Pilaf: Fully Tight Two-Round Threshold Signatures with Adaptive Corruptions

Chen Qian, Xingyu Zhao, Hao Cheng, Zengpeng Li, Puwen Wei, and Quan Yuan

[1] School of Cyber Science and Technology, Shandong University, Qingdao, Shandong, 266237, China
[2] State Key Laboratory of Cryptography and Digital Economy Security, Shandong University, Jinan, 250100, China
[3] Key Laboratory of Cryptologic Technology and Information Security of Ministry of Education, Shandong University, Jinan, Shandong, 250100, China

Abstract. Threshold signatures are deployed in settings where an adversary may run many concurrent signing sessions and corrupt signers adaptively. Two-round schemes make this especially delicate. Their first-round messages are independent of the signed message and can be preprocessed offline, so a later corruption must reveal randomness that is consistent with commitments already published in prior transcripts. Existing adaptive constructions address this tension by adding rounds, relying on algebraic or knowledge assumptions, or paying non-tight losses from guessing the corruption pattern, the decisive session, or the final transcript.

We construct Pilaf, the first two-round threshold signature scheme that combines partially non-interactive signing with a fully tight proof against adaptive corruptions. The scheme is pairing-free and is built in prime-order groups from the MDDH assumption in the random-oracle model. Its first-round messages can be generated offline, and any threshold set of signers can aggregate their second-round shares into a single publicly verifiable signature.

The proof combines two ingredients. First, we introduce a linearly homomorphic dual-mode commitment with targetable opening. This lets the simulator open an already fixed commitment to the aggregate target imposed by a later Fiat-Shamir challenge. Second, we use profile-wise zero-sum masking with posterior completion. Corruption openings and signing responses are therefore sampled from the exact conditional law while values already visible to the adversary remain cached. Together, these tools enable a delayed branch-decision argument. The reduction waits until the adversary's own queries determine the last touched coordinate, completes only latent state, and then binds the forged hidden branch. The final bound has no combinatorial loss in the number of users, threshold, sessions, or corruption patterns, and contains only the explicit bad-event and assumption terms appearing in the theorem.

1 Introduction

A threshold signature scheme [Des90] enables a set of users to sign under a single public key in such a way that any subset of at least $\mathtt{t}$ users can produce a valid signature, whereas any subset of fewer than $\mathtt{t}$ users cannot. Threshold signatures are a fundamental primitive in distributed cryptography: they retain the compact verification interface of ordinary signatures while spreading trust across multiple parties. Consequently, they have become a standard tool in distributed custody, secure multi-party computation, and blockchain systems.

Security Notions for Threshold Signatures. At a high level, unforgeability should hold against an adversary that can interleave many signing sessions, choose messages and signer sets adaptively, and corrupt users to obtain both long-term secret shares and local signing randomness. Bellare et al. [BCK+22] formalized the static-corruption landscape via the hierarchy $\mathsf{TS\text{-}UF}\text{-}\{0,1,2,3,4\}$. Roughly speaking, $\mathsf{TS\text{-}UF}\text{-}\{0,1,2\}$ refine the trivial-win condition by specifying when a forgery on an already signed message should still count, whereas $\mathsf{TS\text{-}UF}\text{-}\{3,4\}$ additionally model attacks with a malicious leader. An important lesson from their framework is that the generic implication from $\mathsf{TS\text{-}UF}\text{-}0$ to $\mathsf{TS\text{-}UF}\text{-}1$ already incurs the combinatorial loss $\binom{\mathsf{N}}{\mathtt{t}-1}$, which is exponential in the threshold parameter. Thus, even in the static setting, direct proofs for stronger notions are preferable whenever concrete security matters. Navot et al. [NT24] later proposed one-more unforgeability, but, as they note, this notion is weaker than $\mathsf{TS\text{-}UF}\text{-}2$. Altogether, the modern viewpoint is that threshold-signature security must account not only for basic unforgeability, but also for transcript reuse, concurrency, corruption, and reduction loss.

Adaptive Corruption. Adaptive corruption makes this problem substantially harder, especially in two-round protocols. In many efficient constructions, the first-round message is independent of the message and can therefore be preprocessed offline. A later corruption then reveals not only the user's long-term secret share, but also the precise randomness underlying previously published first-round messages. A successful simulator must therefore explain both the public transcript and the local state revealed after corruption. This is exactly where adaptive security becomes delicate: hiding the public commitment is not enough if the revealed opening has the wrong conditional distribution. Moreover, across many signing sessions, the adversary may accumulate leakage about the same long-term share, so an adaptively secure proof must also preserve cross-session consistency.

Concrete Security and Tightness. Beyond asymptotic feasibility, concrete security is crucial for threshold signatures. These schemes are intended for deployments with many users, many signing sessions, and often many registered instances. In such settings, even a polynomial loss in the numbers of signing

queries, corruption queries, or random-oracle queries can substantially weaken the final parameters. Threshold signatures face an additional obstacle: selective-style arguments often guess the eventual corruption pattern or surviving set, which introduces combinatorial losses such as $\binom{N}{t-1}$. Eliminating these losses is therefore not merely a matter of proof elegance; it is necessary to obtain meaningful end-to-end security guarantees.

Current Landscape. Recent work has substantially advanced the area, but the central goal remains unresolved. Within the original TS-UF hierarchy of Bellare et al. [BCK+22], threshold-RSA and threshold-BLS instantiate TS-UF-1, FROST2 gives a representative TS-UF-2 protocol, and FROST reaches TS-UF-3 in the static-corruption setting. The FROST line and its descendants further emphasize practicality and partial non-interactivity, while fully adaptive analyses in that direction typically rely on AGM-style reasoning or incur non-tight reductions [KG20,CKM23,BLSW24,BGC+25]. A different line of work, based on standard assumptions such as DDH and DLOG, has obtained adaptive security without pairings [BLT+24,DR24,Che25a,Che26] and, in some cases, full tightness [BW24,TJZ25,BC26]. However, as summarized in table 1, no previous construction achieves full adaptive-corruption security at the TS-UF-2 level together with a fully tight reduction. This is the main target of our work. As we show, this target can in fact be met while simultaneously obtaining several additional desirable properties, including two rounds, partial non-interactivity, and security from standard assumptions without additional algebraic or interactive assumptions. This leaves the following natural question.

> **Can we build a threshold signature scheme that achieves full adaptive security with a fully tight reduction?**

2 Our Contributions

We answer the above question positively. We construct Pilaf, the *first* two-round threshold signature scheme that achieves full adaptive-corruption security with a fully tight reduction. More specifically, in addition to full tightness and adaptive security, Pilaf offers the following properties:

Partially Non-Interactive: The first-round message is message-independent and can be preprocessed offline.

Standard Assumption: The security of our scheme relies on the MDDH assumption in the random-oracle model, without any additional algebraic, interactive, or knowledge assumptions.

Pairing-Free: The scheme does not require pairings and can be instantiated in any group of prime order assuming the MDDH problem is hard.

Fully Tight for Multiple Instances: The scheme remains fully tight even when multiple instances are used.

The adaptive proof requires a stronger commitment layer than the usual Schnorr-style trapdoor interface. That interface lets a simulator open a fixed commitment to targets of the form $\mathsf{F}(\rho) + cX$. This is too restrictive for our threshold proof, where the hidden branch split may later require opening the same committed object either to $\mathsf{F}(x_b)$ or to its complementary value $X - \mathsf{F}(x_b)$. Equivalently, after the commitment has been fixed, the simulator must be able to target any selected linear combination $c\mathsf{F}(1) + dX$. We therefore introduce a homomorphic dual-mode commitment with targetable opening for this larger target family, while preserving the binding and linear aggregation structure needed by a two-round threshold-signature protocol.

Together with this commitment layer, we develop a delayed branch-decision technique based on trapdoor commitments and masking. The masking layer keeps subthreshold first-round views rerandomized, while the homomorphic trapdoor commitments let the proof open the aggregate branch consistently only when the adversary's query pattern has forced the relevant transcript. In this way, the branch decision is not fixed by any single party or by an early simulator guess. It is postponed until the last useful moment and is then realized at the aggregate level.

Scheme	**Assump.**	**Rounds**	**PNI**	**Full adaptive**	**Fully tight**
FROST [KG20]	OMDL	2	✓	✗	✗
Sparkle+ [CKM23]	AOMDL + AGM	3	✓	✓	✗
Twinkle [BLT+24]	DDH	3	✗	✓	✗
Das-Ren BLS [DR24]	DDH + co-CDH	1	✗	✓	✗
HARTS [BLSW24]	AOMDL + AGM	2	✓	✓	✗
Twinkle-T [BW24]	DDH	3	✗	✓	✓
Dazzle [Che25a]	DDH	2	✗	✓	✗
FA-FROST [BGC+25]	AOMDL + AGM	2	✓	✓	✗
Glacius [BDLR25b]	DDH	5	✗	✓	✗
Gargos [BDLR25a]	DDH	3	✗	✓	✗
Glitter [TJZ25]	DDH	5	✗	✓	✓
Niot [NRT25]	DLOG	3	✓	✓	✗
Chen [Che26]	DLOG	2	✓	✓	✗
Earpicks [BC26]	DDH	2	✗	✗	✓
Pilaf (this work)	DDH	2	✓	✓	✓

Table 1. Comparison of recent threshold-signature schemes most relevant to this work. Here "PNI" means that the first-round message is message-independent and can be preprocessed offline.

3 Technical Overview

This section gives an informal roadmap for the construction and the proof. The main idea is to keep the first round deliberately underdetermined. The public transcript is fixed, but the proof postpones the algebraic choice that will make one branch useful. Commitments provide the ability to explain a fixed string later. Masking keeps the local views below threshold from revealing which explanation is being prepared. Freshness then guarantees that the final forgery is made on an identifier for which this hidden choice has not already been exposed.

3.1 Dual-Mode Commitment with Targetable Opening

The first round is produced before the message and before the Fiat–Shamir challenge. At that time the simulator does not yet know which aggregate branch will have to carry the proof. We therefore need commitments that can be fixed in the transcript now and interpreted later in the way the eventual transcript requires.

The needed targets come from splitting the aggregate public key into two branches. Later, one branch may have to behave as $\mathsf{F}(x_b)$, while the other behaves as the complementary value $X - \mathsf{F}(x_b)$. In a one-dimensional picture, the commitment should open to any chosen target $c\mathsf{F}(1) + dX$. The extra representatives that make this possible are kept only in the proof. They are not part of the protocol transcript.

The concrete commitment has the augmented form

$$\big[\mathbf{C}\big] := \big[\mathbf{R}\big] + \big[\mathbf{A}\big] \cdot \mathbf{U}, \qquad \big[\mathbf{S}\big] := \big[\mathbf{Y}\big] \cdot \mathbf{U} + \big[\mathbf{Z}\big] \cdot \mathbf{V}.$$

The first component fixes the visible committed group element. The second component is the linear handle used for retargeting. Under a simulation key, the trapdoor changes only the hidden opening information behind the same public string. Under a binding key, the same commitment syntax fixes a unique value for the final extraction step.

3.2 Masking as posterior completion

Masking is what makes the branch split collective. Each signer locally has two branch contributions, but those local contributions are never exposed in the clear below threshold. They are masked in the first round, and the corresponding second-round values and openings are masked as well. After aggregation the masks cancel, so correctness is unchanged. Before aggregation, each visible local piece still has fresh randomness around it.

For the proof, this is best viewed as posterior completion. Once a mask value or a corrupted local state has been returned, it is fixed forever. Coordinates that remain hidden are not sampled until the proof needs them. The zero-sum aggregate condition tells the simulator how to complete the remaining coordinates. With more than one hidden coordinate, the next one can still be uniform. With one hidden coordinate left, the aggregate equation fixes it. This is the exact flexibility used at the last-touch point.

3.3 Delayed branch decision

For every signing identifier, the protocol carries two branches. A signer splits its share into $x_{\ell,0}$ and $x_{\ell,1}$ with $x_{\ell,0} + x_{\ell,1} = x_\ell$. After aggregation, the two branch values add to the public key, $X_0 + X_1 = \mathsf{vk}_\Sigma$. Locally, however, the branch split remains hidden behind the masking layer until enough signers have been touched.

The proof uses a hidden bit b_{id} to name the branch that will be tested in a future forgery. The important point is that this bit is not learned before a last-touch query on the identifier. If such a last-touch query happens, the identifier is no longer fresh for the final forgery, because the adversary has now obtained a threshold amount of relevant information for that identifier. Thus, on the final fresh identifier, the branch bit is still hidden at the moment the proof uses it.

3.4 Last-touch completion

Fix a second-round transcript. A signer is touched on that transcript once the adversary has either received the signer's second-round answer or corrupted the signer and obtained the relevant local state. Before the touched set reaches the threshold, the simulator can keep the remaining local coordinates latent. The visible transcript constrains their aggregate, but it does not force a unique completion.

The last-touch query is the moment when this changes. The adversary has now chosen the transcript and has fixed all but one of the relevant local coordinates. The simulator completes the aggregate honest transcript first. The remaining signer receives the unique Lagrange residual needed to make the aggregate equation hold. The targetable commitment explains the latent aggregate R commitment after the challenge h is known, and masking turns the aggregate completion back into local-looking shares. This is why the proof does not guess the decisive signer or transcript. It waits for the adversary to identify them.

3.5 Binding the forged branch

At the end of the proof, the branch that stayed hidden becomes the branch from which we extract. Public keys are replaced by uniform elements of $\mathsf{CD}_{\mathsf{LFF}}$. Such a point is outside the DDH image except with small probability. Since the verifier checks $X_0^\star + X_1^\star = \mathsf{vk}_\Sigma^\star$, a non-image public key means that the two forged branches cannot both be DDH images.

The final identifier is fresh, so its hidden branch bit has not been revealed by last-touch simulation. Testing that hidden branch therefore loses only a constant factor. Once the tested branch is outside the DDH image, the proof switches the relevant commitments to binding keys and extracts fixed values $X_{\mathsf{ext}}^\star$ and $R_{\mathsf{ext}}^\star$ before the final challenge is sampled. A valid forgery would then require

$$R_{\mathsf{ext}}^\star = \mathsf{F}\left(\rho^\star\right) + X_{\mathsf{ext}}^\star \mathsf{h}^\star.$$

Since $X_{\mathsf{ext}}^\star$ is outside the image, lossy soundness says that a fresh random-oracle challenge satisfies this equation only with small probability.

3.6 Putting the proof together

The full proof turns this picture into a sequence of hybrids. It replaces the concrete masking implementation by the ideal posterior-completion view, keeps first-round branch data lazy, uses last-touch completion for adaptive signing queries, and finally switches the fresh forged branch to binding mode. The central invariant is that visible values are never changed. Only latent values are completed, and they are completed from the law conditioned on what the adversary has already seen. This is what removes the usual guessing loss.

4 Preliminaries

4.1 Notations

Throughout this paper, we use the following notations.

Algebraic Spaces. Throughout this paper, let $\mathbb{F}$ be a finite field and let $\mathcal{M}$ be a finite-dimensional $\mathbb{F}$-vector space from which uniform sampling is efficient. We denote by $\mathsf{S} \subseteq \mathbb{F}$ a finite set of distinct nonzero points.

Random Oracles. Let H be a hash function modeled as a random oracle, we denote by $\mathsf{H}(\cdot)$ the random oracle query. We denote by $\mathsf{H}[\cdot]$ the internal bookkeeping table for the random oracle H.

Pseudocode Conventions. In UC-style functionality figures, a line **leak** M means that the functionality sends the leakage message M to the adversary, or to the ideal-world simulator acting on its behalf. The return value on a line **return** v is the output delivered to the invoking party or oracle caller, unless the figure explicitly specifies an adversary-controlled delivery step. Unless a figure explicitly initializes a state entry otherwise, all non-set-valued entries are initialized to $\perp$, and all set-valued entries are initialized to $\emptyset$.

4.2 Lagrange Coefficients

Definition 1 (Lagrange Coefficients). Let S be a finite set of distinct nonzero elements of $\mathbb{F}$. For every $i \in \mathsf{S}$, we define the Lagrange coefficient of i in S at the evaluation point 0 by

$$\lambda_{\mathsf{S},i} := \prod_{j \in \mathsf{S} \setminus \{i\}} \frac{-j}{i-j}.$$

Equivalently, for every polynomial $f \in \mathbb{F}[X]$ of degree strictly smaller than $\#(\mathsf{S})$, we have

$$f(0) = \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} f(i).$$

4.3 Profile-wise Zero-Sum Masking

Syntax and Parsed Data. Fix an efficiently decidable set $\mathcal{T}_\Delta$ of legal mask types. For every $\mathsf{type} \in \mathcal{T}_\Delta$, the set $\mathsf{D}(\mathsf{type}) \subseteq \mathcal{M}$ is a finite-dimensional efficiently samplable $\mathbb{F}$-vector subspace. Queries with $\mathsf{type} \notin \mathcal{T}_\Delta$ fail the type check. For legal types, mask values lie in $\mathsf{D}(\mathsf{type})$.

Throughout this subsection, for every total number of users N under consideration, the labels in $[\mathsf{N}]$ are embedded into $\mathbb{F}$ as distinct nonzero field elements. In particular $\mathsf{N} < \#(\mathbb{F})$, and every Lagrange coefficient $\lambda_{\mathsf{S},i}$ for $\mathsf{S} \subseteq [\mathsf{N}]$ is well defined and nonzero because each embedded label is nonzero and all differences $i - j$ with $i \neq j$ are nonzero in $\mathbb{F}$.

Definition 2 (Syntax). A profile-wise zero-sum masking scheme consists of two PPT algorithms $\Delta = (\mathsf{KGen}, \mathsf{Eval})$ with the following syntax.

- $\mathsf{KGen}(1^\lambda, \mathsf{N}, \mathsf{t}) \to \{\mathsf{s}_i\}_{i\in[\mathsf{N}]}$: Takes the unary representation 1^λ of a security parameter λ, the total number of users N, and the threshold t, and outputs user-local seed tuples $\{\mathsf{s}_i\}_{i\in[\mathsf{N}]}$.
- $\mathsf{Eval}(\mathsf{s}_i, \mathsf{S}, \mathsf{type}, \mathsf{ctx}) \to y$: Takes a user seed tuple s_i, a signer set $\mathsf{S} \subseteq [\mathsf{N}]$, a legal mask type $\mathsf{type} \in \mathcal{T}_\Delta$, and a masking context ctx, and outputs a mask value $y \in \mathsf{D}(\mathsf{type}) \subseteq \mathcal{M}$.

We write $\Delta^{\mathsf{type}}_{\mathsf{S},i}(\mathsf{ctx})$ as shorthand for the evaluation $\mathsf{Eval}(\mathsf{s}_i, \mathsf{S}, \mathsf{type}, \mathsf{ctx})$. When the signer set is clear from context, we abbreviate $\Delta^{\mathsf{type}}_{\mathsf{S},i}(\mathsf{ctx})$ by $\Delta^{\mathsf{type}}_{i}(\mathsf{ctx})$.

Definition 2 specifies only syntax. A concrete masking scheme is zero-sum-correct if, for every legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$, every seed tuple output by KGen, and every valid context ctx, its evaluations satisfy

$$\sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} \Delta^{\mathsf{type}}_{\mathsf{S},i}(\mathsf{ctx}) = 0.$$

Lemma 2 proves this correctness property for the ROM construction Δ_{ROM}.

The masking contexts used in this paper carry the signer set that determines the weighted cancellation relation. We parse this data as follows.

Definition 3 (Parsed data of a masking context). For a mask-oracle context ctx, define the partial functions $\mathsf{sid}(\mathsf{ctx})$ and $\mathsf{S}(\mathsf{ctx})$ as follows. If $\mathsf{ctx} = (\mathsf{id}, \mathsf{S}, b)$ and $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, then

$$\mathsf{sid}(\mathsf{ctx}) := \mathsf{sid} \qquad \text{and} \qquad \mathsf{S}(\mathsf{ctx}) := \mathsf{S}.$$

If $\mathsf{ctx} = (\mathsf{sinf}, b)$, where $\mathsf{sinf} = (\mathsf{vk}_\Sigma, \tau, \mathsf{id}, \mathsf{S}, \mathsf{m})$ and $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, then define the same values $\mathsf{sid}(\mathsf{ctx}) := \mathsf{sid}$ and $\mathsf{S}(\mathsf{ctx}) := \mathsf{S}$.

The two context forms are treated as syntactic patterns. If ctx does not match either pattern, or if the nested identifier id does not parse as $(\mathsf{sid}, \mathsf{rid})$, then both $\mathsf{sid}(\mathsf{ctx})$ and $\mathsf{S}(\mathsf{ctx})$ are undefined. Thus malformed contexts never match a legal session or signer set in the functionality bookkeeping below.

For the UC-style ideal functionality below, a profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$ is legal if $\mathsf{sid}(\mathsf{ctx}) \neq \bot$, $\mathsf{S}(\mathsf{ctx}) = \mathsf{S}$, $\mathsf{type} \in \mathcal{T}_\Delta$, $\mathsf{S} \subseteq [\mathsf{N}]$, and $\#(\mathsf{S}) = \mathsf{t}$. Each legal profile is initialized at most once by Init. Values already returned to the adversary or to a party are cached and are never resampled.

Definition 4 (Profile-wise zero-sum masking ideal functionality). The profile-wise zero-sum masking ideal functionality is the stateful UC functionality $\mathcal{F}_\Delta$ specified in fig. 1. It exposes initialization, mask-opening, and corruption-opening interfaces. For each legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$, coordinates remain latent until they are opened. A latent coordinate opened by Read_π is sampled from the exact conditional completion law determined by the fixed coordinates in $\mathsf{Fix}[\pi, \cdot]$. If fewer than $\mathsf{t} - 1$ profile coordinates are fixed, then this law is uniform over $\mathsf{D}(\mathsf{type})$ for the next opened coordinate. Once a coordinate is fixed, the functionality caches it and never resamples it. The completed profile-wise mask vector satisfies

$$\sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{Fix}[\pi, i] = 0.$$

On corruption, the functionality returns an abstract opening state containing all initialized profile coordinates for the corrupted party. When Init later initializes $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$ with $\mathsf{sid} = \mathsf{sid}(\mathsf{ctx})$ and fixes a coordinate for a party already in $\mathcal{C}[\mathsf{sid}]$, the functionality emits the corresponding $\mathsf{CorMask}$ leakage and caches that coordinate in $\mathsf{OSt}[\mathsf{sid}, i]$.

The zero-sum invariant in definition 4 follows from the lazy completion rule. If fewer than $\mathsf{t}-1$ coordinates of $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$ are fixed, then reading one further latent coordinate samples a uniform value in $\mathsf{D}(\mathsf{type})$ and leaves at least one latent coordinate available to complete the equation. Thus, in a proof coupling, this next coordinate may be identified with an independently uniform target in $\mathsf{D}(\mathsf{type})$. It may not be fixed to an arbitrary deterministic or adversarially chosen value without changing the visible marginal law. If exactly $\mathsf{t} - 1$ coordinates are fixed, Read_π sets the unique remaining coordinate to

$$-\lambda_{\mathsf{S},i}^{-1} \sum_{j \in F_\pi} \lambda_{\mathsf{S},j} \mathsf{Fix}[\pi, j],$$

so the completed vector satisfies $\sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{Fix}[\pi, i] = 0$. The state returned by $\mathcal{S}.\mathsf{CorMask}$ is an abstract opening state. It records exactly the mask coordinates for already-initialized profiles involving the corrupted party. Later calls to Init emit separate $\mathsf{CorMask}$ leakage for already corrupted parties in the new profile, and those leaked coordinates are cached in the same OSt table.

The following affine-fiber fact underlies the lazy completion rule. It also applies componentwise to any other vector space whose coordinates are constrained by a Lagrange-weighted aggregate equation.

Lemma 1 (Affine profile completion). *Let V be a finite $\mathbb{F}$-vector space, let $\mathsf{S} \subseteq [\mathsf{N}]$ satisfy $\#(\mathsf{S}) = \mathsf{t}$, and let all Lagrange coefficients $\lambda_{\mathsf{S},i}$ be nonzero. For*

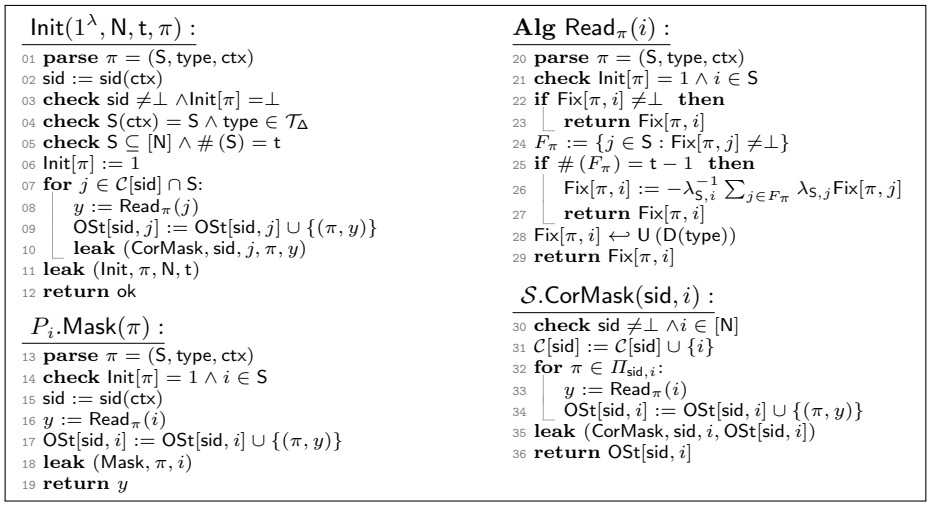


Fig. 1. UC-style ideal functionality $\mathcal{F}_\Delta$ for profile-wise zero-sum masking. The table $\mathsf{Fix}[\pi, i]$ records profile coordinates fixed by visible openings, with unset entries equal to $\perp$. The set-valued entries $\mathcal{C}[\mathsf{sid}]$ and $\mathsf{OSt}[\mathsf{sid}, i]$ default to $\emptyset$ under section 4.1. In $\mathcal{S}$.CorMask, $\Pi_{\mathsf{sid},i} := \{\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx}) : \mathsf{Init}[\pi] = 1 \wedge \mathsf{sid}(\mathsf{ctx}) = \mathsf{sid} \wedge i \in \mathsf{S}\}$.

every aggregate target $A \in V$, define the affine fiber

$$\mathcal{F}_A := \left\{ (y_i)_{i \in \mathsf{S}} \in V^{\mathsf{S}} : \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} y_i = A \right\}.$$

If $(y_i)_{i \in \mathsf{S}}$ is uniform over $\mathcal{F}_A$, then for every proper subset $\mathsf{T} \subsetneq \mathsf{S}$, the projection $(y_i)_{i \in \mathsf{T}}$ is uniform over V^{T}. Moreover, after any proper subset T has been fixed to values $(z_i)_{i \in \mathsf{T}}$, the exact posterior distribution on the remaining coordinates is sampled by choosing all but one remaining coordinate uniformly in V and setting the final coordinate by the Lagrange equation.

