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T}) \neq 1)$
  $\vee(\exists j \in \mathcal{J} : \mathsf{VerSS}(\mathsf{sh}_j, \mathsf{T}) \neq 1)$ **then return** 0;
7: $\mathsf{DSH} \leftarrow\!\!\$\ \{\mathsf{sh}_i\}_{i\in\mathcal{J}} : |\mathsf{DSH}| = t - f - 1$;
8: **if** $\exists i \in \mathcal{C}' : \big(S' \leftarrow \mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})\big) \wedge \big(\mathsf{VerRS}(S', \mathsf{T}) = 1\big)$
  then return 0;
9: **return** 1;

Fig. 13: The tracing $\mathsf{Trace}^{\mathsf{R}}$ and tracing verification algorithms $\mathsf{TrVer}^{\mathsf{R}}$. For simplicity, we assume the tracer and the judge have the coalition size f, which is determined in tracing trigger sub-phase.

6.4 Security analysis

In this section, we show that our TSS-PV satisfies security notions defined in Section 5.

Theorem 6.3 (Perfect Correctness). *Assume that the underlying NIZK proof systems are complete. Then our TSS-PV scheme satisfies perfect correctness.*

Proof. In the experiment $\mathbf{ExpCor}^{n,t}_{\mathsf{TSS\text{-}PV}}$, the honest execution of Share yields public parameters $\mathsf{pp} = (1^\lambda, n, t, \rho)$, transcripts $\mathsf{T} = \{T_i\}_{i\in[n]}$, and shares $\Sigma = \{\mathsf{sh}_i\}_{i\in[n]}$.

The dealer samples $(r, s) \leftarrow\!\!\$\ \mathbb{Z}_q^2$, sets $S \leftarrow g^s$, and for each $i \in [n]$ computes:

$$\mathsf{f}_i = \prod_{j=1}^{t-1} \alpha_j^{x_i^j} \quad \text{and} \quad \mathsf{p}_i = \mathsf{f}_i^r \cdot g^s.$$

Let $\alpha_j = g^{a_j}$ where $a_j \in \mathbb{Z}_q$ is the discrete logarithm of the parameter α_j. Substituting this into the expression for p_i:

$$\mathsf{p}_i = \left(\prod_{j=1}^{t-1} g^{a_j x_i^j}\right)^r \cdot g^s = g^{\sum_{j=1}^{t-1}(ra_j)x_i^j + s}.$$

This form reveals that $\mathsf{p}_i = g^{\tilde{q}(x_i)}$, where $\tilde{q}(\mathsf{X}) = s + \sum_{j=1}^{t-1}(ra_j)\mathsf{X}^j$ is a polynomial of degree at most $t-1$ over $\mathbb{Z}_q$ with constant term $\tilde{q}(0) = s$.

Let $\mathcal{V} \subseteq [n]$ be any subset of valid shares with $|\mathcal{V}| = t$, and let $\mathsf{SH} = \{(x_i, \mathsf{p}_i)\}_{i\in\mathcal{V}}$. The protocol explicitly enforces that all f_i are unique and $\mathsf{f}_i \neq 1_\mathbb{G}$. Therefore, the indices $\{x_i\}_{i\in\mathcal{V}}$ are pairwise distinct and non-zero, ensuring Lagrange interpolation applies. Define the Lagrange basis polynomials at zero as $\Delta_i = \prod_{k\in\mathcal{V}, k\neq i} \frac{x_k}{x_k - x_i}$. By the properties of polynomial interpolation, $\sum_{i\in\mathcal{V}} \Delta_i \tilde{q}(x_i) = \tilde{q}(0) = s$. The reconstruction algorithm $\mathsf{Rec}(\mathsf{SH})$ computes:

$$\prod_{i\in\mathcal{V}} \mathsf{p}_i^{\Delta_i} = \prod_{i\in\mathcal{V}} g^{\Delta_i \cdot \tilde{q}(x_i)} = g^{\sum_{i\in\mathcal{V}} \Delta_i \tilde{q}(x_i)} = g^s = S.$$

Thus, Rec correctly recovers the secret.

For verification, each honest party generates π_i^{PC} using $\mathsf{PC.Prf}(\mathsf{pp}^{\mathsf{PC}}, x_i)$. By the completeness of the PC-PoK, the dealer accepts. Similarly, the dealer generates π_i and ψ honestly. By the completeness of the underlying NIZK system, $\mathsf{VerSD}(\mathsf{T})$, $\mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T})$, and $\mathsf{VerRS}(S, \mathsf{T})$ all return 1.

Theorem 6.4 (Share Distribution Verifiability). *Assume that the soundness of* NIZK_π *and share-submission verifiability (Def. 5) both hold. Then our* TSS-PV *scheme satisfies the verifiable distribution for the security parameter* λ.

Proof. Let $\mathcal{A}$ be an adversary of $\mathbf{ExpVSD}_{\mathcal{A},\mathsf{TSS\text{-}PV}}$, acting as a malicious dealer. $\mathcal{A}$ directly outputs a transcript $\mathsf{T} = \{T_i\}_{i\in[n]}$, where $T_i = (\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}}, \mathsf{p}_i, \pi_i, \psi_i, \mathsf{C}_i)$,

shares $\Sigma = \{\mathsf{sh}_i\}_{i\in[n]}$, and index sets $I_0, I_1 \subseteq [n]$ with $|I_0| = |I_1| = t$. Assume $\mathcal{A}$ wins. Then there exist index sets $I_0, I_1 \subseteq [n]$ with $|I_0| = |I_1| = t$, such that $\mathsf{SH}_b = \{\mathsf{sh}_i\}_{i\in I_b}$ satisfies $\mathsf{VerSD}(\mathsf{T}) = 1$ and $S_0 \neq S_1$, where $S_b = \mathsf{Rec}(\mathsf{SH}_b)$. From $\mathsf{VerSD}(\mathsf{T}) = 1$ and the soundness of the proofs $\{\pi_i\}$, there exist $r, s \in \mathbb{Z}_q$ such that, for every $i \in [n]$,

$$\mathsf{p}_i = \mathsf{f}_i^{\,r} \cdot g^{\,s}, \mathsf{cr}_i = h_1^{\,r}, \text{ and } \mathsf{cs}_i = h_2^{\,s},$$

and the commitments $\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)$ are identical across all indices; equivalently, the same global $\omega = (r, s)$ is used.

Let $\mathcal{V}(\mathsf{T}, \Sigma) = \{\mathsf{sh}_i \in \Sigma : \mathsf{VerSS}(\mathsf{sh}_i, \mathsf{T}) = 1\}$, which has size at least t. Since SH_0, $\mathsf{SH}_1 \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$, every share used to reconstruct S_0 and S_1 is consistent with the same $\omega = (r, s)$. Hence, the reconstruction is uniquely determined by $S = g^s$, and

$$\mathsf{Rec}(\mathsf{SH}_0) = \mathsf{Rec}(\mathsf{SH}_1) = S,$$

contradicting $S_0 \neq S_1$, unless VerSS is violated or some proof π_i accepts a false statement, which occurs with probability $\epsilon_\pi^{\mathsf{snd}}$. Consequently,

$$\mathsf{Adv}^{\mathsf{vsd}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) \leq \mathsf{Adv}^{\mathsf{vss}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) + n \cdot \epsilon_\pi^{\mathsf{snd}},$$

which is negligible.

Theorem 6.5 (Share Submission Verifiability). *Assume that the* 1*-DLOG assumption holds. Then our TSS-PV scheme satisfies verifiable submission for the security parameter* λ*.*

Proof. In $\mathbf{ExpVSS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}$ with an honest dealer, the adversary $\mathcal{A}$ can succeed only in one of the following events: (i) W_0: it outputs distinct valid submissions sh_{i^*} and sh'_{i^*} for the same index, or (ii) W_1: it outputs t valid submissions that reconstruct to $S' \neq S$.

• (**Event** W_0) Assume $\mathcal{A}$ outputs two accepting shares $\mathsf{sh}_{i^*} = (x, y)$ and $\mathsf{sh}'_{i^*} = (x', y')$ for the same transcript $T_{i^*} = (\mathsf{f}_{i^*}, \boldsymbol{V}_{i^*}, \pi_{i^*}^{\mathsf{PC}}, \mathsf{p}_{i^*}, \pi_{i^*}, \psi_{i^*}, \mathsf{C}_{i^*})$. By the definition of VerSS,

$$y = y' = \mathsf{p}_{i^*} \quad \text{and} \quad \prod_{j=1}^{t-1} \alpha_j^{x^j} = \prod_{j=1}^{t-1} \alpha_j^{x'^j} = \mathsf{f}_{i^*}.$$

To reduce to the 1-DLOG problem, let $(g, Z = g^z)$ be a 1-DLOG instance. A reduction algorithm $\mathcal{B}$ simulates $\rho = (\alpha_1, \ldots, \alpha_{t-1})$ as

$$\alpha_j = Z^{s_j} g^{t_j}, \quad \text{where } s_j, t_j \leftarrow\!\!\$\ \mathbb{Z}_q \ \ (j \in [t{-}1]).$$

$\mathcal{B}$ runs the dealer honestly otherwise. Then the exponents $\{e_j = s_j z + t_j\}$ are independently uniform in $\mathbb{Z}_q$, so ρ is identically distributed to the real setup. Let $d_j = x^j - x'^j$ (not all zero as $x \neq x'$). Then discrete logarithms for f_{i^*} in the equality yield

$$\sum_{j=1}^{t-1} (s_j z + t_j) d_j = 0.$$

Hence, $\mathcal{B}$ recovers the solution of the 1-DLOG problem by

$$z = -\sum_{j=1}^{t-1} t_j d_j \cdot \left(\sum_{j=1}^{t-1} s_j d_j\right)^{-1}.$$

Clearly, the denominator above becomes 0 with probability at most $1/q$ as $\{s_1, \ldots, s_{t-1}\}$ are uniform and independent. Hence, $\Pr[W_0](1 - 1/q) \leq \epsilon_{1\text{-}dl}$, where $\epsilon_{1\text{-}dl}$ is the probability that the 1-DLOG assumption is broken.
• (**Event** W_1) Since the dealer is honest, the global proof ψ, all proofs $\{\pi_i\}$, and commitments $\{\mathsf{C}_i = (\mathsf{cr}_i, \mathsf{cs}_i)\}$ are honestly generated. Verification of ψ together with identical commitments across indices ensures that an identical $\omega = (r, s)$ is used for every i. This implies that every accepted share embeds the common secret $S = g^s$. Hence, this case never happens; i.e., $\Pr[W_1] = 0$.

Overall,

$$\mathsf{Adv}^{\mathsf{vss}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = \Pr[W_0 \vee W_1] \leq \frac{\epsilon_{1\text{-}dl}}{1 - 1/q}.$$

Theorem 6.6 (Secret Reconstruction Verifiability). *Assume that the soundness of* NIZK_ψ *holds. Then our TSS-PV scheme satisfies verifiable secret reconstruction for the security parameter* λ*.*

Proof. In $\mathbf{ExpVRS}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$, the adversary $\mathcal{A}$ outputs a transcript $\mathsf{T} = \{T_i\}_{i\in[n]}$ and two distinct candidates S_0, S_1 with $S_0 \neq S_1$. Suppose $\mathcal{A}$ wins. Then $\mathsf{VerRS}(S_0, \mathsf{T}) = \mathsf{VerRS}(S_1, \mathsf{T}) = 1$. By the definition of VerRS, all global commitments and proofs are consistent; in particular, $\mathsf{C}_i = (\mathsf{cr}, \mathsf{cs})$ and $\psi_i = \psi$ do not depend on i. Moreover, acceptance for S_b means that, by soundness of the NIZK_ψ, there exists $s_b \in \mathbb{Z}_q$ such that

$$S_b = g^{s_b} \quad \text{and} \quad \mathsf{cs} = h_2^{s_b} \quad \text{where} \quad b \in \{0, 1\}.$$

By the soundness of the underlying NIZK system, there exist witnesses s_0, s_1 satisfying the above equalities. Since $\mathsf{cs} = h_2^{s_0} = h_2^{s_1}$, it follows that $s_0 = s_1$ and, therefore, $S_0 = g^{s_0} = g^{s_1} = S_1$. This contradicts $S_0 \neq S_1$.

Hence, a successful $\mathcal{A}$ would break the soundness for ψ, which occurs with probability at most $\epsilon^{\mathsf{snd}}_\psi$. That is,

$$\mathsf{Adv}^{\mathsf{vrec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) \leq \epsilon^{\mathsf{snd}}_\psi,$$

which is negligible.

Theorem 6.7 (Dealer Secrecy). *Assume that the DDH assumption holds in* $\mathbb{G}$*, the underlying NIZK proof systems are zero-knowledge in the ROM, and the PC-PoK proof system is knowledge-sound in the ROM. Then our* TSS-PV *scheme satisfies dealer secrecy for the security parameter* λ*.*

Proof. Let $\mathcal{B}$ be a DDH distinguisher given $(g, A = g^a, B = g^b, Z)$, where $Z = g^{ab}$ or $Z \leftarrow\!\!\$\ \mathbb{G}$. $\mathcal{B}$ simulates $\mathbf{ExpDSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$ for an adversary $\mathcal{A}$ as follows.

$\mathcal{B}$ sets $h_1 = g^e$ for $e \leftarrow\!\!\$\ \mathbb{Z}_q$ and $\mathsf{cr} = A^e = h_1^a$, implicitly embedding the dealer randomness $r = a$. Next, $\mathcal{B}$ samples $h_2 \leftarrow\!\!\$\ \mathbb{G}$ and $s \leftarrow\!\!\$\ \mathbb{Z}_q$, and sets $\mathsf{cs} = h_2^s$ and $S = g^s$. To generate $\rho = \boldsymbol{\alpha} \in \mathbb{G}^{t-1}$, for all $j \in [t-1]$, $\mathcal{B}$ samples secret scalars $c_j, w_j \leftarrow\!\!\$\ \mathbb{Z}_q$ and sets

$$\alpha_j = B^{c_j} \cdot g^{w_j} = g^{bc_j + w_j}.$$

Note that c_j is information-theoretically hidden from $\mathcal{A}$ by the one-time pad w_j. $\mathcal{B}$ then sends pp to $\mathcal{A}$. Before the share and proof generation of Share, $\mathcal{A}$ outputs, in a single round, a vector of transcripts $\{(\mathsf{f}_i, \boldsymbol{V}_i, \pi_i^{\mathsf{PC}})\}_{i \in \mathcal{C}}$, where $\mathcal{C}$ is the index set of corrupted parties. $\mathcal{B}$ verifies all proofs and checks that $\mathsf{f}_i \neq 1_{\mathbb{G}}$ for each i; if any check fails, $\mathcal{B}$ aborts. $\mathcal{B}$ then rewinds $\mathcal{A}$ to extract the witness vector $\{x_i\}_{i \in \mathcal{C}}$. This is possible in expected polynomial time, due to the generalized forking lemma [3]. The extraction succeeds except with probability ϵ_{fork}. For honest parties $i \in \{t, \ldots, n\}$, $\mathcal{B}$ samples $x_i \leftarrow\!\!\$\ \mathbb{Z}_q \setminus \{0\}$ and generates valid PC-PoK proofs. While the share and proof generation for $i \in [n]$, using the known or extracted x_i, $\mathcal{B}$ computes the pre-share $\mathsf{f}_i = \prod_{j=1}^{t-1} \alpha_j^{x_i^j}$ and the simulated share

$$\mathsf{p}_i = \left(\prod_{j=1}^{t-1} (Z^{c_j} \cdot A^{w_j})^{x_i^j} \right) \cdot S.$$

The proofs π_i and ψ are simulated via the zero-knowledge simulator, without requiring knowledge of a. The global commitment $\mathsf{C} = (\mathsf{cr}, \mathsf{cs})$ and proof ψ are sent identically to all parties, satisfying the protocol's consistency requirement.