We give the proof in section A.1.

Realization in the Random Oracle Model. We next give a concrete ROM construction that realizes $\mathcal{F}_\Delta$. The ideal functionality itself does not contain the random oracle. The random oracle appears only in the real implementation and in the simulator used to relate that implementation to $\mathcal{F}_\Delta$.

For every legal type $\mathsf{type} \in \mathcal{T}_\Delta$, let $\mathsf{H}_\Delta^{\mathsf{type}}$ be a typed random oracle whose answers lie in $\mathsf{D}(\mathsf{type})$. In each session instance, the construction samples one independent seed for every ordered pair of distinct users. A user's local mask is the weighted antisymmetric sum of the typed random-oracle values incident to that user.

The real execution used in the ROM realization exposes the concrete profile-level calls of Δ_{ROM} and a public typed random oracle. It is shown in fig. 3. A

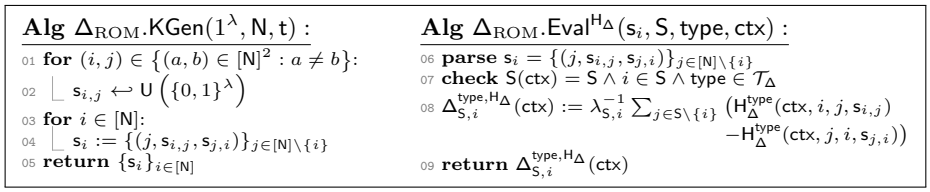


Fig. 2. Pairwise-seed ROM construction Δ_{ROM} for profile-wise zero-sum masking.

real corruption returns the concrete session seed tuple $\mathsf{s}_{i,\mathsf{sid}}$. It does not maintain an ideal opening state. The translation between revealed seed tuples and the abstract openings of $\mathcal{F}_\Delta$ is part of the simulator.

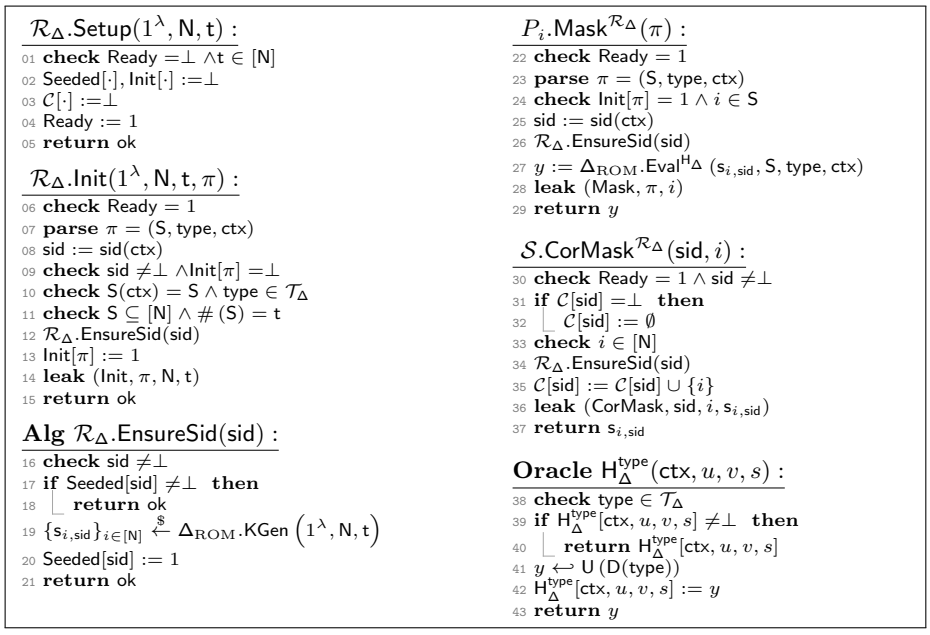


Fig. 3. Session-indexed real execution $\mathcal{R}_\Delta$ for the pairwise-seed masking construction in fig. 2.

Lemma 2 (Exact cancellation of the pairwise-seed masks). *For every legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$, every output $\{\mathsf{s}_i\}_{i \in [\mathsf{N}]}$ of $\Delta_{ROM}.\mathsf{KGen}\left(1^\lambda, \mathsf{N}, \mathsf{t}\right)$, and every random-oracle table, the values returned by $\Delta_{ROM}.\mathsf{Eval}^{\mathsf{H}_\Delta}(\mathsf{s}_i, \mathsf{S}, \mathsf{type}, \mathsf{ctx})$ satisfy*

$$\sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \Delta_{\mathsf{S},i}^{\mathsf{type},\mathsf{H}_\Delta}(\mathsf{ctx}) = 0.$$

We give the proof in section A.1.

Theorem 1 (ROM realization of profile-wise zero-sum masking). *The pairwise-seed construction Δ_{ROM} in fig. 2, run through the real execution $\mathcal{R}_\Delta$ in fig. 3, ROM-realizes the ideal functionality $\mathcal{F}_\Delta$ in fig. 1. The real execution and simulator keep seed-generation and corruption state scoped by* $\mathsf{sid} = \mathsf{sid}(\mathsf{ctx})$. *More precisely, for every* PPT *environment $\mathcal{Z}$ whose execution makes at most $\mathsf{Q_M}$ explicit public mask random-oracle queries, there exists a* PPT *simulator $\mathcal{S}_{ROM}$ such that*

$$\Big|\Pr\left[\mathcal{E}^{\mathsf{real},\mathsf{H}_\Delta}_{\mathcal{R}_\Delta,\mathcal{Z}}\left(1^\lambda,\mathsf{N},\mathsf{t}\right)=1\right] - \Pr\left[\mathcal{E}^{\mathsf{ideal}}_{\mathcal{F}_\Delta,\mathcal{S}_{ROM},\mathcal{Z}}\left(1^\lambda,\mathsf{N},\mathsf{t}\right)=1\right]\Big| \le \frac{\mathsf{Q_M}}{2^\lambda}.$$

The proof of the profile-wise ROM realization is given in section A. It follows the delayed-programming technique of [KRT24,Che25b], but the session-wise simulator, posterior sampling, and hidden activation invariants are spelled out there for the present interface.

4.4 Threshold Signature with Adaptive Security

Syntax. We define a two-round threshold signature scheme as follows. A centralized key-generation procedure KGen generates a single aggregate verification key vk_Σ and an individual secret key sk_i for each user. This centralized procedure can be replaced by a distributed key-generation procedure without affecting the security definition. The signing procedure has two rounds. In the first round, each signer generates a first-round message m_i and an internal state st_i. In the second round, each signer generates a signature share σ_i from the first-round messages and its internal state. The aggregation algorithm SAgg combines these shares into an aggregate signature σ_Σ. Any set of t correctly generated signature shares can produce a valid aggregate signature.

Identifier. We use an identifier id to uniquely identify each signing request. The identifier has the form $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, where sid is a session identifier and rid is a request identifier. Within any execution, id must uniquely determine the entire signing request, namely the pair (S, m) consisting of the signer set and the request message. Thus, every honest signing party identifies the signing request by id, and the same identifier must always refer to the same pair (S, m). In particular, two signing requests with different signer sets or different messages must use different identifiers. This requirement is crucial for security against adaptive adversaries that can interleave signing queries on the same or different requests and can corrupt users between these queries. It can be viewed as an adaptively secure variant of the security notion proposed by Bellare et al. [BCK+22] for statistical threshold signatures. Moreover, the identifier id is implicitly given to every signing algorithm and can be used to derive signing randomness. This is essential when adaptive corruptions reveal secret keys together with the randomness used in local signing sessions.

Definition 5 (Two-Round Threshold Signature). A two-round threshold signature scheme TSig consists of six PPT algorithms $\mathsf{TSig} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{SAgg}, \mathsf{Ver})$ with the following syntax.

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: Takes the unary representation of the security parameter λ, and outputs the public parameters pp. The public parameters pp are implicitly included in all other algorithms in the scheme.
- $\mathsf{KGen}(\mathsf{N}, \mathsf{t}) \to (\mathsf{vk}_\Sigma, (\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathsf{N}))$: Takes the number of users N, and the threshold t, and outputs the aggregate verification key vk_Σ and the secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathsf{N}$ for the N users.

The signing procedure consists of three algorithms:

- $\mathsf{Sig}_0(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{sk}_\ell, \mathsf{S}, \mathsf{m}) \to (\mathsf{m}_\ell, \mathsf{st}_\ell)$: Takes the aggregate verification key vk_Σ, a signer index ℓ, an identifier id, a secret key sk_ℓ, a set of signers S, and a request message m, and outputs a first-round message m_ℓ and an internal state st_ℓ for the ℓ-th user.
- $\mathsf{Sig}_1(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{st}_\ell, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{S}, \mathsf{m}) \to \sigma_\ell$: Takes the aggregate verification key vk_Σ, a signer index ℓ, an identifier id, an internal state st_ℓ, the set of first-round messages $\{\mathsf{m}_i\}_{i\in\mathsf{S}}$, the set of signers S, and a request message m, and outputs a signature share σ_ℓ for the ℓ-th user.
- $\mathsf{SAgg}(\mathsf{vk}_\Sigma, \{\sigma_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m}) \to \sigma_\Sigma$: Takes the aggregate verification key vk_Σ, a set of signature shares $\{\sigma_i\}_{i\in\mathsf{S}}$ from the users in S, an identifier id, a signer set S, and a request message m, and outputs a combined signature σ_Σ.

- $\mathsf{Ver}(\mathsf{vk}_\Sigma, \mathsf{id}, \mathsf{S}, \mathsf{m}, \sigma_\Sigma) \to \{0, 1\}$: Takes the aggregate verification key vk_Σ, an identifier id, a signer set S, a request message m, and a combined signature σ_Σ, and outputs 1 if the signature is valid and 0 otherwise.

Partially non-interactive. A two-round threshold signature scheme TSig is partially non-interactive if the first-round message m_ℓ generated by Sig_0 for each signer ℓ is independent of the request message m. In this case, the first-round messages from all signers can be generated in parallel without any interaction, and only the second-round signing procedures are online. We will focus on partially non-interactive threshold signature schemes in this work. With a slight abuse of notation, we write $\mathsf{Sig}_0(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{sk}_\ell, \mathsf{S})$ for $\mathsf{Sig}_0(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{sk}_\ell, \mathsf{S}, \mathsf{m})$ when the first-round message is independent of the request message.

For completeness, we define the honest joint signing procedure for a two-round threshold signature scheme TS. For an identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, the honest signing procedure for the users in S on the message m is defined in fig. 4 and denoted by $\mathsf{TSJointSig}(\mathsf{vk}_\Sigma, \{\mathsf{sk}_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m})$.

Definition 6 (Completeness). A two-round threshold signature scheme TSig is said to be complete if for every $\lambda \in \mathbb{N}$, every $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, every supported pair $(\mathsf{N}, \mathsf{t} - 1)$, every $(\mathsf{vk}_\Sigma, (\mathsf{sk}_1, \ldots, \mathsf{sk}_\mathsf{N})) \xleftarrow{\$} \mathsf{KGen}(\mathsf{N}, \mathsf{t})$, every identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, every message m, every set of users S such that $\#(\mathsf{S}) = \mathsf{t}$, we have

$$\Pr[\mathsf{Ver}(\mathsf{vk}_\Sigma, \mathsf{id}, \mathsf{S}, \mathsf{m}, \mathsf{TSJointSig}(\mathsf{vk}_\Sigma, \{\mathsf{sk}_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m})) = 1] = 1,$$

where the honest signing procedure $\mathsf{TSJointSig}$ is defined in fig. 4.

Alg $\mathsf{TSJointSig}(\mathsf{vk}_\Sigma, \{\mathsf{sk}_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m})$:

for $\ell \in \mathsf{S}$:
  $(\mathsf{m}_\ell, \mathsf{st}_\ell) \xleftarrow{\$} \mathsf{Sig}_0(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{sk}_\ell, \mathsf{S}, \mathsf{m})$
for $\ell \in \mathsf{S}$:
  $\sigma_\ell \xleftarrow{\$} \mathsf{Sig}_1(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \mathsf{st}_\ell, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{S}, \mathsf{m})$
$\sigma_\Sigma := \mathsf{SAgg}(\mathsf{vk}_\Sigma, \{\sigma_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m})$
return σ_Σ

Fig. 4. Honest joint signing procedure $\mathsf{TSJointSig}$ for a two-round threshold signature scheme $\mathsf{TSig} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{SAgg}, \mathsf{Ver})$ on identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ and message m.

Security. We define adaptive unforgeability with corruption for two-round threshold signature schemes in definition 7. In this security definition, the adversary first obtains the public parameters and may then query a key-generation oracle $\mathsf{KGen}(\mathsf{sid})$ on fresh session identifiers sid to register aggregate verification keys $\mathsf{vk}_\Sigma[\mathsf{sid}]$, where the supported pair $(\mathsf{N}, \mathsf{t}-1)$ is fixed throughout the experiment.

We write $\mathcal{C}$ for the set of globally corrupted users, and for each identifier id we write $\mathcal{Q}[\mathsf{id}]$ for the set of users from whom the adversary has already obtained a second-round answer on that identifier. Upon corrupting a user ℓ via $\mathsf{Corrupt}(\mathsf{sid}, \ell)$, the adversary obtains the secret key $\mathsf{sk}_{\ell,\mathsf{sid}}$ together with the randomness used in the local signing sessions of user ℓ: for every identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ such that $\mathsf{Sig}_0(\mathsf{vk}_\Sigma[\mathsf{sid}], \ell, \mathsf{id}, \mathsf{S}, \mathsf{m})$ has been called, it learns the first-round randomness $\mathsf{r}_{1,\ell,\mathsf{id}}$, and if $\mathsf{Sig}_1(\mathsf{vk}_\Sigma[\mathsf{sid}], \ell, \mathsf{id}, \cdots, \mathsf{m})$ has already been called, it additionally learns the second-round randomness $\mathsf{r}_{2,\ell,\mathsf{id}}$.

Throughout the experiment, the same identifier id must always be used with the same pair (S, m), recorded as $(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}])$. A legal second-round query $\mathsf{Sig}_1(\mathsf{vk}_\Sigma, \ell, \mathsf{id}, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{m})$ with $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ must satisfy $\mathsf{vk}_\Sigma = \mathsf{vk}_\Sigma[\mathsf{sid}]$, must target an uncorrupted signer ℓ that has already completed round 1 on $(\mathsf{id}, \mathsf{S}, \mathsf{m})$, and the first-round message supplied for that queried signer must equal its previously recorded round-1 message. Apart from this consistency requirement for the queried signer itself, the remaining first-round transcript supplied to Sig_1 is chosen by the adversary.

The freshness condition for a forgery on $(\mathsf{id}^\star, \mathsf{S}^\star, \mathsf{m}^\star)$ requires that $\mathsf{S}^\star$ be a valid signer set of size t and that at most $\mathsf{t}-1$ users are either corrupted or have already been queried to complete round 2 on identifier $\mathsf{id}^\star$. The adversary wins if it outputs a valid combined signature satisfying this freshness condition.

Definition 7 (Adaptive Threshold Session Unforgeability with Corruption). A two-round threshold signature scheme TSig as defined in definition 5 is said to be adaptively $(\mathsf{Q}_{\mathsf{Session}}, \mathsf{Q}_{\mathsf{Cor}}, \mathsf{Q}_{\mathsf{S}})$-$\mathsf{aTS\text{-}sUF}$-secure with corruption if, for every security parameter $\lambda \in \mathbb{N}$, every supported pair $(\mathsf{N}, \mathsf{t}-1)$, and every PPT adversary $\mathcal{A}$ that makes at most $\mathsf{Q}_{\mathsf{Session}}$ key-generation queries $\mathsf{KGen}(\mathsf{sid})$, at most Q_{S} second-round signing queries, and at most $\mathsf{Q}_{\mathsf{Cor}}$ corruption queries with $\mathsf{Q}_{\mathsf{Cor}} \le \mathsf{t}-1$, the advantage of $\mathcal{A}$ in the security experiment $\mathsf{Exp}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TSig}}(\lambda)$ as defined in figs. 5 and 6 is negligible.

Namely,

$$\mathsf{Adv}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TSig}}(\lambda) := \Pr\left[\mathsf{Exp}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TSig}}(\lambda) = 1\right] \in \mathsf{negl}(\lambda).$$

Alg $\mathsf{Exp}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TS}}(\lambda)$:

01 **for** sid : $\mathsf{vk}_\Sigma[\mathsf{sid}] := \perp$; $\mathcal{C}[\mathsf{sid}] := \emptyset$
02 **for** id : $\mathcal{Q}[\mathsf{id}] := \emptyset$; $\mathsf{S}[\mathsf{id}] := \perp$; $\mathsf{m}[\mathsf{id}] := \perp$
03 **for** id : **for** ℓ : $\mathsf{Round}_{\ell,\mathsf{id}} = 0$
04 $\mathsf{pp} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
05 $\mathsf{Sig} = (\mathsf{KGen}, \mathsf{Sig}_0, \mathsf{Sig}_1, \mathsf{Corrupt})$
06 $((\mathsf{id}^\star, \mathsf{S}^\star, \mathsf{m}^\star), \sigma^\star_\Sigma) \xleftarrow{\$} \mathcal{A}^{\mathsf{Sig}}(\mathsf{pp})$
07 **parse** $(\mathsf{sid}^\star, \mathsf{rid}^\star) =: \mathsf{id}^\star$
08 $\mathsf{Fresh} := \mathsf{vk}_\Sigma[\mathsf{sid}^\star] \neq \perp \wedge \mathsf{S}^\star \subseteq [\mathsf{N}] \wedge \#(\mathsf{S}^\star) = \mathsf{t}$
$\wedge((\mathsf{S}[\mathsf{id}^\star], \mathsf{m}[\mathsf{id}^\star]) = (\perp, \perp) \vee (\mathsf{S}[\mathsf{id}^\star], \mathsf{m}[\mathsf{id}^\star]) = (\mathsf{S}^\star, \mathsf{m}^\star))$
$\wedge \#(\mathcal{Q}[\mathsf{id}^\star] \cup \mathcal{C}[\mathsf{sid}^\star]) \leq \mathsf{t} - 1$
09 **return** $[\![\mathsf{Fresh} \wedge \mathsf{Ver}(\mathsf{vk}_\Sigma[\mathsf{sid}^\star], \mathsf{id}^\star, \mathsf{S}^\star, \mathsf{m}^\star, \sigma^\star_\Sigma) = 1]\!]$

Fig. 5. Main experiment of $\mathsf{Exp}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TSig}}$ for a two-round threshold signature scheme TS.

4.5 Linear Function Families

We use the vector-space formulation of linear function families from [PW23].

Definition 8 (Linear Function Family). A linear function family LFF is a pair of PPT algorithms $\mathsf{LFF} = (\mathsf{Setup}, \mathsf{F})$ with the following syntax.

- $\mathsf{Setup}\left(1^\lambda\right) \to \mathsf{pp}_{\mathsf{LFF}}$: Takes the unary representation of the security parameter λ, and outputs parameters $\mathsf{pp}_{\mathsf{LFF}}$. The parameters implicitly define a scalar field $\mathsf{S}_{\mathsf{LFF}}$, a domain $\mathsf{D}_{\mathsf{LFF}}$, and a co-domain $\mathsf{CD}_{\mathsf{LFF}}$. The set $\mathsf{S}_{\mathsf{LFF}}$ is a finite field, and $\mathsf{D}_{\mathsf{LFF}}$ and $\mathsf{CD}_{\mathsf{LFF}}$ are finite-dimensional $\mathsf{S}_{\mathsf{LFF}}$-vector spaces from which uniform sampling is efficient.
- $\mathsf{F}(x) \to X$: Deterministically takes an element $x \in \mathsf{D}_{\mathsf{LFF}}$, and outputs an element $X \in \mathsf{CD}_{\mathsf{LFF}}$.

In addition, LFF admits a trapdoor parameter-generation interface

$$\mathsf{Setup}_{\mathsf{im}}\left(1^\lambda\right) \to (\mathsf{pp}_{\mathsf{LFF}}, \mathsf{td}_{\mathsf{im}})$$

and a deterministic predicate

$$\mathsf{InImage}(\mathsf{td}_{\mathsf{im}}, X) \to \{0, 1\}$$

with the following properties. First, the marginal distribution of $\mathsf{pp}_{\mathsf{LFF}}$ output by $\mathsf{Setup}_{\mathsf{im}}\left(1^\lambda\right)$ is identical to the distribution of $\mathsf{pp}_{\mathsf{LFF}}$ output by $\mathsf{Setup}\left(1^\lambda\right)$. Thus, if $\mathsf{td}_{\mathsf{im}}$ is not revealed, the induced distribution of the public map F is

Oracle $\mathsf{KGen}(\mathsf{sid})$:

$\mathbf{check}\ \mathsf{vk}_\Sigma[\mathsf{sid}] = \perp$

$(\mathsf{vk}_\Sigma[\mathsf{sid}], (\mathsf{sk}_{1,\mathsf{sid}}, \ldots, \mathsf{sk}_{N,\mathsf{sid}})) \xleftarrow{\$} \mathsf{KGen}(\mathsf{N}, \mathsf{t})$

$\mathbf{return}\ \mathsf{vk}_\Sigma[\mathsf{sid}]$

Oracle $\mathsf{Corrupt}(\mathsf{sid}, \ell)$:

$\mathbf{check}\ \mathsf{vk}_\Sigma[\mathsf{sid}] \neq \perp \wedge \ell \in [\mathsf{N}] \wedge \#(\mathcal{C}[\mathsf{sid}] \cup \{\ell\}) \leq \mathsf{t} - 1$

$\mathcal{C}[\mathsf{sid}] := \mathcal{C}[\mathsf{sid}] \cup \{\ell\}$

$\mathbf{for}\ \mathsf{id}' = (\mathsf{sid}, \mathsf{rid}') \in \mathsf{D}(\mathsf{Round}_{\ell,\cdot})$:

$\quad \mathsf{Leakr}_{\ell,\mathsf{id}'} := \begin{cases} \mathsf{r}_{1,\ell,\mathsf{id}'} & \text{if } \mathsf{Round}_{\ell,\mathsf{id}'} = 1, \\ (\mathsf{r}_{1,\ell,\mathsf{id}'}, \mathsf{r}_{2,\ell,\mathsf{id}'}) & \text{if } \mathsf{Round}_{\ell,\mathsf{id}'} = 2 \end{cases}$

$\mathsf{Leakr}_\ell := \{(\mathsf{id}', \mathsf{Leakr}_{\ell,\mathsf{id}'})\}_{\mathsf{id}' = (\mathsf{sid},\mathsf{rid}') \in \mathsf{D}(\mathsf{Round}_{\ell,\cdot})}$

$\mathbf{return}\ (\mathsf{sk}_{\ell,\mathsf{sid}}, \mathsf{Leakr}_\ell)$

Oracle $\mathsf{Sig}_0(\mathsf{vk}'_\Sigma, \ell, \mathsf{id}, \mathsf{S}, \mathsf{m})$:

$\mathbf{parse}\ \mathsf{id} = (\mathsf{sid}, \mathsf{rid})$

$\mathbf{check}\ \mathsf{vk}_\Sigma[\mathsf{sid}] \neq \perp \wedge \mathsf{vk}'_\Sigma = \mathsf{vk}_\Sigma[\mathsf{sid}] \wedge \ell \notin \mathcal{C}[\mathsf{sid}]$

$\mathbf{check}\ \ell \in [\mathsf{N}] \wedge \#(\mathsf{S}) = \mathsf{t}$

$\mathbf{check}\ \ell \in \mathsf{S} \wedge \mathsf{Round}_{\ell,\mathsf{id}} = 0$

$\mathbf{check}\ (\mathsf{S}[\mathsf{id}] = \perp \wedge \mathsf{m}[\mathsf{id}] = \perp) \vee (\mathsf{S}[\mathsf{id}] = \mathsf{S} \wedge \mathsf{m}[\mathsf{id}] = \mathsf{m})$

$(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) := (\mathsf{S}, \mathsf{m})$

$\mathsf{Round}_{\ell,\mathsf{id}} := 1$

$(\mathsf{m}_{\ell,\mathsf{id}}, \mathsf{st}_{\ell,\mathsf{id}}) \xleftarrow{\$} \mathsf{Sig}_0(\mathsf{vk}'_\Sigma, \ell, \mathsf{id}, \mathsf{sk}_{\ell,\mathsf{sid}}, \mathsf{S}, \mathsf{m}; \mathsf{r}_{1,\ell,\mathsf{id}})$

$\mathbf{return}\ \mathsf{m}_{\ell,\mathsf{id}}$

Oracle $\mathsf{Sig}_1(\mathsf{vk}'_\Sigma, \ell, \mathsf{id}, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{S}, \mathsf{m})$:

$\mathbf{parse}\ \mathsf{id} = (\mathsf{sid}, \mathsf{rid})$

$\mathbf{check}\ \mathsf{vk}_\Sigma[\mathsf{sid}] \neq \perp \wedge \mathsf{vk}'_\Sigma = \mathsf{vk}_\Sigma[\mathsf{sid}] \wedge \ell \notin \mathcal{C}[\mathsf{sid}]$

$\mathbf{check}\ \ell \in [\mathsf{N}] \wedge \#(\mathsf{S}) = \mathsf{t}$

$\mathbf{check}\ \ell \in \mathsf{S} \wedge \mathsf{Round}_{\ell,\mathsf{id}} = 1$

$\mathbf{check}\ (\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) = (\mathsf{S}, \mathsf{m})$

$\mathbf{check}\ \mathsf{m}_\ell = \mathsf{m}_{\ell,\mathsf{id}}$

$\mathcal{Q}[\mathsf{id}] := \mathcal{Q}[\mathsf{id}] \cup \{\ell\}$

$\mathsf{Round}_{\ell,\mathsf{id}} := 2$

$\sigma_{\ell,\mathsf{id}} \xleftarrow{\$} \mathsf{Sig}_1(\mathsf{vk}'_\Sigma, \ell, \mathsf{id}, \mathsf{st}_{\ell,\mathsf{id}}, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{S}, \mathsf{m}; \mathsf{r}_{2,\ell,\mathsf{id}})$

$\mathbf{return}\ \sigma_{\ell,\mathsf{id}}$

Fig. 6. Oracles for the security experiment $\mathsf{Exp}^{\mathsf{aTS\text{-}sUF}}_{\mathcal{A},\mathsf{TSig}}$. The adversary may make at most $\mathsf{Q}_{\mathsf{Session}}$ key-generation queries to KGen on fresh session identifiers, at most Q_{S} second-round signing queries to Sig_1, and at most $\mathsf{Q}_{\mathsf{Cor}}$ corruption queries. The key-generation oracle registers the aggregate verification key as $\mathsf{vk}_\Sigma[\mathsf{sid}]$ under the chosen session identifier sid, for the fixed supported pair $(\mathsf{N}, \mathsf{t}-1)$. Notice that Sig_0 must be queried before Sig_1 for the same signer and identifier, that the same identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ must always refer to the same pair (S, m) recorded as $(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}])$, and that $\mathcal{Q}[\mathsf{id}]$ records the signers from whom the adversary has already obtained a second-round answer on identifier id.

unchanged. Second, for every $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{td}_{\mathsf{im}})$ output by $\mathsf{Setup}_{\mathsf{im}}\left(1^\lambda\right)$ and every $X \in \mathsf{CD}_{\mathsf{LFF}}$,

$$\mathsf{InImage}\left(\mathsf{td}_{\mathsf{im}}, X\right) = 1 \quad \Longleftrightarrow \quad X \in \{\mathsf{F}\left(x\right) : x \in \mathsf{D}_{\mathsf{LFF}}\}.$$

Throughout the paper, the parameters $\mathsf{pp}_{\mathsf{LFF}}$ are implicit in F once they are fixed by the surrounding experiment, and we do not mention $\mathsf{pp}_{\mathsf{LFF}}$ explicitly in calls to F. We also write the field and vector-space operations as $+$ and $\cdot$. For

every parameter $\mathsf{pp}_{\mathsf{LFF}}$, the map $\mathsf{F}(\cdot)$ is linear. Namely, for every $c \in \mathsf{S}_{\mathsf{LFF}}$ and every $x, y \in \mathsf{D}_{\mathsf{LFF}}$,

$$\mathsf{F}(c \cdot x + y) = c \cdot \mathsf{F}(x) + \mathsf{F}(y).$$

When a linear function family is used with the threshold-signature construction, we identify its scalar field $\mathsf{S}_{\mathsf{LFF}}$ with the global field $\mathbb{F}$ used for the signer labels and Lagrange coefficients in section 4.2. Thus every $\lambda_{\mathsf{S},i}$ is a valid scalar for $\mathsf{D}_{\mathsf{LFF}}$, $\mathsf{CD}_{\mathsf{LFF}}$, and the $\mathsf{S}_{\mathsf{LFF}}$-vector spaces attached to the commitment interface.

Definition 9 (Lossiness-Admitting LFF). A linear function family $\mathsf{LFF} = (\mathsf{Setup}, \mathsf{F})$ is ε_{l}-lossiness admitting if the following properties hold.

Key Indistinguishability. For every polynomially bounded $Q = Q(\lambda)$ and every PPT adversary $\mathcal{A}$, the following advantage is negligible in λ, where $\mathbf{X} := \{X_j\}_{j \in [Q]}$:

$$\mathsf{Adv}^{Q\text{-}\mathsf{KeyInd}}_{\mathcal{A},\mathsf{LFF}}(1^\lambda) = \left| \Pr\left[\mathcal{A}(\mathsf{pp}_{\mathsf{LFF}}, \mathbf{X}) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}(1^\lambda), \\ \textbf{for } j \in [Q]: \\ \quad x_j \hookleftarrow \mathsf{U}(\mathsf{D}_{\mathsf{LFF}}), \\ \quad X_j := \mathsf{F}(x_j) \end{array} \right] - \Pr\left[\mathcal{A}(\mathsf{pp}_{\mathsf{LFF}}, \mathbf{X}) = 1 \;\middle|\; \begin{array}{l} \mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}(1^\lambda), \\ \textbf{for } j \in [Q]: \\ \quad X_j \hookleftarrow \mathsf{U}(\mathsf{CD}_{\mathsf{LFF}}) \end{array} \right] \right| \in \mathsf{negl}(\lambda).$$

Lossy Soundness. For every unbounded adversary $\mathcal{A}$, the following probability is at most ε_{l}:

$$\Pr\left[X \notin \mathsf{F}(\mathsf{D}_{\mathsf{LFF}}) \wedge \mathsf{F}(s) - X \cdot c = R \;\middle|\; \begin{array}{l} \mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}(1^\lambda), \\ (\mathsf{st}, X, R) := \mathcal{A}(\mathsf{pp}_{\mathsf{LFF}}), \\ c \hookleftarrow \mathsf{U}(\mathsf{S}_{\mathsf{LFF}}), \\ s := \mathcal{A}(\mathsf{st}, c) \end{array} \right] \leq \varepsilon_{\mathsf{l}}.$$

5 Special Trapdoor Commitments

We use a linearly homomorphic commitment primitive with a targetable trapdoor opening interface. The honest protocol uses only the public commitment interface. In the proof, commitments generated under simulation keys carry proof-only state that can later be opened, after the Fiat–Shamir challenge is known, to a target of the form $\mathsf{F}(\rho) + Xc$. The targetable opening algorithm is given proof-only representatives of the source and target group elements. These representatives are produced by the simulator when the corresponding group elements are sampled and are not part of the public signature syntax.

Definition 10 (Special Trapdoor Commitment with Targetable Opening). Let $\mathsf{LFF} = (\mathsf{Setup}, \mathsf{F})$ be a linear function family. An $(\varepsilon_{\mathsf{b}}(\lambda, \cdot), \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$-special trapdoor commitment scheme with targetable opening for LFF, with Q-key binding error $\varepsilon_{\mathsf{b}}(\lambda, Q)$, is a tuple of PPT algorithms

$$\mathsf{CMT} = (\mathsf{BGen}, \mathsf{SGen}, \mathsf{Com}, \mathsf{TCom}, \mathsf{TCol}).$$

For every parameter $\mathsf{pp}_{\mathsf{LFF}}$ output by Setup, the commitment interface treats all public auxiliary data that it needs as part of $\mathsf{pp}_{\mathsf{LFF}}$. These parameters determine a key space $\mathcal{K}_{\mathsf{LFF}}$, finite-dimensional $\mathsf{S}_{\mathsf{LFF}}$-vector spaces $\mathcal{O}_{\mathsf{LFF}}$, $\mathcal{R}_{\mathsf{LFF}}$, and $\mathcal{C}_{\mathsf{LFF}}$, a linear representation map $\mathsf{Rep}_{\mathsf{LFF}} : \mathcal{R}_{\mathsf{LFF}} \to \mathsf{CD}_{\mathsf{LFF}}$, an image embedding $\mathsf{Emb}_{\mathsf{LFF}} : \mathsf{D}_{\mathsf{LFF}} \to \mathcal{R}_{\mathsf{LFF}}$, and a trace space $\mathcal{T}_{\mathsf{LFF}}$, such that we have $\mathsf{Rep}_{\mathsf{LFF}}(\mathsf{Emb}_{\mathsf{LFF}}(\rho)) = \mathsf{F}(\rho)$ for every $\rho \in \mathsf{D}_{\mathsf{LFF}}$. We write the vector-space operations on $\mathcal{O}_{\mathsf{LFF}}$ and $\mathcal{C}_{\mathsf{LFF}}$ additively. The algorithms have the following syntax.

- $\mathsf{BGen}(\mathsf{pp}_{\mathsf{LFF}}) \to \mathsf{ck}$: Takes LFF parameters $\mathsf{pp}_{\mathsf{LFF}}$ and outputs a commitment key $\mathsf{ck} \in \mathcal{K}_{\mathsf{LFF}}$.
- $\mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}}) \to (\mathsf{ck}, \mathsf{tk})$: Takes LFF parameters and outputs a simulation commitment key $\mathsf{ck} \in \mathcal{K}_{\mathsf{LFF}}$ together with a trapdoor tk.
- $\mathsf{Com}(\mathsf{ck}, R; \omega) \to \mathsf{c}$: Takes a key ck, a message $R \in \mathsf{CD}_{\mathsf{LFF}}$, and an opening $\omega \in \mathcal{O}_{\mathsf{LFF}}$, and outputs a commitment $\mathsf{c} \in \mathcal{C}_{\mathsf{LFF}}$.
- $\mathsf{TCom}(\mathsf{ck}, \mathsf{tk}, R, \omega_R; \omega) \to (\mathsf{c}, \tau)$: Takes a simulation key and trapdoor, a source message $R \in \mathsf{CD}_{\mathsf{LFF}}$, a representative $\omega_R \in \mathcal{R}_{\mathsf{LFF}}$ satisfying $\mathsf{Rep}_{\mathsf{LFF}}(\omega_R) = R$, and an opening $\omega \in \mathcal{O}_{\mathsf{LFF}}$. It returns $\mathsf{c} = \mathsf{Com}(\mathsf{ck}, R; \omega)$ together with proof-only state $\tau \in \mathcal{T}_{\mathsf{LFF}}$.
- $\mathsf{TCol}(\mathsf{ck}, \mathsf{tk}, \mathsf{c}, \tau, \omega, \omega_X, c, \rho) \to \omega'$: Takes a traced commitment record, a target representative $\omega_X \in \mathcal{R}_{\mathsf{LFF}}$, a challenge $c \in \mathsf{S}_{\mathsf{LFF}}$, and a response $\rho \in \mathsf{D}_{\mathsf{LFF}}$. It outputs an opening $\omega' \in \mathcal{O}_{\mathsf{LFF}}$ for the same commitment to the target $\mathsf{F}(\rho) + \mathsf{Rep}_{\mathsf{LFF}}(\omega_X)\, c$.

The algorithms satisfy the following properties.

Linear homomorphism. For every $\mathsf{pp}_{\mathsf{LFF}}$, every $\gamma \in \mathsf{S}_{\mathsf{LFF}}$, every $\mathsf{ck} \in \mathcal{K}_{\mathsf{LFF}}$, every $R_0, R_1 \in \mathsf{CD}_{\mathsf{LFF}}$, and every $\omega_0, \omega_1 \in \mathcal{O}_{\mathsf{LFF}}$,

$$\begin{aligned}&\mathsf{Com}(\mathsf{ck}, R_0; \omega_0) + \gamma \cdot \mathsf{Com}(\mathsf{ck}, R_1; \omega_1)\\ &\quad = \mathsf{Com}(\mathsf{ck}, R_0 + \gamma \cdot R_1; \omega_0 + \gamma \cdot \omega_1).\end{aligned}$$

Good simulation keys. There is an efficiently decidable set $\mathsf{Good}_{\mathsf{sim}}$ of triples $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}, \mathsf{tk})$ such that

$$\Pr\left[(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}, \mathsf{tk}) \notin \mathsf{Good}_{\mathsf{sim}} \;\middle|\; \begin{array}{l}\mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}(1^{\lambda}),\\ (\mathsf{ck}, \mathsf{tk}) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})\end{array}\right] \leq \varepsilon_{\mathsf{g}}.$$

Representative sampling. For every good simulation triple $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}, \mathsf{tk})$, the map $\mathsf{Rep}_{\mathsf{LFF}}$ is surjective. Moreover, there is an efficient proof-only sampler $\mathsf{SampRep}_{\mathsf{LFF}}(\mathsf{pp}_{\mathsf{LFF}})$ that outputs a pair (ω, X) with $\omega \in \mathcal{R}_{\mathsf{LFF}}$ and

$X \in \mathsf{CD}_{\mathsf{LFF}}$. The sampler always satisfies $X = \mathsf{Rep}_{\mathsf{LFF}}(\omega)$, and the marginal distribution of X is uniform over $\mathsf{CD}_{\mathsf{LFF}}$. Proof procedures may keep such representatives as proof-only state. If represented values are later combined linearly, the corresponding representatives are combined by the same scalars in $\mathcal{R}_{\mathsf{LFF}}$.

Uniform binding keys. For every parameter $\mathsf{pp}_{\mathsf{LFF}}$, the distribution of

$$(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}) \quad \text{with} \quad \mathsf{ck} \hookleftarrow \mathsf{U}(\mathcal{K}_{\mathsf{LFF}})$$

is identical to the distribution of

$$(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}) \quad \text{with} \quad \mathsf{ck} \xleftarrow{\$} \mathsf{BGen}(\mathsf{pp}_{\mathsf{LFF}}).$$

Trace correctness. For every $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}, \mathsf{tk}) \in \mathsf{Good}_{\mathsf{sim}}$, every $R \in \mathsf{CD}_{\mathsf{LFF}}$, every $\omega_R \in \mathcal{R}_{\mathsf{LFF}}$ satisfying $\mathsf{Rep}_{\mathsf{LFF}}(\omega_R) = R$, and every $\omega \in \mathcal{O}_{\mathsf{LFF}}$, if $(\mathsf{c}, \tau) := \mathsf{TCom}(\mathsf{ck}, \mathsf{tk}, R, \omega_R; \omega)$, then $\mathsf{c} = \mathsf{Com}(\mathsf{ck}, R; \omega)$.

Targetable trapdoor opening. For every $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}, \mathsf{tk}) \in \mathsf{Good}_{\mathsf{sim}}$, every randomness $\omega_R, \omega_X \in \mathcal{R}_{\mathsf{LFF}}$, every $c \in \mathsf{S}_{\mathsf{LFF}}$, every $\rho \in \mathsf{D}_{\mathsf{LFF}}$, and every $\omega \in \mathcal{O}_{\mathsf{LFF}}$, let $R := \mathsf{Rep}_{\mathsf{LFF}}(\omega_R)$, $X := \mathsf{Rep}_{\mathsf{LFF}}(\omega_X)$, and

$$(\mathsf{c}, \tau) := \mathsf{TCom}(\mathsf{ck}, \mathsf{tk}, R, \omega_R; \omega)$$

and define the posterior opening fiber

$$\mathsf{Post}_{\mathsf{ck},\mathsf{c},\tau,\omega_X,c,\rho} := \{\omega' \in \mathcal{O}_{\mathsf{LFF}} : \mathsf{Com}(\mathsf{ck}, \mathsf{F}(\rho) + Xc; \omega') = \mathsf{c}\}.$$

This fiber is nonempty. Moreover, we have that the output distribution of $\mathsf{TCol}(\mathsf{ck}, \mathsf{tk}, \mathsf{c}, \tau, \omega, \omega_X, c, \rho)$ is within statistical distance ε_{t} of the uniform distribution over $\mathsf{Post}_{\mathsf{ck},\mathsf{c},\tau,\omega_X,c,\rho}$. In particular, except with this statistical error,

$$\mathsf{Com}(\mathsf{ck}, \mathsf{F}(\rho) + Xc; \mathsf{TCol}(\mathsf{ck}, \mathsf{tk}, \mathsf{c}, \tau, \omega, \omega_X, c, \rho)) = \mathsf{c}.$$

Multi-key indistinguishability. For every polynomially bounded $Q = Q(\lambda)$ and every PPT adversary $\mathcal{A}$, define experiments $\mathsf{KeyDist}^0_{\mathsf{CMT},\mathcal{A}}(1^\lambda, Q)$ and $\mathsf{KeyDist}^1_{\mathsf{CMT},\mathcal{A}}(1^\lambda, Q)$ as follows. Both experiments sample $\mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$. The first experiment samples $\mathsf{ck}_j \xleftarrow{\$} \mathsf{BGen}(\mathsf{pp}_{\mathsf{LFF}})$ for all $j \in [Q]$. The second experiment samples $(\mathsf{ck}_j, \mathsf{tk}_j) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$ for all $j \in [Q]$. Both experiments return $\mathcal{A}(\mathsf{pp}_{\mathsf{LFF}}, \{\mathsf{ck}_j\}_{j\in[Q]})$. The advantage

$$\begin{aligned}\mathsf{Adv}^{\mathsf{keydist}}_{\mathsf{CMT},\mathcal{A}}(1^\lambda, Q) := \big|&\Pr\left[\mathsf{KeyDist}^0_{\mathsf{CMT},\mathcal{A}}(1^\lambda, Q) = 1\right] \\ &- \Pr\left[\mathsf{KeyDist}^1_{\mathsf{CMT},\mathcal{A}}(1^\lambda, Q) = 1\right]\big|\end{aligned}$$

is negligible in λ.

Adaptive multi-key statistical binding. There exists an unbounded extractor Ext such that, for every polynomially bounded $Q = Q(\lambda)$ and every unbounded adversary $\mathcal{A}$, the following probability is at most $\varepsilon_{\mathsf{b}}(\lambda, Q)$. Sample

$\mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}\left(1^{\lambda}\right)$ and independent binding keys $\mathsf{ck}_j \xleftarrow{\$} \mathsf{BGen}\left(\mathsf{pp}_{\mathsf{LFF}}\right)$ for all $j \in [Q]$. Let $(j^{\star}, \mathsf{c}, \mathsf{st}) \xleftarrow{\$} \mathcal{A}\left(\mathsf{pp}_{\mathsf{LFF}}, \{\mathsf{ck}_j\}_{j\in[Q]}\right)$. If $j^{\star} \notin [Q]$, set $\mathsf{BadBind} = 0$. Otherwise define $R := \mathsf{Ext}\left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}_{j^{\star}}, \mathsf{c}\right)$, let $(R_0, \omega_0) \xleftarrow{\$} \mathcal{A}\left(\mathsf{st}\right)$, and set $\mathsf{BadBind} = 1$ exactly when $\mathsf{Com}\left(\mathsf{ck}_{j^{\star}}, R_0; \omega_0\right) = \mathsf{c}$ and $R \neq R_0$. Then

$$\Pr\left[\mathsf{BadBind} = 1\right] \leq \varepsilon_{\mathsf{b}}\left(\lambda, Q\right).$$

6 Pilaf: Fully Tight Threshold Signatures with Adaptive Corruptions

In this section, we present Pilaf, a tightly secure threshold signature scheme against adaptive corruptions.

Definition 11 (Pilaf mask types and contexts). The construction in fig. 7 uses the following legal mask types for the profile-wise masking interface of definition 2:

$$\begin{aligned} \mathsf{D}(\mathsf{c}) &:= \mathcal{C}_{\mathsf{LFF}} \times \mathcal{C}_{\mathsf{LFF}}, & \mathsf{D}(X) &:= \mathsf{CD}_{\mathsf{LFF}}, \\ \mathsf{D}(R) &:= \mathsf{CD}_{\mathsf{LFF}}, & \mathsf{D}(\rho) &:= \mathsf{D}_{\mathsf{LFF}}, \\ \mathsf{D}(\omega) &:= \mathcal{O}_{\mathsf{LFF}} \times \mathcal{O}_{\mathsf{LFF}}. \end{aligned}$$

Type c masks the branch commitment pair $\mathsf{c}_{\ell,b} = (\mathsf{x}_{\ell,b}, \mathsf{r}_{\ell,b})$, types X and R mask LFF images, type ρ masks response preimages, and type ω masks opening pairs $\omega_{\ell,b} = \left(\omega^{X}_{\ell,b}, \omega^{R}_{\ell,b}\right)$.

Pilaf uses two context forms. First-round commitment masks use

$$\mathsf{ctx}_{\mathsf{c}} := (\mathsf{id}, \mathsf{S}, b).$$

Second-round masks use

$$\mathsf{ctx}_{\mathsf{sinf}} := (\mathsf{sinf}, b), \qquad \mathsf{sinf} := (\mathsf{vk}_{\Sigma}, \tau, \mathsf{id}, \mathsf{S}, \mathsf{m}).$$

Both context forms satisfy the parsing convention of definition 3. Hence every call $\Delta^{\mathsf{c}}_{i}(\mathsf{id}, \mathsf{S}, b)$ and every call $\Delta^{\mathsf{type}}_{i}(\mathsf{sinf}, b)$ for $\mathsf{type} \in \{X, R, \rho, \omega\}$ in fig. 7 is a legal $\mathcal{F}_{\Delta}$ profile whenever $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, $\mathsf{S} \subseteq [\mathsf{N}]$, and $\#\left(\mathsf{S}\right) = \mathsf{t}$.

Theorem 2 (Completeness). *If the masking scheme used in fig. 7 is zero-sum-correct on legal profiles, then the threshold signature scheme* Pilaf *satisfies completeness as in definition 6. In particular, this holds for the ROM masking construction of fig. 2.*

We give the proof in section B.

For the proof of theorem 3, we use the masking-context parsing convention from definition 3. Throughout the proof of theorem 3, ordered-pair seed variables $\mathsf{s}_{i,j}[\mathsf{sid}]$ and distinguished table entries $\mathsf{T}^{\mathsf{type}}_{\Delta}(\mathsf{ctx}, i, j)$ are used only for $i \neq j$. Queries with $i = j$ are treated as ordinary non-distinguished random-oracle points and never enter the pairwise masking table.

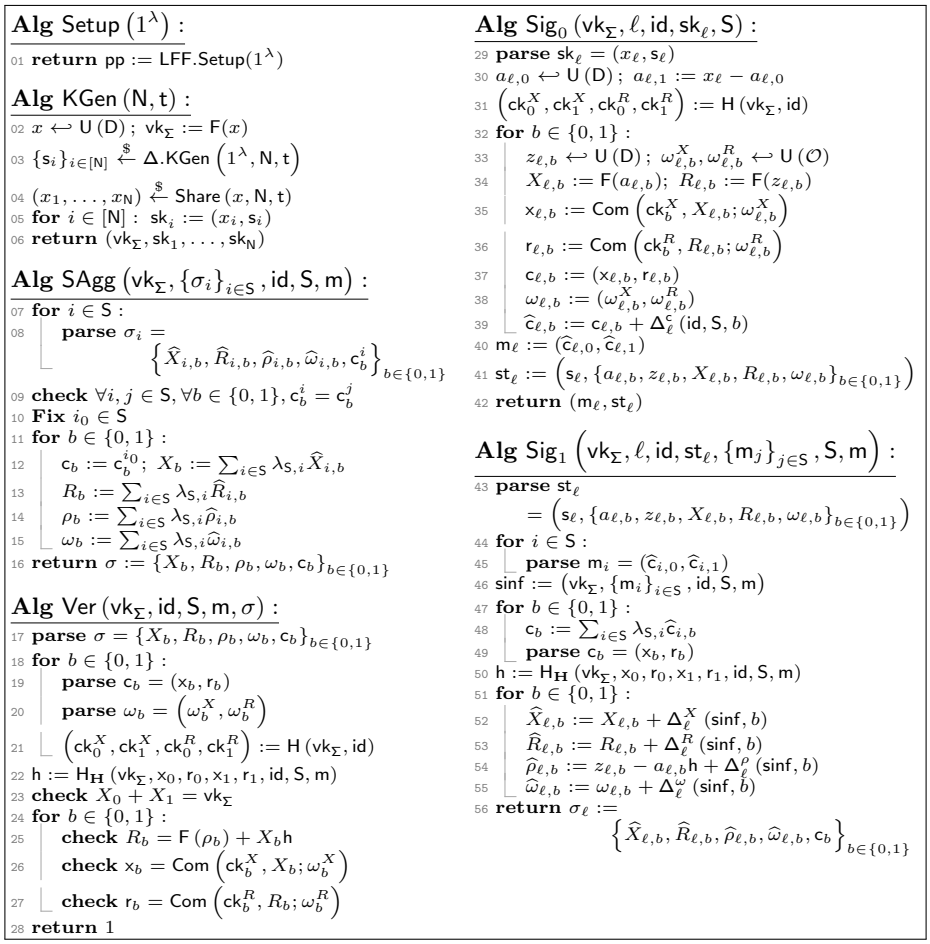


Fig. 7. Two-round threshold signature scheme Pilaf from MDDH, masking Δ, and Com of definition 10. Each branch commitment is $\mathsf{c}_b = (\mathsf{x}_b, \mathsf{r}_b)$, where x_b opens to X_b and r_b opens to R_b.