The simulated p_i is distributed depending on Z, as follows:

• **(Real:** $Z = g^{ab}$**)** Since $Z^{c_j} \cdot A^{w_j} = g^{abc_j + aw_j} = \alpha_j^a$, the equality $\mathsf{p}_i = \mathsf{f}_i^a \cdot S$ holds, which is identically distributed to a real execution with dealer randomness $r = a$.

• **(Random:** $Z = g^{ab+z'}$**)** Let $z' \neq 0$. Then the share becomes

$$\mathsf{p}_i = \mathsf{f}_i^a \cdot S \cdot g^{z' M_i}, \quad \text{where } M_i = \sum_{j=1}^{t-1} c_j x_i^j.$$

Since ρ statistically hides $\boldsymbol{c} = (c_1, ..., c_{t-1})$, $\mathcal{A}$'s choice of x_i is independent of $\boldsymbol{c}$. Moreover, the protocol enforces $x_i \neq 0$, so the vector $(x_i, x_i^2, \ldots, x_i^{t-1})$ is non-zero. Hence, M_i is uniform which implies $\Pr[M_i = 0] = 1/q$. When $M_i \neq 0$, the term $g^{z' M_i}$ acts as a random mask of S, yielding p_i uniformly distributed and independent of S.

Finally, $\mathcal{A}$ outputs S^*; $\mathcal{B}$ outputs 1 if $S^* = S$, and 0 otherwise. The simulation fails only if the following events occur: (1) the extraction fails with probability ϵ_{fork}, (2) $M_i = 0$ for some party with probability at most n/q by the union bound, or (3) $\mathcal{A}$ distinguishes the simulated ZK proofs with probability ϵ_{zk}. Hence,

$$\mathsf{Adv}^{\mathsf{dsec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) \leq \epsilon_{\mathsf{ddh}} + \epsilon_{\mathsf{zk}} + \epsilon_{\mathsf{fork}} + \frac{n}{q}.$$

Theorem 6.8 (Party Secrecy). *Assume that the $(t-1)$-DLOG assumption holds in $\mathbb{G}$ and the PC-PoK proof system is zero-knowledge in the ROM. Then our TSS-PV scheme satisfies party secrecy for the security parameter λ.*

Proof. Let $\mathcal{B}$ be a reduction algorithm given a $(t-1)$-DLOG instance $(g, Z_1, \ldots, Z_{t-1})$ where $Z_j = g^{z^j}$ for a hidden $z \in \mathbb{Z}_q$. Since in $\mathbf{ExpPSec}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$, the adversary $\mathcal{A}$ chooses the parameters n and t, $\mathcal{B}$ first guesses a value $t' \leftarrow\!\!\$\ t_{\mathsf{max}}$ (which is a polynomial bound in λ). If $t \neq t'$ $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ continues the following simulation:

After obtaining the target index i^* from $\mathcal{A}$, $\mathcal{B}$ samples $s_1, \ldots, s_{t-1} \leftarrow\!\!\$\ \mathbb{Z}_q$ and sets $\alpha_j = g^{s_j}$ for $j \in [t-1]$. Then $\rho = \boldsymbol{\alpha} \in \mathbb{G}^{t-1}$ is published. As the honest party i^*, $\mathcal{B}$ executes Share against the malicious dealer controlled by $\mathcal{A}$, and sends

$$\mathsf{f}_{i^*} = \prod_{j=1}^{t-1} Z_j^{s_j} = g^{\sum_{j=1}^{t-1} s_j z^j},$$

together with $\boldsymbol{V}_{i^*} = (Z_1, \ldots, Z_{t-1})$. Since $\mathcal{B}$ does not know z, by zero-knowledge of PC-PoK (Theorem 6.2), it simulates $\pi_{i^*}^{\mathsf{PC}}$ without the witness.

This simulation implicitly fixes the honest secret of i^* to $x_{i^*} = z$, while preserving the distribution of a real execution. If $\mathcal{A}$ finally outputs $\widehat{\mathsf{sh}}_{i^*}$ including $\hat{x}$, $\mathcal{B}$ checks $g^{\hat{x}^j} = Z_j$ for all $j \in [t-1]$. If all checks pass, it outputs $\hat{x}$ as the solution to the $(t-1)$-DLOG problem. By the winning condition, a success of $\mathcal{A}$ implies $\hat{x} = x_{i^*} = z$. Thus, whenever $\mathcal{A}$ wins, $\mathcal{B}$ solves the problem. Consequently,

$$\mathsf{Adv}^{\mathsf{psec}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) \leq \epsilon_{(t-1)\text{-}\mathsf{dl}} \cdot t_{\mathsf{max}} + \epsilon_{\mathsf{zk}},$$

where $\epsilon_{(t-1)\text{-}\mathsf{dl}}$ denotes the advantage against $(t-1)$-DLOG and ϵ_{zk} the zero-knowledge distinguishing advantage.

Theorem 6.9 (Traceability). *Assume that the correctness of our scheme holds, and the reconstruction box* R *is verifiability-aware and perfect. Then our TSS-PV scheme satisfies traceability for the security parameter* λ*.*

Proof. Let $\mathcal{A}$ be the adversary against $\mathbf{ExpTrace}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$. After receiving pp and outputs of honest Share executions, $\mathcal{A}$ adaptively outputs a coalition $\mathcal{C} \subseteq [n]$, a reconstruction box R, and the party outputs $\{\mathsf{op}_i\}_{i\in[n]}$, where $|\mathcal{C}| = f$.

Let $\Sigma = \{\mathsf{sh}_i\}_{i\in[n]}$ be the set of shares generated by Share, and let $\mathcal{V}(\mathsf{T}, \Sigma) = \{\mathsf{sh} \in \Sigma : \mathsf{VerSS}(\mathsf{sh}, \mathsf{T}) = 1\}$. $\mathcal{A}$'s success requires that the trigger $\mathcal{T}(\mathsf{T}, \Sigma, \mathsf{R}; \mathcal{C})$ holds and tracing fails to identify. Consider $\mathsf{E} = \{\mathsf{sh}_i : i \in \mathcal{C}\}$. By the definition of a verifiability-aware perfect box R (Definition 10), there exists a set $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$ with $|\mathsf{SH}| = t - f$ and $\mathsf{SH} \cap \mathsf{E} = \emptyset$, such that $\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1$.

Let TK and VK denote, respectively, the tracing key and verification key collections generated in the same Share executions. Using TK, the tracer $\mathsf{Trace}^{\mathsf{R}}$ fixes a dummy set DSH of size $t - f - 1$. For each $i \in [n]$, consider the set $\mathsf{DSH} \cup \{\mathsf{sh}_i\}$. (i) if $i \in \mathcal{C}$ (i.e., the index is indeed corrupted) then $\mathsf{sh}_i \in \mathsf{E}$ and thus $\mathsf{DSH} \cup \{\mathsf{sh}_i\}$ intersects E; the verification algorithm VerRS does not accept $\mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})$. (ii) Otherwise, i.e., $\mathsf{sh}_i \notin \mathsf{E}$, $\mathsf{DSH} \cup \{\mathsf{sh}_i\}$ is a valid set disjoint from E. Therefore, R returns a secret S_i which VerRS accepts, and sets $\mathcal{C}' = \mathcal{C}' \setminus \{i\}$, where $\mathcal{C}'$ is the accusation set and is initialized by $\mathcal{C}' := \mathcal{I}$. Aggregating these outcomes over all i yields the exact coalition $\mathcal{C}' = \mathcal{C}$ and evidence π_{trace} produced by $\mathsf{Trace}^{\mathsf{R}}$.

Since the keys and transcripts were generated by the honest procedure, $\mathsf{VerSD}(\mathsf{T}) = 1$. Moreover, every share in $\pi_{\mathsf{trace}} = \{\mathsf{sh}_i\}_{i\in\mathcal{C}}$ lies in $\mathcal{V}(\mathsf{T}, \Sigma)$ and hence, passes VerSS. Finally, the accept/reject pattern attested in π_{trace} matches the behavior guaranteed by the verifiability-aware semantics of R together with VerRS. The public verifier can recompute these reconstructions using VK and T, thus accepting $\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}}) = 1$.

$\mathcal{A}$ can succeed only if a correctness failure occurs in one of VerSD, VerRS, or VerSS. Consequently,

$$\mathsf{Adv}^{\mathsf{trace}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) \le \epsilon_{\mathsf{cor}},$$

where ϵ_{cor} denotes the maximal correctness error of these verifiers, which equals 0 for our scheme by Theorem 6.3.

Theorem 6.10 (Non-imputability). *Assume that the party secrecy holds, all three verifiability properties (VerSD, VerSS, and VerRS) hold, and the reconstruction box* R *is verifiability-aware and perfect. Then our TSS-PV scheme satisfies non-imputability for the security parameter* λ*.*

Proof. In $\mathbf{ExpNI}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda)$, the adversary $\mathcal{A}$, given a public parameter pp, fixes a target index i^* and acts as the dealer. Let x_{i^*} denote the private input of the honest player indexed i^*. $\mathcal{A}$ runs Share with the honest party. This yields a transcript $T_{i^*} = (\mathsf{f}_{i^*}, \mathsf{p}_{i^*}, \pi_{i^*}, \psi_{i^*}, \mathsf{C}_{i^*})$ associated with $\mathsf{sh}_{i^*} = (x_{i^*}, \mathsf{p}_{i^*})$. At the end, $\mathcal{A}$'s final output determines an accused set $\mathcal{C}$, evidence π_{trace}, shares Σ, verification keys VK (which encode the shares) and transcripts T, and a reconstruction box R. The winning event is

$$W = (i^* \in \mathcal{C}) \ \wedge \ (\mathsf{TrVer}^{\mathsf{R}}(\mathsf{VK}, \mathsf{T}, \mathcal{C}, \pi_{\mathsf{trace}}) = 1).$$

Let W_0 be the event that x_{i^*} is leaked to $\mathcal{A}$, and $W_1 = \neg W_0$.

• (**Event** W_0) By the party secrecy, $\mathcal{A}$ can obtain x_{i^*} except with at most ϵ_{psec} (the probability of violating the party secrecy); that is, $\Pr[W_0] \le \epsilon_{\mathsf{psec}}$.

• (**Event** W_1) Let $\mathcal{V}(\mathsf{T}, \Sigma)$ be a set of shares accepted by VerSS on T. Since R is verifiability-aware and perfect (Definition 10), there exists $\mathsf{E} = \{\mathsf{sh}_i : i \in \mathcal{C}\}$ such that, for any $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma) \setminus \mathsf{E}$ with $|\mathsf{SH}| = t - f$,

$$\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1] = 1.$$

In $\mathsf{TrVer}^{\mathsf{R}}$, after provenance checks via VerSD and VerSS, a dummy set DSH of size $t - f - 1$ is sampled from VK, which lies in $\mathcal{V}(\mathsf{T}, \Sigma) \setminus \mathsf{E}$ by the verifiability-awareness. Consider $\mathsf{SH}^* = \mathsf{DSH} \cup \{\mathsf{sh}_{i^*}\}$. Since party i^* is honest, $\mathsf{sh}_{i^*} \in \mathcal{V}(\mathsf{T}, \Sigma) \setminus \mathsf{E}$ except with the probability that some verifiability check fails. Hence $\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}^*), \mathsf{T}) = 1] = 1$ unless VerSD, VerSS or VerRS is violated. Therefore, $\Pr[W|W_1] \le \epsilon_{\mathsf{vss}} + \epsilon_{\mathsf{vsd}} + \epsilon_{\mathsf{vrs}}$, where ϵ_i for $i \in \{\mathsf{vss}, \mathsf{vsd}, \mathsf{vrs}\}$ denotes the maximum success probability of violating the corresponding verifiability property.

Combining the two cases,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ni}}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) &= \Pr[W] = \Pr[W \ \wedge \ W_0] + \Pr[W \ \wedge \ W_1] \\ &\le \Pr[W_0] + \Pr[W|W_1] \\ &\le \epsilon_{\mathsf{psec}} + \epsilon_{\mathsf{vss}} + \epsilon_{\mathsf{vsd}} + \epsilon_{\mathsf{vrs}},\end{aligned}$$

which is eventually negligible.

7 Performance

This section evaluates TSS-PV along three axes.

(Q1) Do measured costs match the analytic predictions?
(Q2) Where is the bottleneck, and how do costs scale with the number of parties?
(Q3) Are accountability costs practical at realistic party counts?

We first present closed-form cost expressions (Section 7.1), then validate them against a prototype implementation (Section 7.2).

7.1 Analytic Costs

We count group scalar multiplications (i.e., exponentiations / EC scalar mults) in $\mathbb{G}$, which dominate running time; group additions, hashing, and field operations are lower order unless stated. We analyze the construction in Section 6 for threshold t, m real parties, and $n = m + (t-1)$ total sharing instances, writing $f<t$ for the colluding coalition size from the traceability model.

Table 2 presents closed-form costs and their asymptotics. We separate tracer and judge work into caller-work (denoted as Trace[pub] and TrVer[pub], respectively) and reconstruction-box internal work (denoted as Trace[R] and TrVer[R], respectively) for two reasons: (i) the reconstruction box R may be deployed as a hardware or isolated software module, and its cost should be isolated; and (ii) the split mirrors the algorithms in Section 6 and aligns with our instrumented counters.

Remark 8 (Inherent Cost of Verifiability-Aware Tracing). The $\mathcal{O}(t^2)$ cost per tracing query is inherent to the verifiability-aware box model, not a limitation of our construction: R performs at least VerSS on each of its $t-f$ input shares at $\mathcal{O}(t)$ cost per share, and an adversarial R may inflate this cost further. The caller-side tracing work (Trace[pub]) remains $\mathcal{O}(m)$.

7.2 Experimental Evaluation

We validate the analytic predictions above against a prototype implementation.

Implementation and Setup We set up our experiments as follows.

Prototype and environment. We implemented TSS-PV over Curve25519 in Python with a Rust FFI backend (`curve25519-dalek` v4.1.3, AVX2). Experiments ran on an Intel Core i7–8700K at 4.4 GHz (6C/12T) with 32 GB DDR4–2400, Ubuntu 22.04 (kernel 5.15) under WSL2, with the process pinned to one

Table 2: Analytic costs for all phases of TSS-PV. Asymptotic bounds for Trace[R] and TrVer[R] assume f proportional to t.

Phase	**Scalar Mults in $\mathbb{G}$**	**Asymptotic**
Share $\mathcal{P}_i$ (per party)	$5(t-1)$	$\mathcal{O}(t)$
Share: $\mathcal{D}$ (total)	$5nt + m + 5$	$O(nt)$
Rec (on t shares)	t	$\mathcal{O}(t)$
VerSD	$7n$	$\mathcal{O}(n)$
VerSS (per submission)	$t+6$	$\mathcal{O}(t)$
VerRS (per call)	4	$\mathcal{O}(1)$
Trace[R]	$m(t+6)(t-f) + (m-f)t$	$\mathcal{O}(mt^2)$
Trace[pub]	$4(m-f)$	$\mathcal{O}(m)$
TrVer[R]	$f(t+6)(t-f)$	$\mathcal{O}(ft^2)$
TrVer[pub]	$n(t+13)$	$\mathcal{O}(nt)$
Transcript size	$(t+1)n + 2$ group elem.[†]	$\mathcal{O}(nt)$

[†]At 32 bytes per Curve25519 element: 27 KB (C_1), 102 KB (C_2), 396 KB (C_3), 1.5 MB (C_4).

core. We use four test configurations:

Configurations	(m, t, f)	$n = m + (t-1)$
C_1	$(32, 17, 11)$	48
C_2	$(64, 33, 22)$	96
C_3	$(128, 65, 43)$	192
C_4	$(256, 129, 85)$	384

We perform one warm-up and then ten measured trials, reporting medians. Wall-clock time is measured with `time.perf_counter()`, scalar multiplications are counted by instrumented wrappers, and peak RSS via `getrusage`.