Definition 12 (Touched set). Fix an execution prefix and a second-round transcript

$$\mathsf{sinf} = (\mathsf{vk}_\Sigma, \tau, \mathsf{id}, \mathsf{S}, \mathsf{m}), \qquad \mathsf{id} = (\mathsf{sid}, \mathsf{rid}),$$

where $\tau = \{\widetilde{\mathsf{m}}_i\}_{i\in\mathsf{S}}$ is the first-round transcript supplied to a legal second-round query. Let $\mathcal{Q}_2[\mathsf{sinf}]$ be the set of signers for which the challenger has already returned a legal second-round answer on the exact transcript sinf. The *touched set* of sinf is

$$\mathsf{Touch}[\mathsf{sinf}] := (\mathcal{Q}_2[\mathsf{sinf}] \cup \mathcal{C}[\mathsf{sid}]) \cap \mathsf{S}.$$

The set $\mathsf{Touch}[\mathsf{sinf}]$ is proof bookkeeping for the adversarial view. It records which signers on the transcript sinf have already entered the view through either a corruption or a second-round answer. It is not the set of materialized decompositions. A second-round answer fixes the masked equations returned to the adversary, but it does not separately fix the unmasked local coordinates or the mask coordinates that sum to those values. A later corruption materializes such decompositions only through the joint posterior sampler attached to the current game, and that sampler preserves every cached visible value. Since $\mathsf{Q}_{\mathsf{Corrupt}} \leq \mathsf{t} - 1$, each session always has an uncorrupted signer available as a posterior pivot. For each mask profile, as long as fewer than t coordinates have been materialized, the next opened coordinate has the correct uniform marginal and the final latent coordinate is set by the Lagrange zero-sum equation. Hence a corruption that makes $\mathsf{Touch}[\mathsf{sinf}]$ reach t is not itself a bad event. The touched set counts visibility, while a still-latent decomposition can absorb the zero-sum constraint.

Definition 13 (Honest and malicious transcript coordinates). Fix an execution prefix and a second-round transcript

$$\mathsf{sinf} = (\mathsf{vk}_{\Sigma}, \tau, \mathsf{id}, \mathsf{S}, \mathsf{m}), \qquad \tau = \{\widetilde{\mathsf{m}}_i\}_{i \in \mathsf{S}}, \qquad \mathsf{id} = (\mathsf{sid}, \mathsf{rid}).$$

Define

$$\mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}] := \{i \in \mathsf{S} : \mathsf{m}_{i,\mathsf{id}} \neq \perp \text{ and } \widetilde{\mathsf{m}}_i = \mathsf{m}_{i,\mathsf{id}}\},$$
$$\mathsf{S}^{\mathsf{Mal}}[\mathsf{sinf}] := \mathsf{S} \setminus \mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}].$$

Thus $\mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}]$ records transcript coordinates that equal the already fixed honest first-round message for $(\mathsf{id}, \mathsf{S}, \mathsf{m})$, whether or not the signer has later been corrupted. The set $\mathsf{S}^{\mathsf{Mal}}[\mathsf{sinf}]$ records the remaining coordinates, which are treated as adversarially supplied for this transcript.

Theorem 3 (aTS-sUF bound). *Assume that* LFF *is lossiness-admitting with lossy-soundness error* ε_{l}*, and that* Com *is an* $(\varepsilon_{\mathsf{b}}(\lambda, \cdot), \varepsilon_{\mathsf{g}}, \varepsilon_{\mathsf{t}})$*-special trapdoor commitment scheme with targetable opening for* LFF*. Let* $\varepsilon_{\mathsf{b}}(\lambda, Q)$ *denote the maximum* Q*-key adaptive statistical binding error from definition 10. We denote by* $\varepsilon_{\mathsf{ki}}(\lambda, Q)$ *the maximum* Q*-key indistinguishability advantage from definition 9, and let* $\varepsilon_{\mathsf{key}}(\lambda, Q)$ *denote the maximum* Q*-key commitment-key distinguishing advantage from definition 10.*

Fix polynomially bounded $\mathsf{Q}_{\mathsf{Session}}$*,* $\mathsf{Q}_{\mathsf{Sig}_0}$*,* $\mathsf{Q}_{\mathsf{Sig}_1}$*,* Q_{H}*, and* Q_{M}*, a corruption bound* $\mathsf{Q}_{\mathsf{Cor}} \leq \mathsf{t} - 1$*, and a* PPT *adversary* $\mathcal{A}$ *that makes at most* $\mathsf{Q}_{\mathsf{Session}}$ *key-generation queries, at most* $\mathsf{Q}_{\mathsf{Cor}}$ *corruption queries, at most* $\mathsf{Q}_{\mathsf{Sig}_0}$ *legal first-round signing queries, at most* $\mathsf{Q}_{\mathsf{Sig}_1}$ *legal second-round signing queries, at most* Q_{H} *direct queries in total to commitment-key points of* H *and to* $\mathsf{H}_{\mathbf{H}}$*, and at most* Q_{M} *public/direct adversarial queries to all mask-related random oracles* $\mathsf{H}^{\mathsf{type}}_{\Delta}$*, not counting the signing oracles' internal evaluations. Let*

$$Q_{\mathsf{reg}} := \mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1.$$

Then

$$\mathsf{Adv}^{\mathsf{aTS\text{-}sUF}}_{\mathsf{Pilaf},\mathcal{A}}(1^\lambda) \le \frac{\mathsf{Q_M}}{2^\lambda} + 8Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + 2\mathsf{Q}_{\mathsf{Sig}_1}\varepsilon_{\mathsf{t}} + \frac{\mathsf{Q}_{\mathsf{Session}}Q_{\mathsf{H}}}{\#\left(\mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)\right)} + \varepsilon_{\mathsf{ki}}\left(\lambda, \mathsf{Q}_{\mathsf{Session}}\right) + 2 \cdot \frac{\mathsf{Q}_{\mathsf{Session}}\#\left(\mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)\right)}{\#\left(\mathsf{CD}_{\mathsf{LFF}}\right)} + 8\varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right) + 4\varepsilon_{\mathsf{b}}\left(\lambda, Q_{\mathsf{reg}}\right) + 2\left(Q_{\mathsf{H}} + \mathsf{Q}_{\mathsf{Sig}_1} + 1\right)\varepsilon_{\mathsf{I}}.$$

The proof is given below and uses the game-annotated pseudocode and supporting lemmas in section C.

6.1 Proof of the Adaptive Security Bound

Proof (Proof of theorem 3). Fix any adversary $\mathcal{A}$ covered by the theorem statement. We analyze a sequence of hybrid games and write pr_i for the winning probability in $\mathbf{G}_i$.

Game $\mathbf{G}_0$: This is the original aTS-sUF security game from fig. 5. Hence

$$\mathsf{pr}_0 = \mathsf{Adv}^{\mathsf{aTS\text{-}sUF}}_{\mathsf{Pilaf},\mathcal{A}}\left(1^\lambda\right).$$

Game $\mathbf{G}_1$: Masking abstraction. The challenger replaces the real pairwise-seed implementation of every mask-related random oracle by the ideal functionality $\mathcal{F}_\Delta$ and its ROM simulator from theorem 1. Direct malformed or non-profile mask queries remain ordinary lazy random-oracle queries. The simulator reuses its proof-only opening state for later corruptions, while the corruption oracle keeps the same external return format as the original experiment by the random-tape/state equivalence of lemma 11. After this hop, the hybrid challenger uses the same $\mathcal{S}_{\mathrm{ROM}}$ instance for public mask random-oracle answers, for the seed tuples embedded in signer secret keys, and for corruption openings of mask state. Thus a seed tuple revealed by corruption is exactly the tuple whose ordered-pair seeds are used by $\mathcal{S}_{\mathrm{ROM}}.\mathsf{H}$ to recognize distinguished random-oracle points. By theorem 1,

$$|\mathsf{pr}_1 - \mathsf{pr}_0| \le \frac{\mathsf{Q_M}}{2^\lambda}.$$

Game $\mathbf{G}_2$: Registered commitment-key aborts. From this game onward, calls to H and $\mathsf{H}_{\mathbf{H}}$ are answered from explicit lazy tables T_{H} and $\mathsf{T}_{\mathsf{H}_{\mathbf{H}}}$. A fresh unregistered H entry is sampled from $\mathcal{K}^4_{\mathsf{LFF}}$, and a fresh $\mathsf{H}_{\mathbf{H}}$ entry is sampled from $\mathsf{S}_{\mathsf{LFF}}$. The challenger also changes the handling of key-generation registrations. When a key-generation query registers a session public key $\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}] = \mathsf{F}\left(x[\mathsf{sid}]\right)$, the challenger now aborts with $\mathsf{Bad}_{\mathsf{preH}}$ if some row $\mathsf{H}\left(\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}], (\mathsf{sid}, \mathsf{rid})\right)$ was already initialized by the ordinary lazy table, or if a direct $\mathsf{H}_{\mathbf{H}}$ query already used an input parsing with the same $(\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}], (\mathsf{sid}, \mathsf{rid}))$. Since every fresh registered

$\mathsf{vk}_{\Sigma}[\mathsf{sid}]$ is uniform over $\mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$ before it is revealed, and the explicit tables are only a lazy implementation of the same random oracles, a union bound gives

$$|\mathsf{pr}_2 - \mathsf{pr}_1| \leq \frac{\mathsf{Q}_{\mathsf{Session}} Q_{\mathsf{H}}}{\#(\mathsf{F}(\mathsf{D}_{\mathsf{LFF}}))}.$$

Game $\mathbf{G}_3$: Simulation-openable registered keys. Every fresh registered row $\mathsf{H}(\mathsf{vk}_{\Sigma}[\mathsf{sid}], \mathsf{id})$ now samples a hidden branch bit b_{id}, two simulation-openable X-commitment keys, and two simulation-openable R-commitment keys together with their trapdoors. A direct $\mathsf{H}_{\mathbf{H}}$ query whose input names a registered pair $(\mathsf{vk}_{\Sigma}[\mathsf{sid}], \mathsf{id})$ silently initializes this row before the $\mathsf{H}_{\mathbf{H}}$ answer is sampled, if the row is still unset. Future proof-only completions of latent X and R commitments under these rows are materialized with TCom when the corresponding opening or trace is first needed. Public first-round commitment strings are cached and are never recomputed. By lemma 12,

$$|\mathsf{pr}_3 - \mathsf{pr}_2| \leq 4\left(\mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1\right)\varepsilon_{\mathsf{g}} + 4\varepsilon_{\mathsf{key}}(\lambda, Q_{\mathsf{reg}}).$$

Game $\mathbf{G}_4$: Touch bookkeeping. The challenger additionally maintains $\mathcal{Q}_2[\mathsf{sinf}]$ and $\mathsf{Touch}[\mathsf{sinf}]$ for every second-round transcript and marks legal queries as non-last-touch or last-touch. It does not change any answer, so

$$|\mathsf{pr}_4 - \mathsf{pr}_3| = 0.$$

Game $\mathbf{G}_5$: Missing honest last-touch coordinates. The next hop aborts with $\mathsf{BadHonTouch}$ if a legal last-touch query has an honest non-current coordinate that is not already touched by either a previous second-round answer or corruption. By lemma 13,

$$|\mathsf{pr}_5 - \mathsf{pr}_4| = 0.$$

Game $\mathbf{G}_6$: Uncorrupted malicious last-touch coordinates. A further abort $\mathsf{BadMalTouch}$ is added for legal last-touch queries with a malicious non-current coordinate whose signer is not already corrupted. By lemma 13,

$$|\mathsf{pr}_6 - \mathsf{pr}_5| = 0.$$

Game $\mathbf{G}_7$: Lazy first-round completion records. Every registered identifier now carries a completion record containing the registered row, hidden branch state, simulation trapdoors, and latent local first-round coordinates. Coordinates are opened only from their conditional affine and masking law using the samplers of section C.1. Last-touch rows use the joint masked local-state posterior of lemma 10 if they are corrupted later. By lemma 14,

$$|\mathsf{pr}_7 - \mathsf{pr}_6| = 0.$$

Game $\mathbf{G}_8$: Direct first-round posterior sampler. First-round signing queries now sample only the visible masked branch commitments from the same completion record. The unmasked preimage, randomness, commitment openings, and trace coordinates remain latent until a second-round answer or corruption opens them. By lemma 14,

$$|\mathsf{pr}_8 - \mathsf{pr}_7| = 0.$$

Game $\mathbf{G}_9$: Aggregate last-touch signing. A last-touch second-round answer is produced for the honest-transcript residual first. Malicious coordinates in the supplied first-round transcript are treated as adversarial and are not completed by the signing oracle. At last touch these malicious coordinates are already corrupted. Their long-term share contributions are subtracted from $\mathsf{vk}_{\Sigma}[\mathsf{sid}]$, and their opened mask coordinates determine the honest aggregate mask offsets. The hidden aggregate X branch b_{id} is opened in source mode to an image value. The complementary X branch $1 - b_{\mathsf{id}}$ absorbs the residual to $\mathsf{vk}_{\Sigma}[\mathsf{sid}]$ and is target-opened using $\mathsf{tk}^{X}_{\mathsf{id},1-b_{\mathsf{id}}}$. For the branch $1-b_{\mathsf{id}}$, the still-latent aggregate R trace is then materialized with the source representative kept by the proof and opened to the challenge-dependent represented target using the simulation trapdoor. The hidden branch b_{id} is answered by an ordinary source opening and the response equation $\rho = z - a\mathsf{h}$, so it uses no targetable trapdoor. The current signer's masked second-round values are the unique Lagrange residuals over $\mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}]$, with the honest aggregate mask offsets added. The current signer's local state is not sampled on this path before the share is returned. Non-last-touch queries are answered as in $\mathbf{G}_8$. By lemma 15,

$$|\mathsf{pr}_9 - \mathsf{pr}_8| \leq 2\mathsf{Q}_{\mathsf{Sig}_1}\varepsilon_{\mathsf{t}}.$$

Game $\mathbf{G}_{10}$: Uniform public keys and lazy long-term shares. Key generation now samples each session public key directly from $\mathsf{CD}_{\mathsf{LFF}}$ and opens long-term shares lazily on corruption. The preceding branch-specific X-key invariant is part of this hop. The hidden X branch remains an image point, and the complementary X branch absorbs the residual to the uniform public key without changing any public commitment string. By lemma 16,

$$|\mathsf{pr}_{10} - \mathsf{pr}_9| \leq \varepsilon_{\mathsf{ki}}\left(\lambda, \mathsf{Q}_{\mathsf{Session}}\right).$$

Game $\mathbf{G}_{11}$: Trapdoor LFF parameter generation. In this game, the setup algorithm samples $(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{td}_{\mathsf{im}}) \xleftarrow{\$} \mathsf{Setup}_{\mathsf{im}}\left(1^{\lambda}\right)$ and keeps $\mathsf{td}_{\mathsf{im}}$ hidden. Since the exposed $\mathsf{pp}_{\mathsf{LFF}}$ marginal is unchanged and $\mathsf{td}_{\mathsf{im}}$ is not read in this game,

$$|\mathsf{pr}_{11} - \mathsf{pr}_{10}| = 0.$$

Game $\mathbf{G}_{12}$: Abort on an image branch in the forgery. After parsing a fresh final forgery, the challenger initializes the corresponding registered row if

final verification would do so and aborts with $\mathsf{Bad}_{\mathsf{Forg}}$ if $\mathsf{InImage}\left(\mathsf{td}_{\mathsf{im}}, X^{\star}_{b_{\mathsf{id}^\star}}\right) = 1$. By lemma 17,

$$\mathsf{pr}_{11} - \mathsf{pr}_{12} \leq \frac{\mathsf{pr}_{11}}{2} + \frac{\mathsf{Q}_{\mathsf{Session}} \# \left(\mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)\right)}{\#\left(\mathsf{CD}_{\mathsf{LFF}}\right)}.$$

Let

$$Q_{\mathsf{reg}} := \mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1.$$

Game $\mathbf{G}_{13}$: Binding X key. The hidden-branch X-commitment key $\mathsf{ck}^{X}_{\mathsf{id},b_{\mathsf{id}}}$ in every registered row initialized before the final verification decision is sampled by BGen instead of as the public component of SGen. The complementary branch $\mathsf{ck}^{X}_{\mathsf{id},1-b_{\mathsf{id}}}$ remains simulation-openable, so later last-touch signing does not require guessing the final row. By lemma 18,

$$\left|\mathsf{pr}_{13} - \mathsf{pr}_{12}\right| \leq Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + \varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right).$$

Game $\mathbf{G}_{14}$: Binding R key on the hidden forged branch. The hidden-branch R-commitment key $\mathsf{ck}^{R}_{\mathsf{id},b_{\mathsf{id}}}$ in every registered row initialized before the final verification decision is sampled by BGen instead of as the public component of SGen. By lemma 18,

$$\left|\mathsf{pr}_{14} - \mathsf{pr}_{13}\right| \leq Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + \varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right).$$

The endpoint bound after this switch is the final claim of lemma 18.

Combining the displayed bounds through $\mathbf{G}_{11}$, the forged-branch image-abort bound, the two binding-key switches, and the endpoint lemma gives

$$\begin{aligned}\mathsf{Adv}^{\mathsf{aTS\text{-}sUF}}_{\mathsf{Pilaf},\mathcal{A}}\left(1^{\lambda}\right) = \mathsf{pr}_0 \leq \frac{\mathsf{Q}_{\mathsf{M}}}{2^{\lambda}} + 8\left(\mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1\right)\varepsilon_{\mathsf{g}} + 2\mathsf{Q}_{\mathsf{Sig}_1}\varepsilon_{\mathsf{t}} + \frac{\mathsf{Q}_{\mathsf{Session}} Q_{\mathsf{H}}}{\#\left(\mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)\right)} \\ + \varepsilon_{\mathsf{ki}}\left(\lambda, \mathsf{Q}_{\mathsf{Session}}\right) + 2 \cdot \frac{\mathsf{Q}_{\mathsf{Session}} \# \left(\mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)\right)}{\#\left(\mathsf{CD}_{\mathsf{LFF}}\right)} \\ + 8\varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right) + 4\varepsilon_{\mathsf{b}}\left(\lambda, Q_{\mathsf{reg}}\right) + 2\left(Q_{\mathsf{H}} + \mathsf{Q}_{\mathsf{Sig}_1} + 1\right)\varepsilon_{\mathsf{I}}.\end{aligned}$$

□

7 MDDH Instantiation of Dual-Mode Commitment with Targetable Opening

This section gives the MDDH instantiation behind the dual-mode commitment with targetable opening used by Pilaf. The LFF itself is instantiated from standard DDH. The construction realizes the targetable-opening interface of definition 10. The simulation-key mode is computationally indistinguishable from the binding mode under MDDH, as stated below.

The important point for the adaptive proof is that the equivocation algorithm is given proof-only representatives. It is not required to extract a scalar representation from an arbitrary group element. Instead, whenever the proof samples a source or target group element that may later be used in a trapdoor opening, it also keeps a representative under the full-rank commitment basis. The honest protocol uses only Com, so these representatives are never part of the public signature syntax.

7.1 DDH Linear Function Family

The LFF used by Pilaf is the standard DDH subset-membership family, in a k-fold product form. Let $\mathbb{G}$ be a cyclic group of prime order q and let $\mathsf{pp}_{\mathbb{G}} = \left(\mathbb{G}, q, \left[\cdot\right], k\right)$. The setup algorithm samples $\alpha_1, \ldots, \alpha_k \hookleftarrow \mathsf{U}\left(\mathbb{Z}_q\right)$ and outputs

$$\mathsf{pp}_{\mathsf{LFF}} = \left(\mathsf{pp}_{\mathbb{G}}, \left[\alpha_1\right], \ldots, \left[\alpha_k\right]\right).$$

The scalar field is $\mathsf{S}_{\mathsf{LFF}} = \mathbb{Z}_q$, the domain is $\mathsf{D}_{\mathsf{LFF}} = \mathbb{Z}_q^k$, and the codomain is $\mathsf{CD}_{\mathsf{LFF}} = \mathbb{G}^{2k}$. For $\mathbf{w} = \left(w_1, \ldots, w_k\right)^\top \in \mathbb{Z}_q^k$, define

$$\mathsf{F}\left(\mathbf{w}\right) := \left(\left[w_1\right], w_1 \cdot \left[\alpha_1\right], \ldots, \left[w_k\right], w_k \cdot \left[\alpha_k\right]\right).$$

Equivalently, $\mathsf{F}\left(\mathbf{w}\right) = \left[\bar{\mathbf{A}}_{\mathsf{DDH}}\right]\mathbf{w}$, where $\bar{\mathbf{A}}_{\mathsf{DDH}} \in \mathbb{Z}_q^{2k\times k}$ is the block-diagonal matrix whose jth column has entries 1 and α_j in rows $2j-1$ and $2j$.

The image-membership setup samples the same public parameters and keeps $\mathsf{td}_{\mathsf{im}} = \left(\alpha_1, \ldots, \alpha_k\right)$. On input $X = \left(X_{j,0}, X_{j,1}\right)_{j\in[k]} \in \mathbb{G}^{2k}$, the predicate $\mathsf{InImage}$ returns 1 iff

$$X_{j,1} = \alpha_j \cdot X_{j,0} \qquad \text{for every } j \in [k].$$

Lemma 3 (DDH LFF instantiation). *The family above is linear and $1/q$-lossiness admitting. For every polynomially bounded Q and every PPT adversary $\mathcal{A}$, its Q-key-indistinguishability advantage is bounded by the corresponding kQ-instance standard DDH advantage in $\mathbb{G}$.*

We give the proof in section D.

7.2 Commitment Construction

For commitments, we use the column-wise matrix extension of section 7.1. Thus a representative $\mathbf{W} \in \mathbb{Z}_q^{k\times k}$ is a tuple of k DDH-domain vectors, and $\mathsf{F}\left(\mathbf{W}\right)$ applies the DDH LFF to each column. Let $\bar{\mathbf{A}} \in \mathbb{Z}_q^{2k\times k}$ be the DDH matrix from section 7.1. The commitment-compatible setup samples an independent extension $\mathbf{A}_\perp \hookleftarrow \mathsf{U}\left(\mathbb{Z}_q^{2k\times k}\right)$, sets $\mathbf{A} := \left(\bar{\mathbf{A}} \mid \mathbf{A}_\perp\right)$, and samples $\mathbf{Y} \hookleftarrow \mathsf{U}\left(\mathbb{Z}_q^{4k\times 2k}\right)$. The public parameter $\mathsf{pp}_{\mathsf{LFF}}$ is the DDH LFF parameter together with $\left[\mathbf{A}\right]$ and $\left[\mathbf{Y}\right]$. The image-membership setup samples the same auxiliary public data with the

same marginal distribution and keeps only the DDH image-membership trapdoor. Below, BGen and SGen receive this augmented $\mathsf{pp}_{\mathsf{LFF}}$, and

$$\mathsf{F}(\mathbf{W}) := \left[\bar{\mathbf{A}}\right] \mathbf{W}.$$

The commitment key, opening, trace, and commitment spaces are

$$\begin{aligned}
\mathcal{K}_{\mathsf{LFF}} &:= \left\{\left(\left[\mathbf{A}\right], \left[\mathbf{Y}\right], \left[\mathbf{Z}\right]\right) : \mathbf{Z} \in \mathbb{Z}_q^{4k\times 2k}\right\},\\
\mathcal{O}_{\mathsf{LFF}} &:= \mathbb{Z}_q^{2k\times k} \times \mathbb{Z}_q^{2k\times k},\\
\mathcal{R}_{\mathsf{LFF}} &:= \mathbb{Z}_q^{2k\times k},\\
\mathcal{T}_{\mathsf{LFF}} &:= \mathbb{Z}_q^{2k\times k},\\
\mathcal{C}_{\mathsf{LFF}} &:= \mathbb{G}^{2k\times k} \times \mathbb{G}^{4k\times k}.
\end{aligned}$$

For $\mathbf{W} \in \mathbb{Z}_q^{k\times k}$, write

$$\widehat{\mathbf{W}} := \left(\begin{smallmatrix}\mathbf{W}\\ \mathbf{0}\end{smallmatrix}\right) \in \mathbb{Z}_q^{2k\times k}.$$

The representation map and the image embedding are

$$\mathsf{Rep}_{\mathsf{LFF}}(\mathbf{D}) := \left[\mathbf{A}\right] \mathbf{D}, \qquad \mathsf{Emb}_{\mathsf{LFF}}(\mathbf{W}) := \widehat{\mathbf{W}}.$$

Then $\mathsf{Rep}_{\mathsf{LFF}}\left(\widehat{\mathbf{W}}\right) = \mathsf{F}(\mathbf{W})$. When $\operatorname{rank}(\mathbf{A}) = 2k$, every element of $\mathsf{CD}_{\mathsf{LFF}} = \mathbb{G}^{2k\times k}$ has such a representative. On this event, the proof-only representative sampler chooses $\mathbf{D} \hookleftarrow \mathsf{U}\left(\mathbb{Z}_q^{2k\times k}\right)$, sets $X := \mathsf{Rep}_{\mathsf{LFF}}(\mathbf{D})$, and outputs $(\mathbf{D}, X)$. Since $\mathbf{A}$ has full row rank, the marginal distribution of X is uniform over $\mathsf{CD}_{\mathsf{LFF}}$.