Reconstruction box. The box R embeds f real shares, validates each submitted share via VerSS, enforces per-index uniqueness in x, and returns Rec on the first t unique valid shares (else $\perp$). Thus one R query incurs $(t+6)(t-f)$ scalar mults for validation and an additional t when reconstruction occurs. The reconstruction box R is implemented using the same codebase and runs on the same hardware as the caller-side code.

Results Table 3 reports medians for latency (ms) and total scalar multiplications across all configurations; peak RSS remains below 28 MiB throughout. Figure 14 plots the public-phase and tracing latencies.

Analytic validation (Q1). Measured scalar-mult counts match the closed-form costs in Table 2 exactly: $5(t-1)$ per party for PC-PoK proof generation; $5nt+m+5$ for the dealer; $7n$ for VerSD; $t+6$ per submission for VerSS; t for Rec;

Table 3: Performance of TSS-PV.

Function	C_1 ms / #mult	C_2 ms / #mult	C_3 ms / #mult	C_4 ms / #mult
Standard Lifecycle				
Share $\mathcal{P}_i^*$	3.1 / 80	6.0 / 160	11.8 / 320	24.5 / 640
Share $\mathcal{D}^\dagger$	159.9 / 4117	604.2 / 15909	2424.9 / 62533	9714.4 / 247941
VerSD	14.0 / 336	27.1 / 672	52.7 / 1344	107.6 / 2688
VerSS‡	17.0 / 391	56.2 / 1287	210.2 / 4615	777.8 / 17415
Rec	1.4 / 17	3.8 / 33	11.7 / 65	42.4 / 129
Accountability Mechanism				
Trace[R]	222.7 / 4773	1422.3 / 28842	10048.0 / 205461	78365.7 / 1542699
Trace[pub]	3.2 / 84	6.6 / 168	13.6 / 340	25.8 / 684
TrVer[R]	66.9 / 1518	419.1 / 9438	2988.2 / 67166	23069.0 / 504900
TrVer[pub]	62.3 / 1440	201.0 / 4416	695.0 / 14976	2537.4 / 54528

*Share $\mathcal{P}_i$ is per-party measurement. †Share $\mathcal{D}$ is n-instance measurement. ‡VerSS is measured for t submissions.

4 for VerRS; and the stated forms for all tracing phases. Latency scales linearly with these counts on our platform.

Cost breakdown and scaling (Q2). In the standard lifecycle, the dealer dominates ($\approx 91\%$ of latency at C_4); public verification remains lightweight (VerSD: 108 ms at n=384; VerRS: sub-millisecond throughout). As m doubles from C_1 to C_4, dealer latency grows by $\approx 4\times$ per step (consistent with $O(nt)$ since both n and t roughly double), while per-party sharing grows by $\approx 2\times$ (consistent with $O(t)$). The dealer's work is trivially parallelizable across the n instances; a multi-threaded implementation would reduce wall-clock time proportionally.

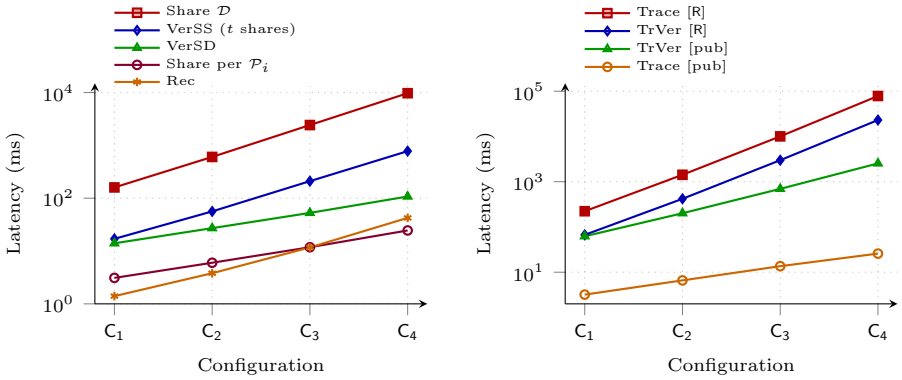
Accountability costs (Q3). Trace[R] accounts for $\approx 97\%$ of total tracing cost at C_4 and exhibits the expected quadratic growth: from 223 ms at C_1 to 78.4 s at C_4 ($\approx 352\times$ as (m, t) quadruples), while caller-side Trace[pub] grows by only $8\times$. Tracing cannot be spam-triggered: no queries to R are issued unless $\mathcal{T}$ is satisfied (Def. 11). For moderate deployments (C_2), the standard lifecycle completes in under 700 ms and the full trace-and-verify cycle in under 2.1 s.

Comparability to prior work. No prior scheme simultaneously provides traceability against verifiability-aware black-box leakage and public verifiability, so a direct end-to-end comparison is impossible. Table 4 contrasts TSS-PV with the lattice-based PVSS of Gentry *et al.* [25] on group-exponentiation cost, omitting field arithmetic, LWE operations, and communication. By avoiding share encryption and heavyweight bulletproofs, our construction trades their $\mathcal{O}(\lambda{+}n)$ sharing cost for $\mathcal{O}(nt)$, while achieving λ-independent, $\mathcal{O}(n)$ distribution verification. The trade-off favors TSS-PV when $t \ll \lambda$, which is the common regime for institutional secret sharing.

Table 4: Asymptotic group-exponentiation cost comparison.

Scheme	Share	Dist. Verif.	Subm. Verif.	Rec	Rec. Verif.
[25]	$O(\lambda{+}n)$[a]	$O(\lambda{+}n)$	—[b]	$O(1)$[c]	—
TSS-PV	$O(nt)$[d]	$O(n)$	$O(t)$	$O(t)$	$O(1)$

[a]TSS-PV counts are λ-independent; [25] verifiers additionally pay $O(\lambda^2{+}\lambda n)$ field mults. [b]Implicit in global Bulletproof. [c]Dominated by LWE; group work $O(1)$. [d]Dealer cost; parties also pay $O(t)$.



(a) Standard Lifecycle

(b) Accountability

Fig. 14: Performance of TSS-PV across configurations C_1–C_4. (a) Standard lifecycle: dealer costs dominate due to PC-PoK proof verification, while public verification remains lightweight. (b) Accountability mechanism: reconstruction-box queries dominate, while caller-side checks remain modest.

Remark 9 (AAC Overhead). Under a metadata-private messaging instantiation [22], delivering $n \leq 384$ messages of $\sim$100 bytes per sharing phase incurs sub-millisecond latency, negligible compared to the cryptographic operations in Table 3.

8 Discussion

We discuss the trade-offs inherent to our scheme and future work.

Box Persistence. Although related works [10,29] assume the reconstruction box to be an oracle without explicit justification, we explicitly discuss its practical grounding. Our traceability analysis models R as a memoryless oracle, assuming the tracer can reset its state between queries. Our construction's indistinguishable dummy shares prevent selective query filtering, any evasion degrades service for all users, and the resulting economic friction (significant counterparty risk

when transmitting shares to anonymous endpoints [1,30]) incentivizes adversaries to distribute local artifacts (e.g., binaries or dongles) subject to forensic capture. Once captured, achieving truly tamper-proof (uncloneable and unresettable) hardware remains an open challenge despite TEE anti-rollback features and PUF-based key storage, as demonstrated by recent attacks [35,37,17].

Tracing Liveness. Tracing does not require universal participation: the dealer holds all dummy shares, and each real party independently proves innocence by submitting its share for testing against R. Parties who submit and pass are cryptographically cleared; parties who refuse or are absent remain in the suspect set. While we cannot cryptographically compel participation, application-layer policies can treat non-cooperation during a forensic audit as a policy violation, making voluntary absence operationally costly.

Future Work. Several directions remain to strengthen guarantees and bridge theory with deployment. First, while Appendix A extends our scheme to pointwise ϵ-good boxes via query amplification, honest dealer and memoryless-oracle model remains an open direction. Second, Blockchain-based time-locked commitments with staked collateral could penalize non-participating parties, transforming forensic obstruction into economic loss. Third, extending TSS-PV to post-quantum assumptions requires new techniques for publicly verifiable dummy-share indistinguishability without discrete-logarithm-based commitments.

References

1. Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? a rational protocol design treatment of bitcoin. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 34–65. Springer, 2018.
2. Karim Baghery, Ehsan Ebrahimi, Omid Mirzamohammadi, and Mahdi Sedaghat. Traceable verifiable secret sharing and applications. *Cryptology ePrint Archive*, 2025.
3. Ali Bagherzandi, Jung-Hee Cheon, and Stanislaw Jarecki. Multisignatures secure under the discrete logarithm assumption and a generalized forking lemma. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 449–458, 2008.
4. Fabio Banfi and Ueli Maurer. Anonymous authenticated communication. In *International Conference on Security and Cryptography for Networks*, pages 289–312. Springer, 2022.
5. Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 390–399, 2006.
6. Allison Bishop, Matthew Green, Yuval Ishai, Abhishek Jain, and Paul Lou. Fully anonymous secret sharing. In *Annual International Cryptology Conference*, pages 356–389. Springer, 2025.
7. George Robert Blakley. Safeguarding cryptographic keys. In *Managing requirements knowledge, international workshop on*, pages 313–313. IEEE Computer Society, 1979.

8. Carlo Blundo and Douglas R Stinson. Anonymous secret sharing schemes. *Discrete Applied Mathematics*, 77(1):13–28, 1997.
9. Dan Boneh. The decision diffie-hellman problem. In *International algorithmic number theory symposium*, pages 48–63. Springer, 1998.
10. Dan Boneh, Aditi Partap, and Lior Rotem. Traceable secret sharing: Strong security and efficient constructions. In *Annual International Cryptology Conference*, pages 221–256. Springer, 2024.
11. Dan Boneh, Aditi Partap, and Lior Rotem. Traceable verifiable random functions. In *Annual International Cryptology Conference*, pages 203–234. Springer, 2025.
12. Jan Camenisch and Markus Stadler. Proof systems for general statements about discrete logarithms. *Technical Report/ETH Zurich, Department of Computer Science*, 260, 1997.
13. Ignacio Cascudo and Bernardo David. Publicly verifiable secret sharing over class groups and applications to dkg and yoso. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 216–248. Springer, 2024.
14. Hervé Chabanne, Duong Hieu Phan, and David Pointcheval. Public traceability in traitor tracing schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 542–558. Springer, 2005.
15. David Chaum and Torben Pryds Pedersen. Wallet databases with observers. In *Annual international cryptology conference*, pages 89–105. Springer, 1992.
16. Yilei Chen, Vinod Vaikuntanathan, Brent Waters, Hoeteck Wee, and Daniel Wichs. Traitor-tracing from lwe made simple and attribute-based. In *Theory of Cryptography Conference*, pages 341–369. Springer, 2018.
17. J Chuang, A Seto, N Berrios, S van Sbochaik, C Garman, and D Genkin. Tee. fail: Breaking trusted execution environments via ddr5 memory bus interposition. In *2026 IEEE Symposium on Security and Privacy (SP). Los Alamitos, CA, USA: IEEE Computer Society*, pages 1894–1912, 2026.
18. Xuan Thanh Do, Duong Hieu Phan, and David Pointcheval. Traceable inner product functional encryption. In *Cryptographers' Track at the RSA Conference*, pages 564–585. Springer, 2020.
19. Oriol Farràs and Miquel Guiot. Traceable secret sharing schemes for general access structures. *Cryptology ePrint Archive*, 2025.
20. Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 427–438. IEEE, 1987.
21. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer, 1986.
22. Kyle Fredrickson, Ioannis Demertzis, James P Hughes, and Darrell DE Long. Sparta: Practical anonymity with long-term resistance to traffic analysis. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1457–1473. IEEE, 2025.
23. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In *Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part II 38*, pages 33–62. Springer, 2018.
24. Sanjam Garg, Abishek Kumarasubramanian, Amit Sahai, and Brent Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In *Proceedings of the 17th ACM conference on computer and communications security*, pages 121–130, 2010.

25. Craig Gentry, Shai Halevi, and Vadim Lyubashevsky. Practical non-interactive publicly verifiable secret sharing with thousands of parties. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 458–487. Springer, 2022.
26. Junqing Gong, Ji Luo, and Hoeteck Wee. Traitor tracing with n 1/3-size ciphertexts and o (1)-size keys from k-lin. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 637–668. Springer, 2023.
27. Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pages 660–670, 2018.
28. Vipul Goyal, Abhishek Jain, and Aditi Partap. Traceable secret sharing revisited. *Cryptology ePrint Archive*, 2025.
29. Vipul Goyal, Yifan Song, and Akshayaram Srinivasan. Traceable secret sharing and applications. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*, pages 718–747. Springer, 2021.
30. Joseph Y Halpern and Rafael Pass. Game theory with translucent players. *International Journal of Game Theory*, 47(3):949–976, 2018.
31. Wataru Kishimoto, Koji Okada, Kaoru Kurosawa, and Wakaha Ogata. On the bound for anonymous secret sharing schemes. *Discrete Applied Mathematics*, 121(1-3):193–202, 2002.
32. Pham Nhat Minh, Khoa Nguyen, Willy Susilo, and Khuong Nguyen-An. Publicly verifiable secret sharing: Generic constructions and lattice-based instantiations in the standard model. *arXiv preprint arXiv:2504.14381*, 2025.
33. Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*, pages 129–140. Springer, 1991.
34. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.
35. Alex Seto, Oytun Kuday Duran, Samy Amer, Jalen Chuang, Stephan van Schaik, Daniel Genkin, and Christina Garman. Wiretap: Breaking server sgx via dram bus interposition. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pages 708–722, 2025.
36. Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
37. Supraja Sridhara, Andrin Bertschi, Benedict Schlüter, and Shweta Shinde. Sigy: Breaking intel sgx enclaves with malicious exceptions & signals. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, pages 1643–1658, 2025.
38. Markus Stadler. Publicly verifiable secret sharing. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 190–199. Springer, 1996.
39. Douglas R Stinson and Scott A Vanstone. A combinatorial approach to threshold schemes. *SIAM Journal on Discrete Mathematics*, 1(2):230–236, 1988.
40. Mark Zhandry. New techniques for traitor tracing: Size and more from pairings. In *Annual international cryptology conference*, pages 652–682. Springer, 2020.

A Pointwise ϵ-Good Reconstruction Boxes

Our main results assume a perfect reconstruction box (Def. 10) that succeeds on every valid input with probability 1. In practice, a box may fail on valid inputs due to network timeouts, rate limiting, or a deliberate adversarial strategy to hinder tracing. This appendix shows that $\mathsf{TSS\text{-}PV}$ extends to *pointwise* ϵ-good boxes via query amplification, at the cost of $O(\lambda/\epsilon)$ overhead per index and a necessary relaxation of the non-imputability guarantee.

A.1 Definition

We relax Def. 10 by replacing the deterministic valid-input clause with a probabilistic one.