The algorithms are given in fig. 8. The targetable-opening algorithm is defined for good simulation keys, where $\operatorname{rank}(\mathbf{A}) = 2k$, $\operatorname{rank}(\mathbf{Y}) = 2k$, and $\operatorname{rank}(\mathbf{T}) = 2k$.

7.3 Properties

We state the MDDH commitment properties here and give their proofs in section D.

Lemma 4 (Homomorphism and targetable opening). *The construction is linearly homomorphic. Moreover, for every good simulation key, every represented source* $(\mathsf{c}, \tau) = \mathsf{TCom}\left(\mathsf{ck}, \mathsf{tk}, \left[\mathbf{R}\right], \mathbf{D}; \mathbf{U}, \mathbf{V}\right)$*, every target representative* $\mathbf{D}_X \in \mathcal{R}_{\mathsf{LFF}}$*, every* $c \in \mathbb{Z}_q$*, and every* $\rho \in \mathbb{Z}_q^{k\times k}$*, the opening*

$$\omega' := \mathsf{TCol}\left(\mathsf{ck}, \mathsf{tk}, \mathsf{c}, \tau, (\mathbf{U}, \mathbf{V}), \mathbf{D}_X, c, \rho\right)$$

is the unique element of the posterior fiber

$$\{\bar{\omega} \in \mathcal{O}_{\mathsf{LFF}} : \mathsf{Com}\left(\mathsf{ck}, \mathsf{F}(\rho) + \mathsf{Rep}_{\mathsf{LFF}}(\mathbf{D}_X)\, c; \bar{\omega}\right) = \mathsf{c}\}.$$

Thus the targetable-opening error is 0.

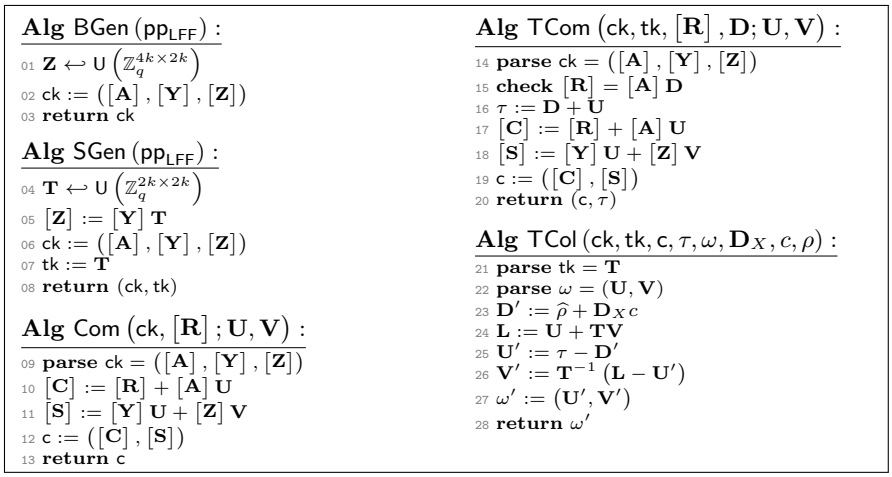


Fig. 8. MDDH dual-mode commitment algorithms.

Lemma 5 (Bad-key and binding bounds). *A simulation key is bad with probability at most $5k/q$. For Q independent binding keys with common $\mathsf{pp}_{\mathsf{LFF}}$, the adaptive Q-key statistical binding error is $\varepsilon_{\mathsf{b}}(\lambda, Q) = 4kQ/q$.*

Lemma 6 (Dual-mode key indistinguishability). *For every $Q \in \mathbb{N}$ and every* PPT *adversary $\mathcal{A}$, there is a* PPT *adversary $\mathcal{A}_{\mathsf{mode}}$ such that*

$$\mathsf{Adv}^{\mathsf{mode}}_{\mathcal{A}}\left(1^{\lambda}, Q\right) \leq \mathsf{Adv}^{2kQ\text{-}\mathcal{D}_{2k,4k}\text{-MDDH}}_{\mathcal{A}_{\mathsf{mode}},\mathbb{G}}\left(1^{\lambda}\right),$$

where the left-hand side distinguishes Q binding keys from Q simulation keys.

Theorem 4 (MDDH commitment instantiation). *The construction above is a linearly homomorphic dual-mode commitment with targetable opening for the column-wise DDH LFF of section 7.1. Its binding keys are uniform, its targetable opening is perfect on good simulation keys, the bad-key probability is at most $5k/q$, and the Q-key binding error is $\varepsilon_{\mathsf{b}}(\lambda, Q) = 4kQ/q$. The binding and simulation key modes are indistinguishable under the $2kQ$-instance $(2k, 4k)$-*MDDH *assumption.*

8 Efficiency

Pilaf is partially non-interactive, since its first-round messages are independent of the message and can therefore be preprocessed offline. The concrete counts below use the MDDH commitment instantiation of section 7. In this instantiation, one commitment has size $6k^2\mathbb{G}$, and one opening has size $4k^2\mathbb{Z}_p$. The dominant costs of Pilaf come from hiding the branch already in round 1, carrying both branches through the second-round share, and committing separately to both X_b and R_b on every branch.

We summarize the exact transcript sizes below. Here we count transmitted group elements and field elements directly from the construction algorithms for Pilaf. For example, a matrix in $\mathbb{G}^{2k\times k}$ contributes $2k^2$ group elements, and a matrix in $\mathbb{Z}_p^{2k\times k}$ contributes $2k^2$ field elements. For compactness in the table, $x\mathbb{G}$ means x transmitted group elements and $x\mathbb{Z}_p$ means x transmitted field elements.

Scheme	**Round-1** m_ℓ	**Round-2** σ_ℓ	**Literal** σ_Σ	**Final** σ_Σ
Pilaf	$24k^2\mathbb{G}$	$32k^2\mathbb{G}+18k^2\mathbb{Z}_p$	$32k^2\mathbb{G}+18k^2\mathbb{Z}_p$	$2k^2\mathbb{G}+\left(18k^2+1\right)\mathbb{Z}_p$

Table 2. Communication cost of Pilaf.

For Pilaf, the literal aggregate signature in fig. 7 is

$$\left\{X_b, R_b, \rho_b, \omega_b^X, \omega_b^R, \mathsf{x}_b, \mathsf{r}_b\right\}_{b\in\{0,1\}},$$

where each of x_b and r_b is one MDDH commitment. This literal form has size $32k^2\mathbb{G}+18k^2\mathbb{Z}_p$. Per branch, X_b and R_b contribute $4k^2\mathbb{G}$, the two commitments contribute $12k^2\mathbb{G}$, ρ_b contributes $k^2\mathbb{Z}_p$, and the two openings contribute $8k^2\mathbb{Z}_p$.

The final Pilaf signature can still omit all commitments and all R_b values by including the challenge h. It is enough to transmit

$$\left(X_0, \left\{\rho_b, \omega_b^X, \omega_b^R\right\}_{b\in\{0,1\}}, \mathsf{h}\right).$$

The verifier reconstructs $X_1 := \mathsf{vk}_\Sigma - X_0$, sets $R_b := \mathsf{F}(\rho_b) + X_b\mathsf{h}$, recomputes $\mathsf{x}_b := \mathsf{Com}\left(\mathsf{ck}_b^X, X_b; \omega_b^X\right)$ and $\mathsf{r}_b := \mathsf{Com}\left(\mathsf{ck}_b^R, R_b; \omega_b^R\right)$, and checks that

$$\mathsf{h} = \mathsf{H}_\mathbf{H}\left(\mathsf{vk}_\Sigma, \mathsf{x}_0, \mathsf{r}_0, \mathsf{x}_1, \mathsf{r}_1, \mathsf{id}, \mathsf{S}, \mathsf{m}\right).$$

Thus the compressed Pilaf signature has size $2k^2\mathbb{G}+\left(18k^2+1\right)\mathbb{Z}_p$. The extra $\mathbb{Z}_p$ term counts the scalar challenge $\mathsf{h}\in\mathsf{S}_{\mathsf{LFF}}$.

The adaptive-security machinery is reflected in the per-signer transcript and final opening sizes. Concretely, the first-round message has size $24k^2\mathbb{G}$, the literal second-round share has size $32k^2\mathbb{G}+18k^2\mathbb{Z}_p$, and the compressed final signature has size $2k^2\mathbb{G}+\left(18k^2+1\right)\mathbb{Z}_p$.

On the computation side, the Pilaf verifier performs one commitment-key hash query, one $\mathsf{H}_\mathbf{H}$ query, one aggregate-key check, two response checks, and four opening checks. For the signers, the work remains a constant-factor amount per branch, since each branch carries independent commitment and opening randomness for both X_b and R_b.

References

BC26. Renas Bacho and Yanbo Chen. Earpicks: Tightly secure two-round multi- and threshold signatures. *Eurocrypt*, 2026. p. 3, 4

BCK+22. Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 517–550. Springer, Cham, August 2022. p. 2, 3, 12

BDLR25a. Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. Adaptively secure three-round threshold Schnorr signatures from DDH. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 390–422. Springer, Cham, August 2025. p. 4

BDLR25b. Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. Glacius: Threshold schnorr signatures from DDH with full adaptive security. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 304–334. Springer, Cham, May 2025. p. 4

BGC+25. Ruben Baecker, Paul Gerhart, Davide Li Calsi, Luigi Russo, Dominique Schröder, and Arkady Yerukhimovich. Fully adaptive FROST in the algebraic group model from falsifiable assumptions. Cryptology ePrint Archive, Report 2025/1950, 2025. p. 3, 4

BLSW24. Renas Bacho, Julian Loss, Gilad Stern, and Benedikt Wagner. HARTS: High-threshold, adaptively secure, and robust threshold Schnorr signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part III*, volume 15486 of *LNCS*, pages 104–140. Springer, Singapore, December 2024. p. 3, 4

BLT+24. Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from DDH with full adaptive security. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 429–459. Springer, Cham, May 2024. p. 3, 4

BW24. Renas Bacho and Benedikt Wagner. Tightly secure threshold signatures over pairing-free groups. Cryptology ePrint Archive, Report 2024/1557, 2024. p. 3, 4

Che25a. Yanbo Chen. Dazzle: Improved adaptive threshold signatures from DDH. Cryptology ePrint Archive, Report 2025/264, 2025. p. 3, 4

Che25b. Yanbo Chen. Round-efficient adaptively secure threshold signatures with rewinding. *CiC*, 2(2):16, 2025. p. 12, 35

Che26. Yanbo Chen. Partially non-interactive two-round threshold and multi-signatures with tighter and adaptive security. *Eurocrypt*, 2026. p. 3, 4

CKM23. Elizabeth Crites, Chelsea Komlo, and Mary Maller. Fully adaptive Schnorr threshold signatures. Cryptology ePrint Archive, Report 2023/445, 2023. p. 3, 4

Des90. Yvo Desmedt. Abuses in cryptography and how to fight them. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 375–389. Springer, New York, August 1990. p. 2

DR24. Sourav Das and Ling Ren. Adaptively secure BLS threshold signatures from DDH and co-CDH. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 251–284. Springer, Cham, August 2024. p. 3, 4

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Berlin, Heidelberg, August 2013. p. 66

KG20. Chelsea Komlo and Ian Goldberg. FROST: Flexible round-optimized Schnorr threshold signatures. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O'Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 34–65. Springer, Cham, October 2020. p. 3, 4

KRT24. Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 459–491. Springer, Cham, August 2024. p. 12, 35

NRT25. Guilhem Niot, Michael Reichle, and Kaoru Takemure. Adaptively-secure three-round threshold Schnorr from DL. Cryptology ePrint Archive, Report 2025/1941, 2025. p. 4

NT24. Sela Navot and Stefano Tessaro. One-more unforgeability for multi - and threshold signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 429–462. Springer, Singapore, December 2024. p. 2

PW23. Jiaxin Pan and Benedikt Wagner. Chopsticks: Fork-free two-round multi-signatures from non-interactive assumptions. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 597–627. Springer, Cham, April 2023. p. 15

TJZ25. Shaolong Tang, Peng Jiang, and Liehuang Zhu. Glitter: A fully adaptive and tightly secure threshold signature. In Willy Susilo and Josef Pieprzyk, editors, *ACISP 25, Part I*, volume 15658 of *LNCS*, pages 184–204. Springer, Singapore, July 2025. p. 3, 4

A ROM Realization of Profile-Wise Zero-Sum Masking

A.1 Proofs for Profile-Wise Masking

Proof (Proof of lemma 1). Fix a proper subset $\mathsf{T} \subsetneq \mathsf{S}$ and values $(z_i)_{i\in\mathsf{T}}$. Choose any pivot $p \in \mathsf{S}\setminus\mathsf{T}$. For every assignment of the non-pivot remaining coordinates

$$(y_i)_{i\in\mathsf{S}\setminus(\mathsf{T}\cup\{p\})} \in V^{\mathsf{S}\setminus(\mathsf{T}\cup\{p\})},$$

there is a unique pivot value

$$y_p = \lambda_{\mathsf{S},p}^{-1}\left(A - \sum_{i\in\mathsf{T}} \lambda_{\mathsf{S},i} z_i - \sum_{i\in\mathsf{S}\setminus(\mathsf{T}\cup\{p\})} \lambda_{\mathsf{S},i} y_i\right).$$

Hence the number of completions is $\#(V)^{\mathsf{t}-\#(\mathsf{T})-1}$, independent of $(z_i)_{i\in\mathsf{T}}$. The projection to T is therefore uniform, and the displayed sampling rule is exactly the conditional law on the fiber. □

Proof (Proof of lemma 2). Expanding the definition of $\Delta_{\mathrm{ROM}}.\mathsf{Eval}^{\mathsf{H}_\Delta}$ gives

$$\begin{aligned}\sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i}\Delta_{\mathsf{S},i}^{\mathsf{type},\mathsf{H}_\Delta}(\mathsf{ctx}) &= \sum_{i\in\mathsf{S}}\sum_{j\in\mathsf{S}\setminus\{i\}} \left(\mathsf{H}_\Delta^{\mathsf{type}}(\mathsf{ctx},i,j,\mathsf{s}_{i,j}) - \mathsf{H}_\Delta^{\mathsf{type}}(\mathsf{ctx},j,i,\mathsf{s}_{j,i})\right)\\ &= 0.\end{aligned}$$

The last equality holds because each ordered-pair term appears once with coefficient $+1$ and once with coefficient -1. □

Lemma 7 (Edge-table image and uniform zero-sum masks). *Let V be a finite $\mathbb{F}$-vector space, let S be a finite set of distinct nonzero field elements, and let*

$$E_\pi = \{(x,y) \in \mathsf{S}^2 : x \neq y\}.$$

For $T \in V^{E_\pi}$ define

$$\Delta_a(T) = \sum_{j\in\mathsf{S}\setminus\{a\}} T[a,j] - \sum_{j\in\mathsf{S}\setminus\{a\}} T[j,a].$$

Let

$$B : V^{E_\pi} \to V^{\mathsf{S}}, \qquad (BT)_a = \lambda_{\mathsf{S},a}^{-1}\Delta_a(T).$$

Then

$$\mathrm{Im}(B) = \left\{ m \in V^{\mathsf{S}} : \sum_{a\in\mathsf{S}} \lambda_{\mathsf{S},a} m_a = 0 \right\}.$$

Consequently, if $T \hookleftarrow \mathsf{U}\left(V^{E_\pi}\right)$, then $B(T)$ is uniform over this zero-sum subspace.

Proof. For every T,

$$\sum_{a\in\mathsf{S}} \lambda_{\mathsf{S},a}\,(BT)_a = \sum_{a\in\mathsf{S}} \Delta_a(T) = 0,$$

because every directed-edge entry occurs once with coefficient $+1$ and once with coefficient -1. This proves the containment from left to right.

Conversely, fix $m \in V^{\mathsf{S}}$ with $\sum_{a\in\mathsf{S}} \lambda_{\mathsf{S},a} m_a = 0$, and choose any pivot $p \in \mathsf{S}$. Set $T[i,p] = \lambda_{\mathsf{S},i} m_i$ for each $i \in \mathsf{S}\setminus\{p\}$, and set all other directed-edge entries to 0. For $i \neq p$ this gives $\Delta_i(T) = \lambda_{\mathsf{S},i} m_i$. For the pivot,

$$\Delta_p(T) = -\sum_{i\in\mathsf{S}\setminus\{p\}} \lambda_{\mathsf{S},i} m_i = \lambda_{\mathsf{S},p} m_p.$$

Hence $B(T) = m$, proving surjectivity onto the zero-sum subspace. Since B is a linear map between finite vector spaces and is onto this subspace, all of its nonempty fibers have the same size. Therefore the image of a uniform edge table is uniform over the zero-sum subspace. □

Lemma 8 (Directed-edge posterior sampling under admissible packing). *Fix a legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$, write $V = \mathsf{D}(\mathsf{type})$, and let*

$$E_\pi = \big\{(x,y) \in \mathsf{S}^2 : x \neq y\big\}.$$

Let $T \hookleftarrow \mathsf{U}\left(V^{E_\pi}\right)$ and define

$$\Delta_a(T) = \sum_{j\in\mathsf{S}\setminus\{a\}} T[a,j] - \sum_{j\in\mathsf{S}\setminus\{a\}} T[j,a].$$

Let $A \subseteq \mathsf{S}$ be the parties already packed for π. Suppose a partial table is distributed as the uniform table T conditioned on the edge entries already returned and on the equations $\Delta_a(T) = \lambda_{\mathsf{S},a} m_a$ for all $a \in A$. Suppose moreover that, for every $a \in A$, all entries in

$$\mathsf{Inc}_a = \{(a,j) : j \in \mathsf{S}\setminus\{a\}\} \cup \{(j,a) : j \in \mathsf{S}\setminus\{a\}\}$$

are already fixed in the partial table. When a new opened coordinate m_i with $i \notin A$ is added, define

$$U_i = \{(x,y) \in \mathsf{Inc}_i : T[x,y] \text{ is not fixed in the partial table}\}.$$

Call i admissible if either $U_i \neq \emptyset$, or $A = \mathsf{S}\setminus\{i\}$, the opened values including m_i satisfy $\sum_{a\in\mathsf{S}} \lambda_{\mathsf{S},a} m_a = 0$, and the equation for i is the final dependent equation forced by $\sum_{a\in\mathsf{S}} \Delta_a(T) = 0$. If i is admissible, the rule in $\mathcal{S}_{ROM}.\mathsf{Pack}_\pi$ samples the exact posterior distribution of the remaining edge entries conditioned additionally on $\Delta_i(T) = \lambda_{\mathsf{S},i} m_i$. The resulting partial table satisfies the same invariant with $A \cup \{i\}$.

Proof. Condition on the current partial edge table and on the current opened coordinate equations. By the invariant, every equation indexed by $a \in A$ involves only fixed table entries. Hence no previous opened coordinate equation constrains an unset entry in Inc_i. The equation $\Delta_i(T) = \lambda_{\mathsf{S},i} m_i$ is affine in the unset entries in Inc_i. Each unset incident entry has coefficient 1 or -1 in this equation. If $U_i \neq \emptyset$, choose one pivot entry $e^\star \in U_i$. The conditional law of independent uniform edge entries under this single affine equation is sampled by choosing all other unset incident entries uniformly in V and setting the pivot entry to the unique value satisfying the equation. This is exactly the update performed by $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Pack}_\pi$.

If $U_i = \emptyset$, admissibility gives $A = \mathsf{S} \setminus \{i\}$ and says that the equation for i is the final dependent equation forced by

$$\sum_{a \in \mathsf{S}} \Delta_a(T) = 0$$

together with $\sum_{a \in \mathsf{S}} \lambda_{\mathsf{S},a} m_a = 0$. Hence the already fixed incident entries necessarily satisfy $\Delta_i(T) = \lambda_{\mathsf{S},i} m_i$. The posterior conditioned additionally on this equation is the same point mass on the current table, and the check in $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Pack}_\pi$ enforces the equality. Thus each admissible packing step is an exact conditional sampler, fixes every entry in Inc_i, and preserves the invariant with $A \cup \{i\}$. □

Proof. The argument below is self-contained for the present profile-wise interface. It uses the delayed-programming technique of [KRT24,Che25b].

The simulator $\mathcal{S}_{\mathrm{ROM}}$ is specified in figs. 9 and 10. The superscript leak is proof notation for collecting leakage from an internal ideal-world call. The assignment $(r, L) := \mathcal{F}_\Delta.\mathsf{Init}^{\mathsf{leak}}\left(1^\lambda, \mathsf{N}, \mathsf{t}, \pi\right)$ means that the simulator runs the ordinary ideal procedure $\mathcal{F}_\Delta.\mathsf{Init}\left(1^\lambda, \mathsf{N}, \mathsf{t}, \pi\right)$, sets r to its return value, and sets L to the ordered transcript of the **leak** messages that this call sends to the simulator under section 4.1. It is not a new environment-facing oracle. For a leakage transcript L, let $\mathsf{CorLeak}_\pi(L)$ be the subsequence of $\mathsf{CorMask}$ messages in L for profile π. The simulator receives ideal leakage internally and emits the real-format leakage messages explicitly shown in the simulator figure. Seed tables, concrete seed tuples, corruption sets, and distinguished simulator tables are scoped by the session label $\mathsf{sid} = \mathsf{sid}(\mathsf{ctx})$. The simulator separates internal activation from environment-facing initialization. The flag $\mathsf{Act}_{\mathcal{S}}[\pi]$ records that the ordinary ideal initialization of π has been run inside the simulator and that the ideal coordinates and leakage transcript have been fixed. The flag $\mathsf{Init}_{\mathcal{S}}[\pi]$ records that the environment-facing initialization call for π has occurred. The internal procedure $\mathsf{Ensure}(\pi)$ may set only $\mathsf{Act}_{\mathcal{S}}[\pi]$. It emits no environment-facing leakage and it does not enable $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Mask}(\pi, i)$. The external procedure $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Init}$ first calls $\mathsf{Ensure}(\pi)$, then sets $\mathsf{Init}_{\mathcal{S}}[\pi]$ and emits the single real-format initialization leakage. Thus internal activation is only a lazy sampling device for the simulator's posterior table.

Fix an environment $\mathcal{Z}$ and write pr_i for the probability that $\mathcal{Z}$ outputs 1 in $\mathbf{G}_i$.



Fig. 9. Simulator $\mathcal{S}_{\mathrm{ROM}}$ for the ROM realization of profile-wise zero-sum masking. The environment-facing procedures are Setup, Init, H, Mask, and CorMask. The procedures EnsureSid, Ensure, Record, and RO are simulator-internal. The **leak** lines in the environment-facing procedures are exposed to the environment.

Game $\mathbf{G}_0$: Real pairwise-seed execution. This game is $\mathcal{E}^{\mathsf{real},\mathsf{H}_\Delta}_{\mathcal{R}_\Delta,\mathcal{Z}}\left(1^\lambda, \mathsf{N}, \mathsf{t}\right)$. Each activated session label sid owns an independent ordered-pair seed table $\{\mathsf{s}_{u,v}[\mathsf{sid}]\}_{u \neq v}$. The public typed random oracle is an ordinary lazy table. Legal mask openings run $\Delta_{\mathrm{ROM}}.\mathsf{Eval}^{\mathsf{H}_\Delta}$ with $\mathsf{s}_{i,\mathsf{sid}}$, and corruptions in sid return the

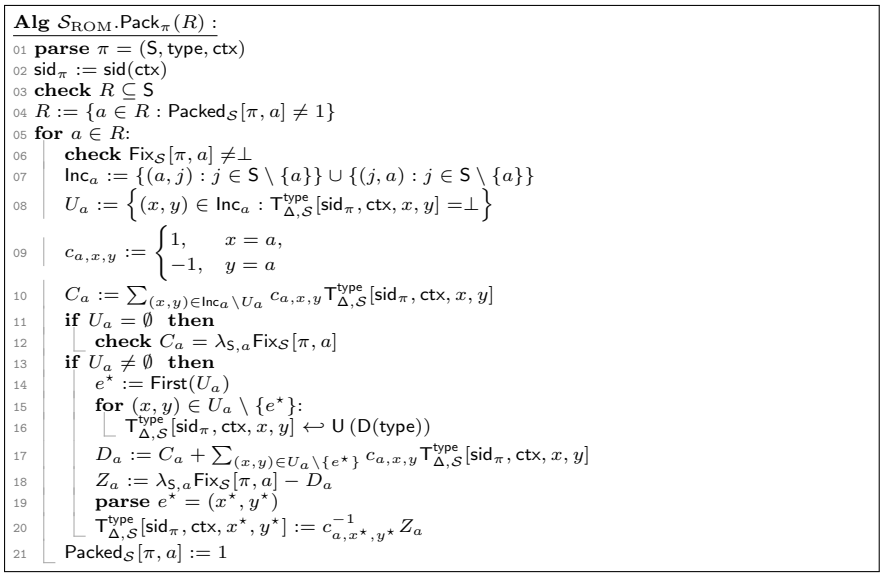


Fig. 10. Simulator-internal packing procedure for the ROM masking simulator. This procedure is not exposed to the environment.

concrete seed tuple $\mathsf{s}_{i,\mathsf{sid}}$. Thus

$$\mathsf{pr}_0 = \Pr\left[\mathcal{E}^{\mathsf{real},\mathsf{H}_\Delta}_{\mathcal{R}_\Delta,\mathcal{Z}}\left(1^\lambda, \mathsf{N}, \mathsf{t}\right) = 1\right].$$

Game $\mathbf{G}_1$: Distinguished table notation. This game exposes the part of the lazy random-oracle table that is relevant to pairwise masking. For every legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$, write $\mathsf{sid}_\pi = \mathsf{sid}(\mathsf{ctx})$. For every ordered pair $(u, v) \in \mathsf{S}^2$ with $u \neq v$, introduce a conceptual table entry $T_{\mathsf{sid}_\pi}[\pi, u, v]$ for the distinguished point $\mathsf{H}^{\mathsf{type}}_\Delta(\mathsf{ctx}, u, v, \mathsf{s}_{u,v}[\mathsf{sid}_\pi])$. A public query $\mathsf{H}^{\mathsf{type}}_\Delta(\mathsf{ctx}, u, v, s)$ whose type or context is malformed, whose endpoints do not parse as such an ordered pair, or whose seed input is different from $\mathsf{s}_{u,v}[\mathsf{sid}_\pi]$ is answered from the ordinary lazy random-oracle table. A query to a distinguished point returns $T_{\mathsf{sid}_\pi}[\pi, u, v]$, sampled uniformly if it has not been set. Calls to $\Delta_{\mathrm{ROM}}.\mathsf{Eval}^{\mathsf{H}_\Delta}$ use these same T entries for legal profiles. This only renames entries of the real lazy table, so

$$\mathsf{pr}_1 = \mathsf{pr}_0.$$

Game $\mathbf{G}_2$: Delayed pairwise seeds. This game treats the hidden ordered-pair seed tables independently for each sid. Since profile-level mask computations now refer directly to $T_{\mathsf{sid}}[\pi, u, v]$, they no longer need to reveal the concrete seed before an endpoint is opened. When an endpoint is corrupted in session sid, the game returns the concrete seed tuple $\mathsf{s}_{i,\mathsf{sid}}$ in the same format as $\mathcal{R}_\Delta$.

Define $\mathsf{Bad}_{\mathsf{seed}}$ as the event that the environment makes an explicit public query $\mathsf{H}^{\mathsf{type}}_\Delta(\mathsf{ctx}, u, v, s)$ before either endpoint is corrupted in the session $\mathsf{sid}(\mathsf{ctx})$

and later $s = \mathsf{s}_{u,v}[\mathsf{sid}(\mathsf{ctx})]$ for the hidden ordered-pair seed that makes the query distinguished. For each explicit public query, the parsed session and ordered pair determine at most one hidden seed that can make the point distinguished. Hence

$$\Pr\left[\mathsf{Bad}_{\mathsf{seed}}\right] \leq \frac{\mathsf{Q}_{\mathsf{M}}}{2^{\lambda}}.$$

Conditioned on $\neg\mathsf{Bad}_{\mathsf{seed}}$, no hidden distinguished point has already been returned as an ordinary random-oracle answer before the simulator needs to materialize it. Therefore

$$|\mathsf{pr}_2 - \mathsf{pr}_1| \leq \frac{\mathsf{Q}_{\mathsf{M}}}{2^{\lambda}}.$$

Game $\mathbf{G}_3$: Reverse profile-coordinate sampling. This game changes only the internal order in which mask coordinates and distinguished T entries are sampled. For a legal profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$ with $\mathsf{sid}_\pi = \mathsf{sid}(\mathsf{ctx})$, define

$$\delta_{\pi,i} = \sum_{j \in \mathsf{S}\setminus\{i\}} T_{\mathsf{sid}_\pi}[\pi, i, j] - \sum_{j \in \mathsf{S}\setminus\{i\}} T_{\mathsf{sid}_\pi}[\pi, j, i].$$

The real local mask is $\lambda_{\mathsf{S},i}^{-1}\delta_{\pi,i}$. Thus, when the ideal functionality fixes the coordinate $\mathsf{Fix}[\pi, i]$, the distinguished table must satisfy

$$\delta_{\pi,i} = \lambda_{\mathsf{S},i}\mathsf{Fix}[\pi, i].$$

In this game, when party i is opened for profile π, the game first fixes $\mathsf{Fix}[\pi, i]$ according to the lazy completion rule of $\mathcal{F}_{\Delta}$. It then materializes every still-hidden distinguished entry incident to i. If some incident entries are already fixed, their contribution is subtracted from the displayed equation. Among the remaining incident entries, the game samples all but one uniformly from $\mathsf{D}(\mathsf{type})$ and sets the last entry to the unique value satisfying the equation. This value is well defined because every incident entry has coefficient $+1$ or -1 in $\delta_{\pi,i}$, and $\lambda_{\mathsf{S},i} \neq 0$.

This reverse sampling has the same posterior law as the real table-based execution. Conditioned on $\neg\mathsf{Bad}_{\mathsf{seed}}$, the game maintains the following reachable-state invariant for each profile π. First, $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Pack}_\pi$ removes already packed coordinates from its input set before changing the table. Therefore repeated calls on a packed coordinate are idempotent and preserve all previously returned entries. Second, a hidden distinguished point

$$\mathsf{H}_{\Delta}^{\mathsf{type}}\left(\mathsf{ctx}, u, v, \mathsf{s}_{u,v}[\mathsf{sid}_\pi]\right)$$

is never returned as an ordinary public random-oracle answer before the simulator materializes it as a distinguished entry. Every materialized distinguished point that has already been returned to the environment is one of the visible edge entries in the conditioning of lemma 8. Third, whenever $\mathcal{S}_{\mathrm{ROM}}.\mathsf{Pack}_\pi$ is called on a coordinate that has not already been packed, that coordinate is admissible in the sense of lemma 8.

To see the third part, consider a new coordinate i. If an incident distinguished edge to another not-yet-packed coordinate remains unset, this edge can serve as the pivot. A public query cannot remove the last such pivot while both endpoints are uncorrupted, because guessing the hidden seed first is exactly $\mathsf{Bad}_{\mathsf{seed}}$. If a correct public query uses a seed whose endpoint has already been revealed, then the ideal corruption bookkeeping has already recorded that endpoint's profile coordinate, and Ensure and Record pack the endpoint before the public query returns its incident distinguished entry. Hence public queries through revealed seeds add only visible edge entries whose fixed endpoint is already packed. These visible entries are exactly the returned edge entries allowed in the posterior lemma. If all incident entries of i have been fixed by other packed coordinates, then every coordinate in $\mathsf{S} \setminus \{i\}$ has already been packed, so i is the final dependent coordinate forced by

$$\sum_{a \in \mathsf{S}} \Delta_a(T) = 0 \qquad \text{and} \qquad \sum_{a \in \mathsf{S}} \lambda_{\mathsf{S},a} \mathsf{Fix}_{\mathcal{S}}[\pi, a] = 0.$$

Thus every nontrivial packing call is admissible. After a party is packed, every distinguished entry incident to that party is fixed, so the invariant of lemma 8 holds for the set of packed parties. By the lemma, the packing rule samples the directed distinguished table from the exact posterior conditioned on all visible distinguished entries and opened coordinates. Hence the induced local-mask coordinates have the real posterior distribution given the same visible data.

It remains to identify this coordinate posterior with the lazy completion law of $\mathcal{F}_{\Delta}$. By lemma 7, independent uniform distinguished edge tables induce a uniform mask vector over the zero-sum subspace $\sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{Fix}[\pi, i] = 0$. Therefore the posterior after any proper subset of coordinates has been fixed is exactly the affine-fiber posterior of lemma 1: fewer than $\mathsf{t}-1$ fixed coordinates leave the next coordinate uniform in $\mathsf{D}(\mathsf{type})$, and the last coordinate is forced by the Lagrange zero-sum equation. In particular, cached coordinates and visible distinguished entries are never resampled. Thus

$$\mathsf{pr}_3 = \mathsf{pr}_2.$$

Game $\mathbf{G}_4$: Ideal corruption bookkeeping. This game introduces the ideal corruption-opening state. A corruption query for session sid still returns the concrete seed tuple $\mathsf{s}_{i,\mathsf{sid}}$ with the real distribution. Internally, the simulator also calls $\mathcal{S}.\mathsf{CorMask}$ to fix all initialized profile coordinates for the corrupted party, then materializes the corresponding incident distinguished entries using the reverse-sampling rule from $\mathbf{G}_3$.

The game maintains the following activation invariant. If $\mathsf{Init}_{\mathcal{S}}[\pi] = \perp$, then no environment-facing Mask call for π is accepted and no initialization leakage for π has been emitted. The stronger condition $\mathsf{Act}_{\mathcal{S}}[\pi] \neq \perp$ only means that the simulator has internally sampled the ideal coordinates of π and received the ideal leakage transcript under section 4.1. This internal activation is not itself an environment event.

If a revealed-seed public oracle query names a profile that has not yet been externally initialized, the simulator may therefore run $\mathsf{Ensure}(\pi)$ internally. Any $\mathsf{CorMask}$ leakage emitted by that ideal initialization is used only to cache coordinates for parties already corrupted in the same session. The requested distinguished entry is then packed and returned from the conditional law given those cached coordinates. This public random-oracle answer is the only visible effect of the hidden activation, and by the posterior invariant of $\mathbf{G}_3$ it has the same law as the corresponding real lazy-table answer. When the environment later initializes π, the simulator reuses the stored return value, sets $\mathsf{Init}_{\mathcal{S}}[\pi]$, and emits only the real-format initialization leakage. Thus hidden activation is not visible as an initialization event, all previously returned values are preserved, and the environment sees the same real-format leakage messages as in $\mathcal{R}_{\Delta}$. Hence

$$\mathsf{pr}_4 = \mathsf{pr}_3.$$

Game $\mathbf{G}_5$: Ideal execution with the ROM simulator. This game is the ideal execution with $\mathcal{F}_{\Delta}$ and the simulator $\mathcal{S}_{\mathrm{ROM}}$ defined at the start of the proof. Malformed, non-profile, and wrong-seed public random-oracle queries are answered by an ordinary lazy table. Distinguished entries are materialized only through the packed conditional law above, and no visible random-oracle answer is ever changed. Corruption openings for session sid return concrete seed tuples $\mathsf{s}_{i,\mathsf{sid}}$ externally and keep the proof-only ideal opening state synchronized with $\mathcal{F}_{\Delta}$. Hence

$$\mathsf{pr}_5 = \Pr\left[\mathcal{E}^{\mathsf{ideal}}_{\mathcal{F}_{\Delta},\mathcal{S}_{\mathrm{ROM}},\mathcal{Z}}\left(1^{\lambda},\mathsf{N},\mathsf{t}\right)=1\right] = \mathsf{pr}_4.$$

Combining the hybrid bounds gives

$$\begin{aligned}&\Big|\Pr\left[\mathcal{E}^{\mathsf{real},\mathsf{H}_{\Delta}}_{\mathcal{R}_{\Delta},\mathcal{Z}}\left(1^{\lambda},\mathsf{N},\mathsf{t}\right)=1\right]\\&-\Pr\left[\mathcal{E}^{\mathsf{ideal}}_{\mathcal{F}_{\Delta},\mathcal{S}_{\mathrm{ROM}},\mathcal{Z}}\left(1^{\lambda},\mathsf{N},\mathsf{t}\right)=1\right]\Big| = |\mathsf{pr}_0-\mathsf{pr}_5| \le \frac{\mathsf{Q}_{\mathsf{M}}}{2^{\lambda}}.\end{aligned}$$

□

B Completeness of Pilaf

Proof. We prove theorem 2. Fix an honest execution on identifier $\mathsf{id}=(\mathsf{sid},\mathsf{rid})$, message m, and signer set S with $\#(\mathsf{S})=\mathsf{t}$. Let $\left(\mathsf{vk}_{\mathbf{\Sigma}},\{\mathsf{sk}_i\}_{i\in[\mathsf{N}]}\right)$ be the key-generation output and write $\mathsf{sk}_i=(x_i,\mathsf{s}_i)$. Key generation samples x with $\mathsf{vk}_{\mathbf{\Sigma}}=\mathsf{F}(x)$ and Shamir shares $(x_1,\ldots,x_{\mathsf{N}})$, so

$$\sum_{i\in\mathsf{S}}\lambda_{\mathsf{S},i}x_i = x.$$

For each signer $i\in\mathsf{S}$ and branch $b\in\{0,1\}$, let

$$a_{i,b},z_{i,b},X_{i,b},R_{i,b},\omega_{i,b}=\left(\omega^{X}_{i,b},\omega^{R}_{i,b}\right),\mathsf{c}_{i,b}=\left(\mathsf{x}_{i,b},\mathsf{r}_{i,b}\right),\widehat{\mathsf{c}}_{i,b}$$

be the first-round values produced by Sig_0, where

$$\left(\mathsf{ck}_0^X, \mathsf{ck}_1^X, \mathsf{ck}_0^R, \mathsf{ck}_1^R\right) = \mathsf{H}\left(\mathsf{vk}_{\mathbf{\Sigma}}, \mathsf{id}\right),$$

and

$$\mathsf{x}_{i,b} = \mathsf{Com}\left(\mathsf{ck}_b^X, X_{i,b}; \omega_{i,b}^X\right), \qquad \mathsf{r}_{i,b} = \mathsf{Com}\left(\mathsf{ck}_b^R, R_{i,b}; \omega_{i,b}^R\right).$$

The state returned by Sig_0 also carries s_i, so the same mask seed tuple is used for the first-round commitment mask and for the second-round masks computed by Sig_1.

Since every honest signer receives the same first-round transcript $\{\mathsf{m}_i\}_{i\in\mathsf{S}}$, every honest second-round signer uses the same signing context

$$\mathsf{sinf} := \left(\mathsf{vk}_{\mathbf{\Sigma}}, \{\mathsf{m}_i\}_{i\in\mathsf{S}}, \mathsf{id}, \mathsf{S}, \mathsf{m}\right).$$

We use two invariants instantiated by the honest execution. First, for every legal masking profile $\pi = (\mathsf{S}, \mathsf{type}, \mathsf{ctx})$ used below, zero-sum correctness of the masking scheme gives

$$\sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} \Delta_i^{\mathsf{type}}\left(\mathsf{ctx}\right) = 0.$$

This applies to the first-round commitment-mask profiles with context $(\mathsf{id}, \mathsf{S}, b)$ and to the second-round profiles with context (sinf, b). Second, the honest first-round algorithm samples an additive split of each Shamir share and maps both branch preimages through F. Thus the local branch preimages and images satisfy

$$a_{i,0} + a_{i,1} = x_i \qquad \text{and} \qquad X_{i,b} = \mathsf{F}\left(a_{i,b}\right)$$

for every $i \in \mathsf{S}$ and $b \in \{0,1\}$.

The ideal masking invariant for the first-round commitment-mask profile gives, for each branch b,

$$\mathsf{c}_b := \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} \widehat{\mathsf{c}}_{i,b} = \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{c}_{i,b}.$$

Hence all honest signers compute the same aggregate commitments. Write $\mathsf{c}_b = (\mathsf{x}_b, \mathsf{r}_b)$. They compute the same challenge

$$\mathsf{h} := \mathsf{H}_{\mathbf{H}}\left(\mathsf{vk}_{\mathbf{\Sigma}}, \mathsf{x}_0, \mathsf{r}_0, \mathsf{x}_1, \mathsf{r}_1, \mathsf{id}, \mathsf{S}, \mathsf{m}\right).$$

The commitment-consistency check in SAgg therefore holds.

For each branch $b \in \{0,1\}$, define

$$a_b := \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} a_{i,b}, \qquad z_b := \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} z_{i,b}, \qquad \bar{\omega}_b := \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i} \omega_{i,b}.$$

Write $\bar{\omega}_b = \left(\bar{\omega}_b^X, \bar{\omega}_b^R\right)$. Applying the same masking invariant to the second-round profiles gives the values output by SAgg:

$$\begin{aligned} X_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \widehat{X}_{i,b} = \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} X_{i,b} = \mathsf{F}\left(a_b\right), \\ R_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \widehat{R}_{i,b} = \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} R_{i,b} = \mathsf{F}\left(z_b\right), \\ \rho_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \widehat{\rho}_{i,b} = z_b - a_b \mathsf{h}, \\ \omega_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \widehat{\omega}_{i,b} = \bar{\omega}_b. \end{aligned}$$

Here the equalities involving F use the linearity of the linear function family.

The aggregate-key check follows from the local split relation and Shamir reconstruction:

$$\begin{aligned} X_0 + X_1 &= \mathsf{F}\left(a_0\right) + \mathsf{F}\left(a_1\right) \\ &= \mathsf{F}\left(\sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i}\left(a_{i,0} + a_{i,1}\right)\right) = \mathsf{F}\left(x\right) = \mathsf{vk}_{\mathbf{\Sigma}}. \end{aligned}$$

For each branch b, the response equation also holds:

$$\begin{aligned} \mathsf{F}\left(\rho_b\right) + X_b \mathsf{h} &= \mathsf{F}\left(z_b - a_b \mathsf{h}\right) + \mathsf{F}\left(a_b\right) \mathsf{h} \\ &= \mathsf{F}\left(z_b\right) = R_b. \end{aligned}$$

It remains to check the concrete openings behind the ideal commitment handles. Since for each branch b all honest signers use the same branch key ck_b^X for the X commitments and the same key ck_b^R for the branch-b R commitments, the homomorphism of Com from definition 10 gives

$$\begin{aligned} \mathsf{x}_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{x}_{i,b} = \mathsf{Com}\left(\mathsf{ck}_b^X, \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} X_{i,b}; \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \omega_{i,b}^X\right) \\ &= \mathsf{Com}\left(\mathsf{ck}_b^X, X_b; \bar{\omega}_b^X\right), \\ \mathsf{r}_b &= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \mathsf{r}_{i,b} = \mathsf{Com}\left(\mathsf{ck}_b^R, \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} R_{i,b}; \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \omega_{i,b}^R\right) \\ &= \mathsf{Com}\left(\mathsf{ck}_b^R, R_b; \bar{\omega}_b^R\right). \end{aligned}$$

Since $\omega_b = \left(\bar{\omega}_b^X, \bar{\omega}_b^R\right)$, the verifier recomputes the same commitments. All checks in Ver accept, so every honestly generated aggregate signature is accepted with probability one. □

C Hybrid Games for Pilaf

The following figures give game-annotated pseudocode for the hybrid sequence used in the proof. The line-end annotation marks the active range, such as $\mathbf{G}_i$-$\mathbf{G}_j$. An annotation on a branch or loop also applies to its unmarked indented body, and otherwise unmarked lines are common to all displayed games. Every displayed procedure call is one of the protocol algorithms, an LFF algorithm, a commitment algorithm, or an interface from $\mathcal{F}_\Delta$ in fig. 1 or the ROM simulator $\mathcal{S}_{\mathrm{ROM}}$ from theorem 1. Lines that sample from a conditional law specify the posterior distribution directly.

C.1 Completion Records and Posterior Samplers

The hybrids use completion records only as proof state. They are not part of the external syntax of Pilaf. A completion record for an identifier $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ is written $\mathsf{CRec}[\mathsf{id}]$ and contains

$$(\mathsf{sid}, \mathsf{id}, \mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}], \mathsf{vk}_\Sigma[\mathsf{sid}], \mathsf{T}_\mathsf{H}\,[\mathsf{vk}_\Sigma[\mathsf{sid}], \mathsf{id}]\,, \mathsf{RowState}[\mathsf{id}], \mathsf{Loc}[\mathsf{id}, \cdot, \cdot], \mathsf{Agg}[\mathsf{id}, \cdot], \mathsf{Vis}[\mathsf{id}])\,.$$

Here $\mathsf{RowState}[\mathsf{id}]$ stores the hidden branch bit b_{id} and the available simulation trapdoors. It contains both X trapdoors and both R trapdoors in $\mathbf{G}_3$-$\mathbf{G}_{12}$, only the complementary X trapdoor and both R trapdoors in $\mathbf{G}_{13}$, and only the complementary X and R trapdoors in $\mathbf{G}_{14}$. For each signer i and branch b, $\mathsf{Loc}[\mathsf{id}, i, b]$ stores handles for

$$a_{i,b}, z_{i,b}, X_{i,b}, R_{i,b}, \omega^X_{i,b}, \omega^R_{i,b}, \omega^X_{i,b}, \omega^R_{i,b}, \mathsf{x}_{i,b}, \mathsf{r}_{i,b}, \mathsf{c}_{i,b}, \widehat{\mathsf{c}}_{i,b}, \tau^X_{i,b}, \tau^R_{i,b}.$$

Each handle is marked either latent or fixed. A fixed handle is immutable. For an honest-transcript aggregate sinf, $\mathsf{Agg}[\mathsf{id}, \mathsf{sinf}]$ stores the honest-coordinate aggregate values

$$X^{\mathcal{H}}_b, R^{\mathcal{H}}_b, \rho^{\mathcal{H}}_b, \omega^{X,\mathcal{H}}_b, \omega^{R,\mathcal{H}}_b, \omega^{X,\mathcal{H}}_b, \omega^{R,\mathcal{H}}_b, \tau^{X,\mathcal{H}}_{\mathsf{id},b}, \tau^{R,\mathcal{H}}_{\mathsf{id},b} \qquad (b \in \{0,1\}),$$

again as latent or fixed handles. The set $\mathsf{Vis}[\mathsf{id}]$ records every value returned to the adversary and every branch commitment string that has appeared in a $\mathsf{H}_\mathbf{H}$ input.

The following invariant is maintained in $\mathbf{G}_7$-$\mathbf{G}_{14}$.

Immutable visibility. Once a value is placed in $\mathsf{Vis}[\mathsf{id}]$, returned by an oracle, or used in a $\mathsf{H}_\mathbf{H}$ input, later procedures read the same value and never overwrite it.

Visible/decomposition separation. Masked first-round commitments and masked second-round shares are stored as visible equations. Their decompositions into unmasked local values and profile mask coordinates remain latent unless LocalState or LocalStateAfterLastTouch explicitly opens them. A sampler may add constraints from visible equations and may sample latent variables from the corresponding conditional fiber, but it may not overwrite a visible value or treat a masked value as revealing either side of its decomposition.

Local affine fiber. For each fixed id and signer set S, the latent local coordinates are distributed as the honest local variables conditioned on the immutable values in $\mathsf{Vis}[\mathsf{id}]$, on the equations $a_{i,0} + a_{i,1} = x_{i,\mathsf{sid}}$ for opened long-term shares, and on all initialized mask profiles $(\mathsf{S}, \mathsf{c}, (\mathsf{id}, \mathsf{S}, b))$ and $(\mathsf{S}, \mathsf{type}, (\mathsf{sinf}, b))$ for $\mathsf{type} \in \{X, R, \rho, \omega\}$.

Commitment fiber. If a commitment string γ under a binding key ck is immutable and the target value Y is fixed later, the opening handle is sampled from

$$\mathsf{OpenFib}_{\mathsf{ck},\gamma}(Y) := \{\omega \in \mathcal{O}_{\mathsf{LFF}} : \mathsf{Com}(\mathsf{ck}, Y; \omega) = \gamma\}.$$

The hidden branch is maintained in source mode. If a complementary-branch key is in simulation mode, the corresponding traced commitment may be materialized lazily by TCom the first time a trace is needed. A later target opening is obtained by TCol on the cached trace and the chosen proof-only representative. This retargets only the opening handle. No public commitment string is ever changed. In the MDDH instantiation these fibers are the linear solution spaces in the exponent variables of fig. 8.

Masking fiber. Mask coordinates are sampled by the profile-wise completion law of definition 4. Thus a profile with fewer than t fixed coordinates is sampled by choosing non-pivot coordinates uniformly and setting the pivot by the Lagrange zero-sum equation, as in lemma 1.

Last-touch residual fiber. If $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$, the second-round answer $\sigma_{\ell,\mathsf{id}}$ was produced by the aggregate residual rule of fig. 18. The record stores the returned masked local coordinates, the aggregate handles fixed by that rule, and the exact transcript $\mathsf{SInf}_{\ell,\mathsf{id}}$. It does not materialize the current signer local state before a later corruption. A later corruption opens that state only through $\mathsf{LocalStateAfterLastTouch}$, which samples the unmasked local coordinates and their mask coordinates jointly from the conditional fiber cut out by the cached first-round message and the cached second-round answer.

For a record $C = \mathsf{CRec}[\mathsf{id}]$, the proof procedures below are the only posterior samplers used by the hybrids.

$\mathsf{InitCRec}(\mathsf{id}, \mathsf{S}, \mathsf{m})$**.** The procedure checks $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$, $\mathsf{vk}_{\Sigma}[\mathsf{sid}] \neq \perp$, and $(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) = (\mathsf{S}, \mathsf{m})$. It reads the registered row $\mathsf{T}_{\mathsf{H}}[\mathsf{vk}_{\Sigma}[\mathsf{sid}], \mathsf{id}] = \left(\mathsf{ck}^{X}_{\mathsf{id},0}, \mathsf{ck}^{X}_{\mathsf{id},1}, \mathsf{ck}^{R}_{\mathsf{id},0}, \mathsf{ck}^{R}_{\mathsf{id},1}\right)$ and $\mathsf{RowState}[\mathsf{id}]$, creates empty local and aggregate tables, and sets $\mathsf{Vis}[\mathsf{id}] := \emptyset$. It fails if the registered row is absent.