Definition 14 (Verifiability-Aware Pointwise ϵ-Good Reconstruction Box). *Fix (T, Σ) with $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$. An oracle R is a* pointwise ϵ-good reconstruction box *for coalition $\mathcal{C} \subseteq [n]$ with leakage $f = |\mathcal{C}|$ and quality parameter $\epsilon \in (0, 1]$ if its internal state contains exactly the shares $\mathsf{E} = \{\mathsf{sh}_i : i \in \mathcal{C}\} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$ and the following holds for all $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$:*

- ***(Valid inputs)*** *If $|\mathsf{SH}| = t - f$ and $\mathsf{SH} \cap \mathsf{E} = \emptyset$, then*

$$\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1] \geq \epsilon.$$

- ***(Invalid inputs)*** *Otherwise,*

$$\Pr[\mathsf{R}(\mathsf{SH}) = \perp \ \vee \ \mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 0] = 1.$$

The perfect reconstruction box (Def. 10) is the special case $\epsilon = 1$. The probability is over the internal randomness of R. The valid-input clause requires $\Pr[\mathsf{VerRS}(\mathsf{R}(\mathsf{SH}), \mathsf{T}) = 1] \geq \epsilon$ for every *valid input; the consequences of this universal quantifier for non-imputability are discussed in Section A.5.*

Remark 10 (Relation to Boneh et al.'s Universally-Good Boxes). Boneh et al. [10] define a *universally-good* reconstruction box (Def. 7 of [10]) by requiring

$$\Pr[\mathsf{R}(\mathsf{sh}'_1, \ldots, \mathsf{sh}'_{t-f}) = \mathsf{Rec}(\mathsf{sh}_1, \ldots, \mathsf{sh}_f, \mathsf{sh}'_1, \ldots, \mathsf{sh}'_{t-f})] \geq \epsilon,$$

where the probability is over shares drawn from a conditional sharing distribution and R's coins. Def. 14 differs in three ways.

First, the success criterion changes from output agreement with Rec to verification by VerRS. Since Rec returns a group element on every input regardless of validity, while a verifiability-aware R returns $\perp$ on inputs whose shares lack provenance in T, the equality $\mathsf{R} = \mathsf{Rec}$ fails by design on unprovenienced inputs, making the universally-good criterion structurally inapplicable. This forces the two-clause formulation: deterministic rejection on invalid inputs (without which tracing cannot distinguish leakers from unlucky honest parties), and a success guarantee on valid ones.

Second, the success guarantee is *pointwise* (per valid input) rather than averaged over a random share distribution. This strengthening is necessary because TSS-PV tests each index i by querying $\mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})$ on a fixed input; under an average-case guarantee, an adversary can set the per-input success probability to zero for a targeted share while maintaining $\mathbb{E}_x[p(x)] \geq \epsilon$, defeating amplification. Boneh et al.'s tracing avoids this problem by querying R with random field-element pairs and extracting the leaker polynomial via list decoding (Fig. 4 of [10]), without testing any specific party's share.

Whether per-index tracing can operate under the weaker average-case notion remains open.

Assumption A1 (Independent Resettable Box Access) *Every oracle call to* R *made by the trigger, tracer, or judge is answered using fresh internal randomness, independent of all other calls. Formally, each call* $\mathsf{R}(\mathsf{SH})$ *is an independent execution of the same randomized algorithm with embedded set* E *and fresh coins. This models a captured artifact amenable to VM snapshotting, physical reset, or binary cloning. A stateful box that detects repeated queries and suppresses reconstruction falls outside this model.*

A.2 Statistical Tracing Trigger

In the perfect-box setting, the trigger $\mathcal{T}$ (Def. 11) fires from a single successful query. Under a pointwise ϵ-good box, repeated testing is needed to gain statistical confidence.

Definition 15 (Statistical Tracing Trigger). *Let* $\epsilon_0 \in (0, 1]$ *be a known lower bound on the box quality (i.e.,* $\epsilon \geq \epsilon_0$*) and let* $K_{\mathsf{trig}} = \lceil \lambda \ln 2/\epsilon_0 \rceil$*. The statistical trigger* $\mathcal{T}^{\epsilon_0}_{\mathsf{stat}}(\mathsf{T}, \Sigma, \mathsf{R}; \mathcal{C})$ *fires if there exists* $\mathsf{SH} \subseteq \mathcal{V}(\mathsf{T}, \Sigma)$ *with* $|\mathsf{SH}| = t-f$ *and* $\mathsf{SH} \cap \{\mathsf{sh}_i : i \in \mathcal{C}\} = \emptyset$ *such that at least one of* K_{trig} *independent queries returns a valid reconstruction:*

$$\sum_{k=1}^{K_{\mathsf{trig}}} \mathbf{1}\big[\mathsf{VerRS}(\mathsf{R}^{(k)}(\mathsf{SH}), \mathsf{T}) = 1\big] \geq 1.$$

Under Assumption A1, the false-negative probability is $(1-\epsilon_0)^{K_{\mathsf{trig}}} \leq 2^{-\lambda}$*; the bound follows from* $1-\epsilon_0 \leq e^{-\epsilon_0}$*, giving* $(1-\epsilon_0)^{K_{\mathsf{trig}}} \leq e^{-\epsilon_0 \cdot K_{\mathsf{trig}}} \leq e^{-\lambda \ln 2} = 2^{-\lambda}$*.*

Remark 11 (Operational Realization and Leakage-Size Discovery). Def. 15 is a definitional event where the coalition $\mathcal{C}$ and leakage size f are known to the experiment. In the concrete protocol, the dealer discovers f by probing R with subsets of its $t-1$ dummy shares at candidate sizes $f' \in \{1, \ldots, t-1\}$, sending $t-f'$ shares per probe. Only the candidate $f' = f$ satisfies the valid-input clause of Def. 14 (correct cardinality and disjoint from E); all others fail deterministically under the invalid-input clause. The dealer runs K_{trig} trials at the correct candidate to confirm functionality and eliminates wrong candidates after a single deterministic failure. The total trigger cost is $O(t + \lambda/\epsilon_0)$ queries.

A.3 Amplified Tracing and Security

By the same technique as Def. 15, but with a union bound over at most n indices, the per-index failure must be $\leq 2^{-(\lambda+\log_2 n)}$ to achieve overall failure $2^{-\lambda}$. This gives $K_{\mathsf{amp}} = \lceil(\lambda+\log_2 n)\ln 2/\epsilon_0\rceil$. The amplified algorithms $\mathsf{Trace}^{\mathsf{R},\epsilon_0}$ and $\mathsf{TrVer}^{\mathsf{R},\epsilon_0}$ replace each single-query step in Figure 13 with a K_{amp}-repetition loop: for each real index $i \in \mathcal{I}$, the tracer (resp. judge) queries $\mathsf{R}(\mathsf{DSH} \cup \{\mathsf{sh}_i\})$ up to K_{amp} times, where sh_i is the share in ptk_i (resp. pvk_i). If any query yields $\mathsf{VerRS}(S', \mathsf{T}) = 1$, index i is *cleared*; otherwise i remains accused.

Theorem 1.11 (Traceability under Pointwise ϵ-Good Boxes). *Let* R *be a pointwise* ϵ*-good reconstruction box (Def. 14) with* $\epsilon \geq \epsilon_0$ *satisfying Assumption A1, and let* $\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$*. Assume the leakage size* $f = |\mathcal{C}^*|$ *is known to the tracer and judge (e.g., via Remark 11). Then* TSS-PV *with amplification* $K_{\mathsf{amp}} = \lceil(\lambda + \log_2 n)\ln 2/\epsilon_0\rceil$ *satisfies traceability:*

$$\Pr\left[\mathcal{C}' = \mathcal{C}^* \ \wedge \ \mathsf{TrVer}^{\mathsf{R},\epsilon_0}(\mathsf{VK}, \mathsf{T}, \mathcal{C}', \pi_{\mathsf{trace}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda),$$

where $(\mathcal{C}', \pi_{\mathsf{trace}}) \leftarrow \mathsf{Trace}^{\mathsf{R},\epsilon_0}(\mathsf{TK}, \mathsf{T})$.

Proof. Let $\mathcal{C}^* \subseteq [n]$ be the true leaker set embedded in R.

Completeness (leakers remain accused). For $i \in \mathcal{C}^*$, every query $\mathsf{DSH} \cup \{\mathsf{sh}_i\}$ contains a leaked share $\mathsf{sh}_i \in \mathsf{E}$. By the invalid-input clause of Def. 14, every such query fails deterministically; hence i is never cleared.

Soundness (innocent parties are cleared). For $i \notin \mathcal{C}^*$, the query $\mathsf{DSH} \cup \{\mathsf{sh}_i\}$ is valid: $|\mathsf{DSH} \cup \{\mathsf{sh}_i\}| = t - f$ (since $|\mathsf{DSH}| = t-f-1$ and $\mathsf{sh}_i \notin \mathsf{DSH}$ by construction), $\mathsf{sh}_i \notin \mathsf{E}$ (since $i \notin \mathcal{C}^*$), and $\mathsf{DSH} \cap \mathsf{E} = \emptyset$ (dummy shares are disjoint from E). By the valid-input clause and Assumption A1, each of the K_{amp} trials succeeds independently with probability $\geq \epsilon_0$. The probability that all trials fail is $(1-\epsilon_0)^{K_{\mathsf{amp}}} \leq 2^{-(\lambda+\log_2 n)}$. A union bound over at most $n-f$ innocent parties gives failure probability at most $(n-f)\cdot 2^{-(\lambda+\log_2 n)} \leq 2^{-\lambda}$.

Judge acceptance. The judge runs the same K_{amp}-repetition test on each accused index. Since $\mathcal{C}' = \mathcal{C}^*$ except with probability $2^{-\lambda}$, every accused index has $\mathsf{sh}_i \in \mathsf{E}$; the judge's queries fail deterministically and the judge accepts. The overall failure probability is $\mathsf{negl}(\lambda)$.

Theorem 1.12 (Non-Imputability under Pointwise ϵ-Good Boxes). *Under honest share distribution* ($\Phi_{\mathsf{dist}}(\mathsf{T}, \Sigma) = 1$)*, party secrecy (Theorem 6.8), and Assumption A1,* TSS-PV *with amplification* $K_{\mathsf{amp}} = \lceil(\lambda + \log_2 n)\ln 2/\epsilon_0\rceil$ *satisfies:*

$$\Pr\left[\mathbf{ExpNI}^{\epsilon_0}_{\mathcal{A},\mathsf{TSS\text{-}PV}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

where $\mathbf{ExpNI}^{\epsilon_0}$ *is the experiment of Fig. 8 modified as follows: (i)* $\mathcal{A}$ *follows* Share *honestly (but controls the coalition and constructs* R*); (ii)* R *is a pointwise* ϵ*-good box with* $\epsilon \geq \epsilon_0$ *satisfying Assumption A1; (iii)* TrVer *is replaced by* $\mathsf{TrVer}^{\mathsf{R},\epsilon_0}$*.*

Proof. Let $i^* \in \mathcal{I}$ be the honest challenge party. Since the experiment restricts the adversary to honest dealing and i^* is not in the controlled coalition $\mathcal{C}$, a box satisfying Def. 14 for coalition $\mathcal{C}$ has embedded set $\mathsf{E} = \{\mathsf{sh}_i : i \in \mathcal{C}\}$ and therefore $\mathsf{sh}_{i^*} \notin \mathsf{E}$. If the adversary instead constructs a box whose internal state depends on sh_{i^*} (e.g., by learning sh_{i^*} and embedding it), this either violates party secrecy (Theorem 6.8), which occurs with probability $\epsilon_{\mathrm{psec}}(\lambda) = \mathsf{negl}(\lambda)$, or violates Def. 14.

Conditioned on $\mathsf{sh}_{i^*} \notin \mathsf{E}$, the query $\mathsf{DSH} \cup \{\mathsf{sh}_{i^*}\}$ is valid. By the valid-input clause and Assumption A1, each of the judge's K_{amp} trials succeeds independently with probability $\geq \epsilon_0$. For i^* to be falsely convicted, all trials must fail:

$$\Pr[i^* \text{ convicted} \mid \mathsf{sh}_{i^*} \notin \mathsf{E}] \leq (1 - \epsilon_0)^{K_{\mathsf{amp}}} \leq 2^{-\lambda}.$$

By total probability: $\Pr[\mathbf{ExpNI}^{\epsilon_0} = 1] \leq \epsilon_{\mathrm{psec}}(\lambda) + 2^{-\lambda} = \mathsf{negl}(\lambda)$.

A.4 Overhead

Table 5 summarizes the query overhead introduced by amplification.

Table 5: Query overhead: perfect vs. pointwise ϵ-good boxes ($\epsilon \geq \epsilon_0$).

Operation	**Perfect ($\epsilon = 1$)**	**Pointwise ϵ-Good**
Trigger testing	1 query	$O(t + \lambda/\epsilon_0)$ queries
Tracing (total)	m queries	$O(m(\lambda + \log n)/\epsilon_0)$ queries
TrVer (per accused)	1 query	$O((\lambda + \log n)/\epsilon_0)$ queries

For $\lambda = 128$, $n = 384$, and $\epsilon_0 = 0.9$, the amplification factor is $K_{\mathsf{amp}} = \lceil (128 + \lceil \log_2 384 \rceil) \cdot \ln 2/0.9 \rceil = 106$. At configuration $\mathsf{C_4}$, the perfect-box tracing latency is $\approx 78\,\mathrm{s}$ (Table 3), yielding an amplified cost of $\approx 106 \times 78 \approx 2.3$ hours, acceptable for forensic operations that occur only upon confirmed leakage.

A.5 Trade-offs and Limitations

We identify four limitations inherent to the amplification approach.

Non-imputability requires honest dealing. In the perfect-box setting, Theorem 6.10 achieves non-imputability against a fully malicious dealer: deterministic box behavior prevents input-dependent failures from being engineered. Under a probabilistic box, a malicious dealer could construct R to fail deterministically on queries containing a targeted honest party's share while succeeding with probability ϵ on all others. Such an R violates the pointwise guarantee of Def. 14 (which requires success probability $\geq \epsilon$ for *every* valid input), but the amplification mechanism observes only accept/reject outcomes and cannot distinguish this violation from legitimate random failure. Theorem 1.12 therefore restricts to

honest dealing. This is operationally justified: tracing is invoked only after Φ_{dist} is verified, and a malicious dealer can frame parties more directly via inconsistent shares, which VerSD detects.

Resettability is load-bearing. The amplified protocol submits the same input K_{amp} times per index. A stateful box can detect this repetition and suppress reconstruction, defeating amplification. Assumption A1 is therefore necessary, restricting the extension to captured artifacts amenable to VM snapshotting or physical reset.

Non-negligible ϵ is required. The factor K_{amp} is polynomial only when ϵ_0 is non-negligible. If $\epsilon = \mathsf{negl}(\lambda)$, then K_{amp} becomes super-polynomial and tracing is infeasible. This is not a practical concern: a box with negligible success probability fails on almost all buyer queries, rendering it commercially worthless, but this remains an informal market argument rather than a cryptographic guarantee.

Cubic tracing cost. Each R query costs $O(t^2)$ scalar multiplications (Table 2), so the amplified tracing cost is $O(mt^2 \cdot (\lambda + \log n)/\epsilon_0)$. At $\mathsf{C_4}$ ($m = 256$, $t = 129$), this corresponds to ≈ 2.3 hours versus ≈ 78 s in the perfect-box setting (Table 3). While acceptable for rare forensic events, this cost may become prohibitive for larger deployments or lower ϵ_0 values.