$\mathsf{FirstRoundVisible}(C, \ell)$**.** If the first-round pair for (id, ℓ) is already cached, this procedure returns the cached pair. Otherwise it samples only the visible masked branch-commitment pair $(\widehat{\mathsf{c}}_{\ell,0}, \widehat{\mathsf{c}}_{\ell,1})$ from the exact visible marginal of the completion record's joint posterior. For each $b \in \{0, 1\}$, it records the visible equation

$$\widehat{\mathsf{c}}_{\ell,b} = \mathsf{c}_{\ell,b} + \Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b), \qquad \mathsf{c}_{\ell,b} := (\mathsf{x}_{\ell,b}, \mathsf{r}_{\ell,b}).$$

The right-hand side is not decomposed by this procedure. It does not read or set $\mathsf{ShareTab}[\mathsf{sid}, \ell]$. It does not sample $x_{\ell,\mathsf{sid}}$, $a_{\ell,b}$, $z_{\ell,b}$, $X_{\ell,b}$, $R_{\ell,b}$, commitment openings, proof-only representatives, or TCom traces. It also does not read or open the profile coordinate $\Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b)$. That coordinate remains latent unless a later corruption or local-state opening, or a public distinguished mask-oracle query in the ROM simulator, forces it to be materialized. The procedure appends $\widehat{\mathsf{c}}_{\ell,0}, \widehat{\mathsf{c}}_{\ell,1}$ to $\mathsf{Vis}[\mathsf{id}]$ and fails only if the visible-marginal posterior fiber is empty. Under the preceding invariants this failure event is impossible. It returns $(\widehat{\mathsf{c}}_{\ell,0}, \widehat{\mathsf{c}}_{\ell,1})$.

$\mathsf{LocalState}(C, \ell, \mathsf{sinf})$**.** Except for rows marked $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$, this is the first ordinary procedure that materializes the local decomposition. It first ensures that the cached first-round masked values $(\widehat{\mathsf{c}}_{\ell,0}, \widehat{\mathsf{c}}_{\ell,1})$ are present and conditions on their visible equations. When $\mathsf{sinf} \neq \bot$, it also conditions on the cached second-round masked equations for the exact transcript sinf. If the local share handle is needed and $\mathsf{ShareTab}[\mathsf{sid}, \ell] = \bot$, then in $\mathbf{G}_{10}$-$\mathbf{G}_{14}$ the procedure samples $x_{\ell,\mathsf{sid}}$ from the long-term share posterior of lemma 16, stores it in $\mathsf{ShareTab}[\mathsf{sid}, \ell]$, and reuses the same handle for all later local openings and corruptions in this session. It never requires a global witness x satisfying $\mathsf{F}(x) = \mathsf{vk}_{\Sigma}[\mathsf{sid}]$ after the uniform public-key hop. The procedure then samples $x_{\ell,\mathsf{sid}}$, the branch preimages, the X and R values, commitment openings, proof-only representatives when needed, the first-round c-mask coordinates, and the second-round mask coordinates jointly from the conditional fiber. It reads the fixed seed tuple $\mathsf{s}_{\ell,\mathsf{sid}}$ and returns

$$\left(\mathsf{s}_{\ell,\mathsf{sid}}, \{a_{\ell,b}, z_{\ell,b}, X_{\ell,b}, R_{\ell,b}, \omega_{\ell,b}\}_{b\in\{0,1\}}\right).$$

If a second-round answer on sinf is already immutable, the returned local state is sampled from the current affine and masking fiber conditioned on reproducing that answer. The procedure fixes only handles that are latent, and it is not called for rows with $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$.

$\mathsf{LocalStateAfterLastTouch}(C, \ell, \mathsf{sinf}, \sigma_{\ell,\mathsf{id}})$**.** This procedure is defined only when $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$. It parses the cached first-round message as $\mathsf{m}_{\ell,\mathsf{id}}[b] = \widehat{\mathsf{c}}_{\ell,b}$ and the cached second-round answer as

$$\sigma_{\ell,\mathsf{id}} = \left\{\widehat{X}_{\ell,b}, \widehat{R}_{\ell,b}, \widehat{\rho}_{\ell,b}, \widehat{\omega}_{\ell,b}, \mathsf{c}_b\right\}_{b\in\{0,1\}}.$$

Thus it conditions simultaneously on the cached first-round masked commitment equations and on the cached residual second-round masked equations. The residual share does not determine the unmasked local state by itself. It samples

$$\left(a_{\ell,b}, z_{\ell,b}, X_{\ell,b}, R_{\ell,b}, \omega^{X}_{\ell,b}, \omega^{R}_{\ell,b}, \omega^{X}_{\ell,b}, \omega^{R}_{\ell,b}, \Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b), \left\{\Delta^{\mathsf{type}}_{\ell}(\mathsf{sinf}, b)\right\}_{\mathsf{type}\in\{X,R,\rho,\omega\}}\right)_{b\in\{0,1\}}$$

from the joint conditional fiber of lemma 10. The fiber includes

$$\begin{aligned}
&a_{\ell,0}+a_{\ell,1}=x_{\ell,\mathsf{sid}}, && X_{\ell,b}=\mathsf{F}\left(a_{\ell,b}\right), && R_{\ell,b}=\mathsf{F}\left(z_{\ell,b}\right),\\
&X_{\ell,b}=\mathsf{Rep}_{\mathsf{LFF}}\left(\omega^{X}_{\ell,b}\right), && R_{\ell,b}=\mathsf{Rep}_{\mathsf{LFF}}\left(\omega^{R}_{\ell,b}\right), && \rho_{\ell,b}=z_{\ell,b}-a_{\ell,b}\mathsf{h},\\
&\mathsf{x}_{\ell,b}=\mathsf{Com}\left(\mathsf{ck}^{X}_{\mathsf{id},b},X_{\ell,b};\omega^{X}_{\ell,b}\right), && \mathsf{r}_{\ell,b}=\mathsf{Com}\left(\mathsf{ck}^{R}_{\mathsf{id},b},R_{\ell,b};\omega^{R}_{\ell,b}\right), && \widehat{\mathsf{c}}_{\ell,b}=\mathsf{c}_{\ell,b}+\Delta^{\mathsf{c}}_{\ell}(\mathsf{id},\mathsf{S}\\
&\widehat{Y}_{\ell,b}=Y_{\ell,b}+\Delta^{Y}_{\ell}\left(\mathsf{sinf},b\right)\quad(Y\in\{X,R,\rho\}), && \widehat{\omega}_{\ell,b}=\omega_{\ell,b}+\Delta^{\omega}_{\ell}\left(\mathsf{sinf},b\right), && \omega_{\ell,b}:=\left(\omega^{X}_{\ell,b},\omega^{R}_{\ell,b}\right).
\end{aligned}$$

The same sample also fixes the corresponding mask coordinates in the ROM simulator's corruption-opening records. The procedure returns

$$\left(\mathsf{s}_{\ell,\mathsf{sid}},\{a_{\ell,b},z_{\ell,b},X_{\ell,b},R_{\ell,b},\omega_{\ell,b}\}_{b\in\{0,1\}}\right).$$

$\mathsf{OpenXAgg}(C,\mathsf{vk}^{\mathcal{H}}_{\Sigma},\{\mathsf{x}^{\mathcal{H}}_{b}\}_{b\in\{0,1\}},b^{\star})$**.** The procedure first opens the hidden branch $b^{\star}$ in source mode. It samples $\left(a^{\mathcal{H}}_{b^{\star}},X^{\mathcal{H}}_{b^{\star}},\omega^{X,\mathcal{H}}_{b^{\star}},\omega^{X,\mathcal{H}}_{b^{\star}}\right)$ from the current first-round conditional law with

$$\begin{aligned}
X^{\mathcal{H}}_{b^{\star}}=\mathsf{F}\left(a^{\mathcal{H}}_{b^{\star}}\right), \qquad X^{\mathcal{H}}_{b^{\star}}=\mathsf{Rep}_{\mathsf{LFF}}\left(\omega^{X,\mathcal{H}}_{b^{\star}}\right),\\
\mathsf{Com}\left(\mathsf{ck}^{X}_{\mathsf{id},b^{\star}},X^{\mathcal{H}}_{b^{\star}};\omega^{X,\mathcal{H}}_{b^{\star}}\right)=\mathsf{x}^{\mathcal{H}}_{b^{\star}}.
\end{aligned}$$

This source-mode branch does not call TCol and does not require $\mathsf{tk}^{X}_{\mathsf{id},b^{\star}}$. The complementary branch $\bar{b}^{\star}:=1-b^{\star}$ absorbs the public-key residual. The procedure sets

$$X^{\mathcal{H}}_{\bar{b}^{\star}}:=\mathsf{vk}^{\mathcal{H}}_{\Sigma}-X^{\mathcal{H}}_{b^{\star}},$$

and samples a proof-only representative $\omega^{X,\mathcal{H}}_{\bar{b}^{\star}}$ with $X^{\mathcal{H}}_{\bar{b}^{\star}}=\mathsf{Rep}_{\mathsf{LFF}}\left(\omega^{X,\mathcal{H}}_{\bar{b}^{\star}}\right)$. If the simulation trace for $\mathsf{x}^{\mathcal{H}}_{\bar{b}^{\star}}$ is latent, the procedure materializes a source tuple $\left(\widehat{X}^{\mathcal{H}}_{\bar{b}^{\star}},\widehat{\omega}^{X,\mathcal{H}}_{\bar{b}^{\star}},\widehat{\omega}^{X,\mathcal{H}}_{\bar{b}^{\star}},\widehat{\tau}^{X,\mathcal{H}}_{\mathsf{id},\bar{b}^{\star}}\right)$ from the current first-round conditional law under $\mathsf{ck}^{X}_{\mathsf{id},\bar{b}^{\star}}$. It then calls TCol under $\mathsf{tk}^{X}_{\mathsf{id},\bar{b}^{\star}}$ with $\rho=0\in\mathsf{D}_{\mathsf{LFF}}$, $c=1\in\mathsf{S}_{\mathsf{LFF}}$, and target representative $\omega^{X,\mathcal{H}}_{\bar{b}^{\star}}$. The resulting opening $\omega^{X,\mathcal{H}}_{\bar{b}^{\star}}$ opens the same public string to the residual target. The procedure stores the sampled aggregate handles and the complementary X trace, and fails exactly when the represented residual fiber or a required source fiber is empty.

$\mathsf{Resp}(C,b,\mathsf{sinf})$**.** The procedure samples $\rho^{\mathcal{H}}_{b}$ from the affine fiber of honest aggregate response preimages conditioned on immutable second-round answers on the exact transcript sinf and on the already fixed mask profile $(\mathsf{S},\rho,(\mathsf{sinf},b))$. If no immutable response constrains this branch, the fiber is the full space $\mathsf{D}_{\mathsf{LFF}}$. The sampled value is stored in $\mathsf{Agg}[\mathsf{id},\mathsf{sinf}]$.

$\mathsf{OpenRAgg}(C,\mathsf{r}^{\mathcal{H}}_{b},\eta,b)$**.** If $\eta=\perp$, the procedure samples an ordinary source opening $\left(z^{\mathcal{H}}_{b},R^{\mathcal{H}}_{b},\omega^{R,\mathcal{H}}_{b}\right)$ from the conditional first-round fiber satisfying $R^{\mathcal{H}}_{b}=\mathsf{F}\left(z^{\mathcal{H}}_{b}\right)$ and $\mathsf{Com}\left(\mathsf{ck}^{R}_{\mathsf{id},b},R^{\mathcal{H}}_{b};\omega^{R,\mathcal{H}}_{b}\right)=\mathsf{r}^{\mathcal{H}}_{b}$. This is the only mode used for

the hidden branch b_{id}. If $\eta = R_b^{\mathcal{H}}$ is a fixed target and an opening for $\mathsf{r}_b^{\mathcal{H}}$ to $R_b^{\mathcal{H}}$ is already immutable, the procedure returns it. If the branch has a latent ordinary opening to that target, it samples from $\mathsf{OpenFib}_{\mathsf{ck}_{\mathsf{id},b}^R,\mathsf{r}_b^{\mathcal{H}}}\left(R_b^{\mathcal{H}}\right)$ and stores the opening. If the branch is the complementary branch used by $\mathsf{LastTouchSig}$ and the simulation trace is still latent, it first materializes the traced source commitment with TCom using the cached source representative and then calls TCol with the current challenge and response. It fails exactly when the requested source or target fiber is empty.

Lemma 9 (First-round masked commitment posterior). *Fix a completion record $C = \mathsf{CRec}[\mathsf{id}]$, a signer set S, a signer $\ell \in \mathsf{S}$, and the cached first-round visible pair $(\widehat{\mathsf{c}}_{\ell,0}, \widehat{\mathsf{c}}_{\ell,1})$. Condition on all immutable visible values and on all already materialized profile coordinates. If, for each first-round c-mask profile $(\mathsf{S}, \mathsf{c}, (\mathsf{id}, \mathsf{S}, b))$, fewer than t coordinates have been materialized outside the current opening, then*

$$\left(\mathsf{c}_{\ell,b}, X_{\ell,b}, R_{\ell,b}, a_{\ell,b}, z_{\ell,b}, \omega_{\ell,b}^X, \omega_{\ell,b}^R, \omega_{\ell,b}^X, \omega_{\ell,b}^R, \Delta_\ell^{\mathsf{c}}(\mathsf{id}, \mathsf{S}, b)\right)_{b\in\{0,1\}}$$

is distributed as the marginal of the uniform conditional fiber cut out by the cached visible values and, for $b \in \{0,1\}$, the equations

$$\begin{aligned}
&\widehat{\mathsf{c}}_{\ell,b} = \mathsf{c}_{\ell,b} + \Delta_\ell^{\mathsf{c}}(\mathsf{id}, \mathsf{S}, b), && \mathsf{c}_{\ell,b} := (\mathsf{x}_{\ell,b}, \mathsf{r}_{\ell,b}),\\
&\mathsf{x}_{\ell,b} = \mathsf{Com}\left(\mathsf{ck}_{\mathsf{id},b}^X, X_{\ell,b}; \omega_{\ell,b}^X\right), && \mathsf{r}_{\ell,b} = \mathsf{Com}\left(\mathsf{ck}_{\mathsf{id},b}^R, R_{\ell,b}; \omega_{\ell,b}^R\right),\\
&X_{\ell,b} = \mathsf{F}(a_{\ell,b}), && R_{\ell,b} = \mathsf{F}(z_{\ell,b}),\\
&X_{\ell,b} = \mathsf{Rep}_{\mathsf{LFF}}\left(\omega_{\ell,b}^X\right), && R_{\ell,b} = \mathsf{Rep}_{\mathsf{LFF}}\left(\omega_{\ell,b}^R\right).
\end{aligned}$$

If $x_{\ell,\mathsf{sid}}$ has already been opened, the fiber also includes $a_{\ell,0} + a_{\ell,1} = x_{\ell,\mathsf{sid}}$. If it is still latent, that equation is imposed only when the share handle is opened.

Proof. Before conditioning, the local branch preimages, commitment representatives, commitment openings, and first-round c-mask coordinates are sampled subject only to the displayed equations and to the profile-wise zero-sum mask equations. The visible masked commitment is an affine equality linking an unmasked commitment pair and a mask coordinate. It does not reveal either component separately.

For each relevant first-round mask profile, fewer than t coordinates have been materialized outside the current opening. The profile-wise zero-sum masking functionality and lemma 1 give a pivot coordinate and the exact uniform Lagrange completion for the remaining mask values. The commitment equations are handled by the corresponding opening fibers under the registered keys, sampled jointly with the values that they open. Combining these conditional completions gives the stated marginal. The conditional fiber is nonempty under the invariant because the honest execution gives one assignment before the lazy reordering, and the reordering only delays materializing its decomposition. □

Lemma 10 (Joint masked local-state posterior). *Fix a completion record* $C = \mathsf{CRec}[\mathsf{id}]$*, a signer set* S*, a signer* $\ell \in \mathsf{S}$*, and a transcript*

$$\mathsf{sinf} = (\mathsf{vk}_{\Sigma}[\mathsf{sid}], \tau, \mathsf{id}, \mathsf{S}, \mathsf{m})\,.$$

Assume that the current view is legal and that, for every initialized profile containing ℓ*, fewer than* t *coordinates have been fixed outside the value being opened. Conditioned on all values in* $\mathsf{Vis}[\mathsf{id}]$*, on the cached first-round masked commitment equations from lemma 9, and, if present, on the cached second-round answer* $\sigma_{\ell,\mathsf{id}}$ *on* sinf*, the joint variables*

$$\begin{pmatrix} a_{\ell,b}, z_{\ell,b}, X_{\ell,b}, R_{\ell,b}, \omega^X_{\ell,b}, \omega^R_{\ell,b}, \\ \omega^X_{\ell,b}, \omega^R_{\ell,b}, \Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b)\,, \\ \left\{\Delta^{\mathsf{type}}_{\ell}(\mathsf{sinf}, b)\right\}_{\mathsf{type}\in\{X,R,\rho,\omega\}} \end{pmatrix}_{b\in\{0,1\}}$$

are distributed as the marginal of the uniform distribution on the fiber $\mathsf{JFib}_{\ell,\mathsf{id},\mathsf{sinf}}$ *consisting of all assignments satisfying the cached immutable values and the equations*

$$\begin{aligned}
a_{\ell,0} + a_{\ell,1} &= x_{\ell,\mathsf{sid}}, \\
X_{\ell,b} &= \mathsf{F}(a_{\ell,b})\,, \\
R_{\ell,b} &= \mathsf{F}(z_{\ell,b})\,, \\
X_{\ell,b} &= \mathsf{Rep}_{\mathsf{LFF}}\left(\omega^X_{\ell,b}\right), \\
R_{\ell,b} &= \mathsf{Rep}_{\mathsf{LFF}}\left(\omega^R_{\ell,b}\right), \\
\rho_{\ell,b} &= z_{\ell,b} - a_{\ell,b}\mathsf{h}, \\
\mathsf{x}_{\ell,b} &= \mathsf{Com}\left(\mathsf{ck}^X_{\mathsf{id},b}, X_{\ell,b}; \omega^X_{\ell,b}\right), \\
\mathsf{r}_{\ell,b} &= \mathsf{Com}\left(\mathsf{ck}^R_{\mathsf{id},b}, R_{\ell,b}; \omega^R_{\ell,b}\right), \\
\widehat{\mathsf{c}}_{\ell,b} &= \mathsf{c}_{\ell,b} + \Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b)\,, \\
\widehat{Y}_{\ell,b} &= Y_{\ell,b} + \Delta^{Y}_{\ell}(\mathsf{sinf}, b) \quad (Y \in \{X, R, \rho\})\,, \\
\widehat{\omega}_{\ell,b} &= \omega_{\ell,b} + \Delta^{\omega}_{\ell}(\mathsf{sinf}, b)\,, \\
\omega_{\ell,b} &:= \left(\omega^X_{\ell,b}, \omega^R_{\ell,b}\right), \\
\sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i}\Delta^{\mathsf{type}}_{i}(\mathsf{ctx}) &= 0 \quad \textit{for each initialized profile } (\mathsf{S}, \mathsf{type}, \mathsf{ctx})\,.
\end{aligned}$$

Equations involving $\widehat{X}_{\ell,b}$*,* $\widehat{R}_{\ell,b}$*,* $\widehat{\rho}_{\ell,b}$*, or* $\widehat{\omega}_{\ell,b}$ *are imposed exactly when the corresponding second-round value is already visible. The lemma applies equally when* $\mathsf{Touch}[\mathsf{sinf}]$ *reaches* t *through a corruption. The set* $\mathsf{Touch}[\mathsf{sinf}]$ *counts visibility, whereas the joint fiber is cut out only by materialized decompositions and cached masked equations. The corruption opening samples the newly materialized decomposition while leaving a still-latent pivot coordinate whenever a zero-sum equation must be completed.*

Proof. Before conditioning, the local preimages, commitment representatives, commitment openings, and profile mask coordinates used by an honest signer are sampled independently subject only to the displayed Shamir split equation, the LFF image equations, the commitment equations, and the profile-wise zero-sum equations. Revealing a first-round masked commitment or a second-round masked coordinate adds an affine equality linking an unmasked coordinate and a mask coordinate. It does not determine either coordinate by itself.

For every initialized mask profile, the legality condition leaves fewer than t coordinates fixed before the current opening. The profile-wise masking functionality of definition 4 and the cancellation law of lemma 2 give the conditional distribution of the remaining mask coordinates as the uniform Lagrange zero-sum completion. The ROM realization of theorem 1 exposes the same conditional law when the corruption later opens the corresponding concrete seeds.

The commitment part is sampled jointly with the value it opens. For an immutable commitment string and a newly fixed target value, the opening is sampled from $\mathsf{OpenFib}$ under the same registered key. When a simulation trace is needed, the trace is materialized from that sampled source value and opening and is then cached. Thus the value, proof-only representative, opening, commitment string, and any trace belong to one conditional assignment in $\mathsf{JFib}_{\ell,\mathsf{id},\mathsf{sinf}}$.

Combining these independent conditional completions gives the uniform distribution over $\mathsf{JFib}_{\ell,\mathsf{id},\mathsf{sinf}}$ and hence the stated marginal for the current signer's local state and mask coordinates. In a last-touch row, the cached signature contributes only the displayed masked equations. The sampler therefore opens a compatible local state jointly with the mask coordinates, instead of reconstructing an unmasked state from the signature alone. □

Corollary 1 (Simulatable aggregate X openings with branch-specific keys). *Whenever a legal last-touch query reaches* $\mathsf{OpenXAgg}\left(C, \mathsf{vk}_{\Sigma}^{\mathcal{H}}, \{\mathsf{x}_b^{\mathcal{H}}\}_{b\in\{0,1\}}, b_{\mathsf{id}}\right)$ *in* $\boldsymbol{G}_9$-$\boldsymbol{G}_{14}$*, the hidden X branch b_{id} is opened in source mode to an image value, and the complementary X branch $1-b_{\mathsf{id}}$ is target-opened to the residual value. Therefore the aggregate X relation is simulatable without requiring* $\mathsf{vk}_{\Sigma}^{\mathcal{H}} \in \mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)$ *and without using the hidden X-branch trapdoor.*

Proof. After the revised $\mathbf{G}_3$ hop, each registered row has branch-specific X-commitment keys. The aggregate commitment strings $\mathsf{x}_b^{\mathcal{H}}$ are immutable public values. The procedure $\mathsf{OpenXAgg}$ first opens the hidden branch b_{id} as an ordinary source commitment to $X_{b_{\mathsf{id}}}^{\mathcal{H}} = \mathsf{F}\left(a_{b_{\mathsf{id}}}^{\mathcal{H}}\right)$. This keeps the hidden branch compatible with the later binding switch.

It then sets the complementary target to $X_{1-b_{\mathsf{id}}}^{\mathcal{H}} = \mathsf{vk}_{\Sigma}^{\mathcal{H}} - X_{b_{\mathsf{id}}}^{\mathcal{H}}$ and samples a proof-only representative for that represented target. The complementary branch key remains simulation-openable in $\mathbf{G}_{13}$ and $\mathbf{G}_{14}$, so $\mathsf{OpenXAgg}$ calls TCol only under $\mathsf{tk}_{\mathsf{id},1-b_{\mathsf{id}}}^{X}$ with $\rho = 0$ and $c = 1$. By the targetable-opening property of definition 10, this opens the same complementary commitment string to the residual target, except for the statistical error charged in lemma 15. Thus $X_0^{\mathcal{H}} + X_1^{\mathcal{H}} = \mathsf{vk}_{\Sigma}^{\mathcal{H}}$ is enforced by the complementary branch rather than by decoding the public key as an image. □

Lemma 11 (Random-tape and state leakage equivalence). *For the concrete algorithms of fig. 7, there are deterministic encoders* $\mathsf{Enc}_1\left(\mathsf{sk}_{\ell,\mathsf{sid}}, \mathsf{st}_{\ell,\mathsf{id}}\right)$ *and* $\mathsf{Enc}_2\left(\mathsf{st}_{\ell,\mathsf{id}}, \mathsf{sinf}, \sigma_{\ell,\mathsf{id}}\right)$*, and matching decoders, between the random tapes used by fig. 6 and the proof-state leakage used below. The round-one tape consists of the coins that generate* $\left(a_{\ell,b}, z_{\ell,b}, \omega^X_{\ell,b}, \omega^R_{\ell,b}\right)_{b\in\{0,1\}}$ *and the mask-seed tuple* $\mathsf{s}_{\ell,\mathsf{sid}}$ *from the secret key. Given* $\mathsf{sk}_{\ell,\mathsf{sid}}$ *and this tape,* Sig_0 *deterministically reconstructs* $\mathsf{m}_{\ell,\mathsf{id}}$ *and* $\mathsf{st}_{\ell,\mathsf{id}}$*. Conversely,* $\mathsf{st}_{\ell,\mathsf{id}}$ *together with* $\mathsf{sk}_{\ell,\mathsf{sid}}$ *determines the canonical round-one tape. The round-two tape is the deterministic record of the mask profile reads* $\Delta^{\mathsf{type}}_{\ell}(\mathsf{sinf}, b)$ *for* $\mathsf{type} \in \{X, R, \rho, \omega\}$ *and* $b \in \{0, 1\}$*. Given* $\mathsf{st}_{\ell,\mathsf{id}}$*, the supplied transcript* sinf*, and this tape,* Sig_1 *deterministically reconstructs* $\sigma_{\ell,\mathsf{id}}$*. Therefore returning the random tapes in the security definition is distributionally equivalent to returning the corresponding state objects and already returned signature share used as proof notation in fig. 19. If the returned share was produced by* LastTouchSig*, the state object encoded by* Enc_1 *is the output of* LocalStateAfterLastTouch*. The joint posterior lemma fixes the local state and the mask coordinates together, so* Enc_2 *still encodes the same cached share and transcript.*

Alg $\mathsf{Exp}^{\mathsf{hyb}}_{\mathcal{A},\mathsf{Pilaf}}\left(1^\lambda\right)$:

for sid:
  $\mathsf{vk}_\Sigma[\mathsf{sid}] := \perp$; $\mathcal{C}[\mathsf{sid}] := \emptyset$
for id:
  $\mathcal{Q}[\mathsf{id}] := \emptyset$; $\mathsf{S}[\mathsf{id}] := \perp$; $\mathsf{m}[\mathsf{id}] := \perp$
for ℓ, id:
  $\mathsf{Round}_{\ell,\mathsf{id}} := 0$; $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} := 0$
$\mathsf{pp}_{\mathsf{LFF}} \xleftarrow{\$} \mathsf{Setup}\left(1^\lambda\right)$ // $\mathbf{G}_0$-$\mathbf{G}_{10}$
$(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{td}_{\mathsf{im}}) \xleftarrow{\$} \mathsf{Setup}_{\mathsf{im}}\left(1^\lambda\right)$ // $\mathbf{G}_{11}$-$\mathbf{G}_{14}$
$\mathcal{S}_{\mathrm{ROM}}.\mathsf{Setup}\left(1^\lambda, \mathsf{N}, \mathsf{t}\right)$ // $\mathbf{G}_1$-$\mathbf{G}_{14}$
$\left((\mathsf{id}^\star, \mathsf{S}^\star, \mathsf{m}^\star), \sigma^\star_\Sigma\right) \xleftarrow{\$} \mathcal{A}^{\mathsf{KGen},\mathsf{Sig}_0,\mathsf{Sig}_1,\mathsf{Corrupt}}(\mathsf{pp}_{\mathsf{LFF}})$
parse $\mathsf{id}^\star = (\mathsf{sid}^\star, \mathsf{rid}^\star)$
$\mathsf{Fresh} := \mathsf{vk}_\Sigma[\mathsf{sid}^\star] \neq \perp \wedge \mathsf{S}^\star \subseteq [\mathsf{N}] \wedge \#(\mathsf{S}^\star) = \mathsf{t}$
$\mathsf{Fresh} := \mathsf{Fresh} \wedge ((\mathsf{S}[\mathsf{id}^\star], \mathsf{m}[\mathsf{id}^\star]) = (\perp, \perp) \vee (\mathsf{S}[\mathsf{id}^\star], \mathsf{m}[\mathsf{id}^\star]) = (\mathsf{S}^\star, \mathsf{m}^\star))$
$\mathsf{Fresh} := \mathsf{Fresh} \wedge \#(\mathcal{Q}[\mathsf{id}^\star] \cup \mathcal{C}[\mathsf{sid}^\star]) \leq \mathsf{t} - 1$
$\mathsf{Valid} := \mathsf{Ver}(\mathsf{vk}_\Sigma[\mathsf{sid}^\star], \mathsf{id}^\star, \mathsf{S}^\star, \mathsf{m}^\star, \sigma^\star_\Sigma)$
if $\mathsf{Fresh} \wedge \mathsf{Valid} \wedge \mathsf{InImage}\left(\mathsf{td}_{\mathsf{im}}, X^\star_{b_{\mathsf{id}^\star}}\right) = 1$ **then abort** // $\mathbf{G}_{12}$-$\mathbf{G}_{14}$
return $[\![\mathsf{Fresh} \wedge \mathsf{Valid}]\!]$

Fig. 11. Hybrid experiment wrapper.

For a challenge-oracle input h, the helper $\mathsf{RecordH}(\mathsf{h})$ sets $\mathsf{PreH}[\mathsf{vk}_\Sigma, \mathsf{id}] := 1$ when h parses as $(\mathsf{vk}_\Sigma, \mathsf{x}_0, \mathsf{r}_0, \mathsf{x}_1, \mathsf{r}_1, \mathsf{id}, \mathsf{S}, \mathsf{m})$ with $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$. The helper $\mathsf{MaybeRegHash}(\mathsf{h})$ performs the same parsing and, when additionally $\mathsf{vk}_\Sigma = \mathsf{vk}_\Sigma[\mathsf{sid}]$ and $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}] = \perp$, initializes $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}]$ by calling $\mathsf{H}(\mathsf{vk}_\Sigma, \mathsf{id})$.

Figure 12 uses the same masking-ROM simulator instance after the masking-abstraction hop. For $\mathbf{G}_1$-$\mathbf{G}_{14}$, the variables $\mathsf{s}_{i,\mathsf{sid}}$ used by the signing and cor-

ruption oracles are the concrete seed tuples generated by $\mathsf{EnsureSid}(\mathsf{sid})$ inside $\mathcal{S}_{\mathrm{ROM}}$. No independent call to $\Delta.\mathsf{KGen}$ is made outside $\mathcal{S}_{\mathrm{ROM}}$ after this hop.

Oracle $\mathsf{KGen}(\mathsf{sid})$:

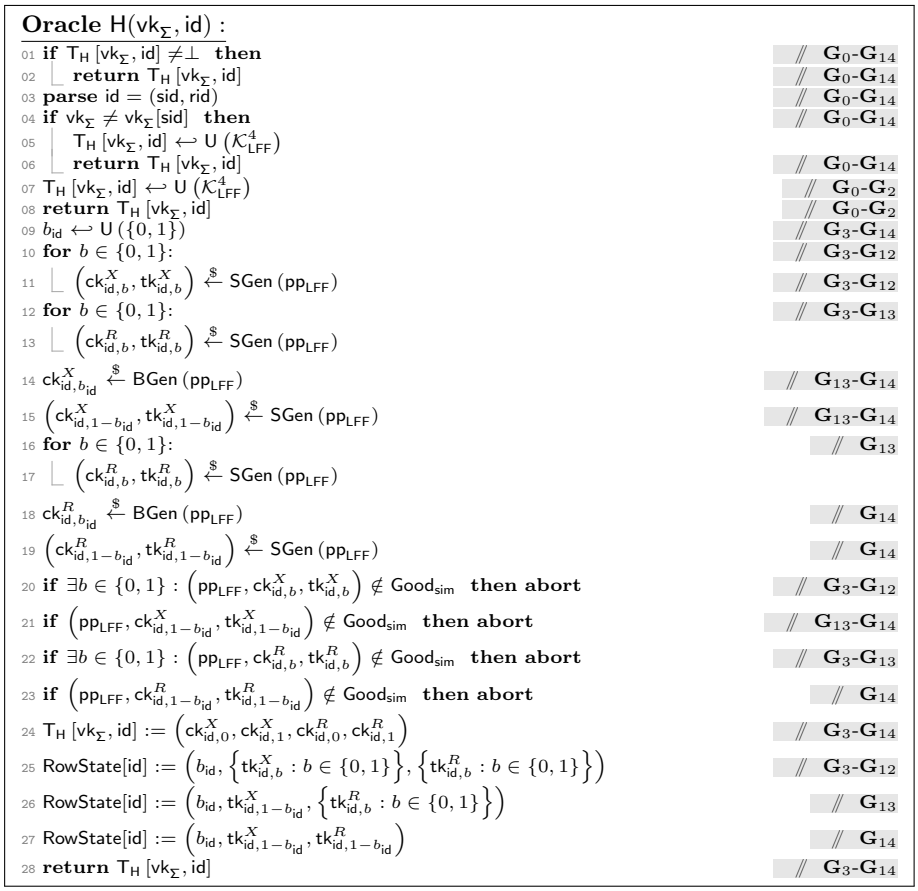
01 **check** $\mathsf{vk}_\Sigma[\mathsf{sid}] = \perp$
02 $x[\mathsf{sid}] \hookleftarrow \mathsf{U}(\mathsf{D}_{\mathsf{LFF}})$ // $\mathbf{G}_0$-$\mathbf{G}_9$
03 $\mathsf{vk}_\Sigma[\mathsf{sid}] := \mathsf{F}(x[\mathsf{sid}])$ // $\mathbf{G}_0$-$\mathbf{G}_9$
04 $(x_{1,\mathsf{sid}}, \ldots, x_{\mathsf{N},\mathsf{sid}}) \xleftarrow{\$} \mathsf{Share}(x[\mathsf{sid}], \mathsf{N}, \mathsf{t})$ // $\mathbf{G}_0$-$\mathbf{G}_9$
05 $\left(\omega_{\mathsf{vk}_\Sigma[\mathsf{sid}]}, \mathsf{vk}_\Sigma[\mathsf{sid}]\right) \xleftarrow{\$} \mathsf{SampRep}_{\mathsf{LFF}}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_{10}$-$\mathbf{G}_{14}$
06 $\{\mathsf{s}_{i,\mathsf{sid}}\}_{i\in[\mathsf{N}]} \xleftarrow{\$} \Delta.\mathsf{KGen}\left(1^\lambda, \mathsf{N}, \mathsf{t}\right)$ // $\mathbf{G}_0$
07 $\mathcal{S}_{\mathrm{ROM}}.\mathsf{EnsureSid}(\mathsf{sid})$ // $\mathbf{G}_1$-$\mathbf{G}_{14}$
08 **for** $i \in [\mathsf{N}]$: // $\mathbf{G}_1$-$\mathbf{G}_{14}$
09 $\quad \mathsf{s}_{i,\mathsf{sid}} := \{(j, \mathsf{s}_{i,j}[\mathsf{sid}], \mathsf{s}_{j,i}[\mathsf{sid}])\}_{j\in[\mathsf{N}]\setminus\{i\}}$ // $\mathbf{G}_1$-$\mathbf{G}_{14}$
10 **for** $i \in [\mathsf{N}]$: // $\mathbf{G}_0$-$\mathbf{G}_9$
11 $\quad \mathsf{sk}_{i,\mathsf{sid}} := (x_{i,\mathsf{sid}}, \mathsf{s}_{i,\mathsf{sid}})$ // $\mathbf{G}_0$-$\mathbf{G}_9$
12 **for** $i \in [\mathsf{N}]$: // $\mathbf{G}_{10}$-$\mathbf{G}_{14}$
13 $\quad \mathsf{ShareTab}[\mathsf{sid}, i] := \perp$ // $\mathbf{G}_{10}$-$\mathbf{G}_{14}$
14 **if** $\exists \mathsf{rid} : \mathsf{T}_{\mathsf{H}}[\mathsf{vk}_\Sigma[\mathsf{sid}], (\mathsf{sid}, \mathsf{rid})] \neq \perp \vee \mathsf{PreH}[\mathsf{vk}_\Sigma[\mathsf{sid}], (\mathsf{sid}, \mathsf{rid})] = 1$ **then abort** // $\mathbf{G}_2$-$\mathbf{G}_{14}$
15 **return** $\mathsf{vk}_\Sigma[\mathsf{sid}]$

Fig. 12. Hybrid key-generation oracle.

After $\mathbf{G}_1$, mask-related public random-oracle queries are answered by the ROM simulator from theorem 1. Malformed, non-profile, and wrong-seed points remain ordinary lazy random-oracle points inside that simulator. The public mask oracle below only forwards $\mathbf{G}_1$-$\mathbf{G}_{14}$ queries to $\mathcal{S}_{\mathrm{ROM}}.\mathsf{H}$, so hidden activation, if any, is internal to that simulator.

In fig. 16, $\mathsf{FirstRoundVisible}$ is visible-only in all games. The $\mathbf{G}_7$ state materialization comes from the following $\mathsf{LocalState}(\mathsf{CRec}[\mathsf{id}], \ell, \perp)$ call, while $\mathbf{G}_8$-$\mathbf{G}_{14}$ omit that call after the first-round answer.

In the corruption oracle below, $\mathsf{View}_{\ell,\mathsf{id}'}$ denotes the already visible transcript constraints for signer ℓ and identifier id'. It contains the first-round message $\mathsf{m}_{\ell,\mathsf{id}'}$ when $\mathsf{Round}_{\ell,\mathsf{id}'} \geq 1$, the exact second-round share $\sigma_{\ell,\mathsf{id}'}$ and its supplied transcript when $\mathsf{Round}_{\ell,\mathsf{id}'} = 2$, and every mask coordinate for ℓ that has already been materialized by an initialized profile. Conditioning on $\mathsf{View}_{\ell,\mathsf{id}'}$ means that the corruption opening extends the existing adversarial view rather than resampling any value already returned. If $\mathsf{LastTouchRow}_{\ell,\mathsf{id}'} = 1$, the view contains the residual second-round answer and the corruption oracle opens the local state with $\mathsf{LocalStateAfterLastTouch}$. In $\mathbf{G}_1$-$\mathbf{G}_{14}$, the corruption oracle invokes $\mathcal{S}_{\mathrm{ROM}}.\mathsf{CorMask}(\mathsf{sid}, \ell)$. This call returns the real-format concrete mask seed tuple and synchronizes the simulator's corrupted-endpoint set and ideal opening records. The hybrid oracle does not call the ideal corruption-opening oracle directly after the masking-abstraction hop, and the global set $\mathcal{C}[\mathsf{sid}]$ remains the threshold-signature bookkeeping set.



Oracle $\mathsf{H}(\mathsf{vk}_\Sigma, \mathsf{id})$:

01 **if** $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}] \neq \perp$ **then** // $\mathbf{G}_0$-$\mathbf{G}_{14}$
02 **return** $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}]$ // $\mathbf{G}_0$-$\mathbf{G}_{14}$
03 **parse** $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$ // $\mathbf{G}_0$-$\mathbf{G}_{14}$
04 **if** $\mathsf{vk}_\Sigma \neq \mathsf{vk}_\Sigma[\mathsf{sid}]$ **then** // $\mathbf{G}_0$-$\mathbf{G}_{14}$
05 $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}] \hookleftarrow \mathsf{U}(\mathcal{K}^4_{\mathsf{LFF}})$
06 **return** $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}]$ // $\mathbf{G}_0$-$\mathbf{G}_{14}$
07 $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}] \hookleftarrow \mathsf{U}(\mathcal{K}^4_{\mathsf{LFF}})$ // $\mathbf{G}_0$-$\mathbf{G}_2$
08 **return** $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}]$ // $\mathbf{G}_0$-$\mathbf{G}_2$
09 $b_{\mathsf{id}} \hookleftarrow \mathsf{U}(\{0,1\})$ // $\mathbf{G}_3$-$\mathbf{G}_{14}$
10 **for** $b \in \{0,1\}$: // $\mathbf{G}_3$-$\mathbf{G}_{12}$
11 $\left(\mathsf{ck}^X_{\mathsf{id},b}, \mathsf{tk}^X_{\mathsf{id},b}\right) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_3$-$\mathbf{G}_{12}$
12 **for** $b \in \{0,1\}$: // $\mathbf{G}_3$-$\mathbf{G}_{13}$
13 $\left(\mathsf{ck}^R_{\mathsf{id},b}, \mathsf{tk}^R_{\mathsf{id},b}\right) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$
14 $\mathsf{ck}^X_{\mathsf{id},b_{\mathsf{id}}} \xleftarrow{\$} \mathsf{BGen}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_{13}$-$\mathbf{G}_{14}$
15 $\left(\mathsf{ck}^X_{\mathsf{id},1-b_{\mathsf{id}}}, \mathsf{tk}^X_{\mathsf{id},1-b_{\mathsf{id}}}\right) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_{13}$-$\mathbf{G}_{14}$
16 **for** $b \in \{0,1\}$: // $\mathbf{G}_{13}$
17 $\left(\mathsf{ck}^R_{\mathsf{id},b}, \mathsf{tk}^R_{\mathsf{id},b}\right) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$
18 $\mathsf{ck}^R_{\mathsf{id},b_{\mathsf{id}}} \xleftarrow{\$} \mathsf{BGen}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_{14}$
19 $\left(\mathsf{ck}^R_{\mathsf{id},1-b_{\mathsf{id}}}, \mathsf{tk}^R_{\mathsf{id},1-b_{\mathsf{id}}}\right) \xleftarrow{\$} \mathsf{SGen}(\mathsf{pp}_{\mathsf{LFF}})$ // $\mathbf{G}_{14}$
20 **if** $\exists b \in \{0,1\} : \left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^X_{\mathsf{id},b}, \mathsf{tk}^X_{\mathsf{id},b}\right) \notin \mathsf{Good}_{\mathsf{sim}}$ **then abort** // $\mathbf{G}_3$-$\mathbf{G}_{12}$
21 **if** $\left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^X_{\mathsf{id},1-b_{\mathsf{id}}}, \mathsf{tk}^X_{\mathsf{id},1-b_{\mathsf{id}}}\right) \notin \mathsf{Good}_{\mathsf{sim}}$ **then abort** // $\mathbf{G}_{13}$-$\mathbf{G}_{14}$
22 **if** $\exists b \in \{0,1\} : \left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^R_{\mathsf{id},b}, \mathsf{tk}^R_{\mathsf{id},b}\right) \notin \mathsf{Good}_{\mathsf{sim}}$ **then abort** // $\mathbf{G}_3$-$\mathbf{G}_{13}$
23 **if** $\left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^R_{\mathsf{id},1-b_{\mathsf{id}}}, \mathsf{tk}^R_{\mathsf{id},1-b_{\mathsf{id}}}\right) \notin \mathsf{Good}_{\mathsf{sim}}$ **then abort** // $\mathbf{G}_{14}$
24 $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}] := \left(\mathsf{ck}^X_{\mathsf{id},0}, \mathsf{ck}^X_{\mathsf{id},1}, \mathsf{ck}^R_{\mathsf{id},0}, \mathsf{ck}^R_{\mathsf{id},1}\right)$ // $\mathbf{G}_3$-$\mathbf{G}_{14}$
25 $\mathsf{RowState}[\mathsf{id}] := \left(b_{\mathsf{id}}, \left\{\mathsf{tk}^X_{\mathsf{id},b} : b \in \{0,1\}\right\}, \left\{\mathsf{tk}^R_{\mathsf{id},b} : b \in \{0,1\}\right\}\right)$ // $\mathbf{G}_3$-$\mathbf{G}_{12}$
26 $\mathsf{RowState}[\mathsf{id}] := \left(b_{\mathsf{id}}, \mathsf{tk}^X_{\mathsf{id},1-b_{\mathsf{id}}}, \left\{\mathsf{tk}^R_{\mathsf{id},b} : b \in \{0,1\}\right\}\right)$ // $\mathbf{G}_{13}$
27 $\mathsf{RowState}[\mathsf{id}] := \left(b_{\mathsf{id}}, \mathsf{tk}^X_{\mathsf{id},1-b_{\mathsf{id}}}, \mathsf{tk}^R_{\mathsf{id},1-b_{\mathsf{id}}}\right)$ // $\mathbf{G}_{14}$
28 **return** $\mathsf{T}_\mathsf{H}[\mathsf{vk}_\Sigma, \mathsf{id}]$ // $\mathbf{G}_3$-$\mathbf{G}_{14}$

Fig. 13. Hybrid commitment-key random oracle.



Oracle $\mathsf{H}_\mathbf{H}(\mathsf{h})$:

01 $\mathsf{RecordH}(\mathsf{h})$ // $\mathbf{G}_2$-$\mathbf{G}_{14}$
02 $\mathsf{MaybeRegHash}(\mathsf{h})$ // $\mathbf{G}_3$-$\mathbf{G}_{14}$
03 **if** $\mathsf{T}_{\mathsf{H}_\mathbf{H}}[\mathsf{h}] = \perp$ **then**
04 $\mathsf{T}_{\mathsf{H}_\mathbf{H}}[\mathsf{h}] \hookleftarrow \mathsf{U}(\mathsf{S}_{\mathsf{LFF}})$
05 **return** $\mathsf{T}_{\mathsf{H}_\mathbf{H}}[\mathsf{h}]$

Fig. 14. Hybrid challenge random oracle.

Lemma 12 (Registered simulation-openable rows). *Let* $Q_{\mathsf{reg}} := \mathsf{Q}_{\mathsf{Sig}_0} + Q_\mathsf{H} + 1$. *The replacement made in* $\boldsymbol{G}_3$ *changes the winning probability by at most*

$$4Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + 4\varepsilon_{\mathsf{key}}(\lambda, Q_{\mathsf{reg}}).$$

Proof. Before the final verification decision, a registered row can be initialized by a legal first-round signing query, by a direct commitment-key query, by the registered-row preinitialization made inside a direct $\mathsf{H}_\mathbf{H}$ query, or by the final

Oracle $\mathsf{H}^{\mathsf{type}}_{\Delta}(\mathsf{ctx}, u, v, s)$:

01 **check** $\mathsf{type} \in \mathcal{T}_{\Delta}$
02 **if** $\mathsf{H}^{\mathsf{type}}_{\Delta}[\mathsf{ctx}, u, v, s] \neq \perp$ **then** // $\mathbf{G}_0$
03 **return** $\mathsf{H}^{\mathsf{type}}_{\Delta}[\mathsf{ctx}, u, v, s]$ // $\mathbf{G}_0$
04 $y \hookleftarrow \mathsf{U}(\mathsf{D}(\mathsf{type}))$ // $\mathbf{G}_0$
05 $\mathsf{H}^{\mathsf{type}}_{\Delta}[\mathsf{ctx}, u, v, s] := y$ // $\mathbf{G}_0$
06 **return** y // $\mathbf{G}_0$
07 **return** $\mathcal{S}_{\mathrm{ROM}}.\mathsf{H}(\mathsf{type}, \mathsf{ctx}, u, v, s)$ // $\mathbf{G}_1$-$\mathbf{G}_{14}$

Fig. 15. Hybrid mask-related random oracle.

Oracle $\mathsf{Sig}_0(\mathsf{vk}'_{\Sigma}, \ell, \mathsf{id}, \mathsf{S}, \mathsf{m})$:

01 **parse** $\mathsf{id} = (\mathsf{sid}, \mathsf{rid})$
02 **check** $\mathsf{vk}_{\Sigma}[\mathsf{sid}] \neq \perp \wedge \mathsf{vk}'_{\Sigma} = \mathsf{vk}_{\Sigma}[\mathsf{sid}] \wedge \ell \notin \mathcal{C}[\mathsf{sid}]$
03 **check** $\ell \in \mathsf{S} \wedge \#(\mathsf{S}) = \mathsf{t} \wedge \mathsf{Round}_{\ell,\mathsf{id}} = 0$
04 **check** $(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) = (\perp, \perp) \vee (\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) = (\mathsf{S}, \mathsf{m})$
05 $(\mathsf{S}[\mathsf{id}], \mathsf{m}[\mathsf{id}]) := (\mathsf{S}, \mathsf{m})$
06 $\left(\mathsf{ck}_0^X, \mathsf{ck}_1^X, \mathsf{ck}_0^R, \mathsf{ck}_1^R\right) := \mathsf{H}(\mathsf{vk}_{\Sigma}[\mathsf{sid}], \mathsf{id})$
07 $(\mathsf{m}_{\ell,\mathsf{id}}, \mathsf{st}_{\ell,\mathsf{id}}) \xleftarrow{\$} \mathsf{Sig}_0(\mathsf{vk}'_{\Sigma}, \ell, \mathsf{id}, \mathsf{sk}_{\ell,\mathsf{sid}}, \mathsf{S}, \mathsf{m})$ // $\mathbf{G}_0$-$\mathbf{G}_6$
08 **if** $\mathsf{CRec}[\mathsf{id}] = \perp$ **then** // $\mathbf{G}_7$-$\mathbf{G}_{14}$
09 $\mathsf{CRec}[\mathsf{id}] := \mathsf{InitCRec}(\mathsf{id}, \mathsf{S}, \mathsf{m})$ // $\mathbf{G}_7$-$\mathbf{G}_{14}$
10 $\mathsf{m}_{\ell,\mathsf{id}} := \mathsf{FirstRoundVisible}(\mathsf{CRec}[\mathsf{id}], \ell)$ // $\mathbf{G}_7$
11 $\mathsf{st}_{\ell,\mathsf{id}} := \mathsf{LocalState}(\mathsf{CRec}[\mathsf{id}], \ell, \perp)$ // $\mathbf{G}_7$
12 $\mathsf{m}_{\ell,\mathsf{id}} := \mathsf{FirstRoundVisible}(\mathsf{CRec}[\mathsf{id}], \ell)$ // $\mathbf{G}_8$-$\mathbf{G}_{14}$
13 $\mathsf{Round}_{\ell,\mathsf{id}} := 1$
14 **return** $\mathsf{m}_{\ell,\mathsf{id}}$

Fig. 16. Hybrid first-round signing oracle.

verification. Since Q_{H} counts direct commitment-key and $\mathsf{H}_{\mathbf{H}}$ queries together, there are at most $\mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1$ such rows. Each $\mathbf{G}_3$ row contains a hidden branch bit b_{id}, two X-commitment SGen outputs, and two R-commitment SGen outputs. The game aborts if any of these four simulation keys is not in $\mathsf{Good}_{\mathsf{sim}}$. This abort is symmetric in the two branch labels and therefore does not leak or condition on b_{id}. A union bound over the four sampled simulation keys in each registered row gives

$$\Pr[\mathsf{Bad}_{\mathsf{sim}}] \leq 4Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}}.$$

Before this abort is tested, the registered row in $\mathbf{G}_2$ contains four uniform keys. By the uniform binding-key property of definition 10, these are identical to four BGen keys. The two X public components and the two R public components are changed from BGen keys to SGen public components by four applications of the multi-key indistinguishability property. The hidden branch bit and trapdoors are proof state. They are not revealed as part of the hash answer. X and R commitments later materialized with TCom have the same visible value as Com by trace correctness. Therefore the visible difference from $\mathbf{G}_2$ is bounded by the displayed key-switching and abort terms. □

Lemma 13 (Touch bookkeeping). *The bookkeeping in $\boldsymbol{G}_4$ is distribution preserving, and the abort events* BadHonTouch *and* BadMalTouch *introduced in $\boldsymbol{G}_5$ and $\boldsymbol{G}_6$ have probability zero on legal second-round queries.*

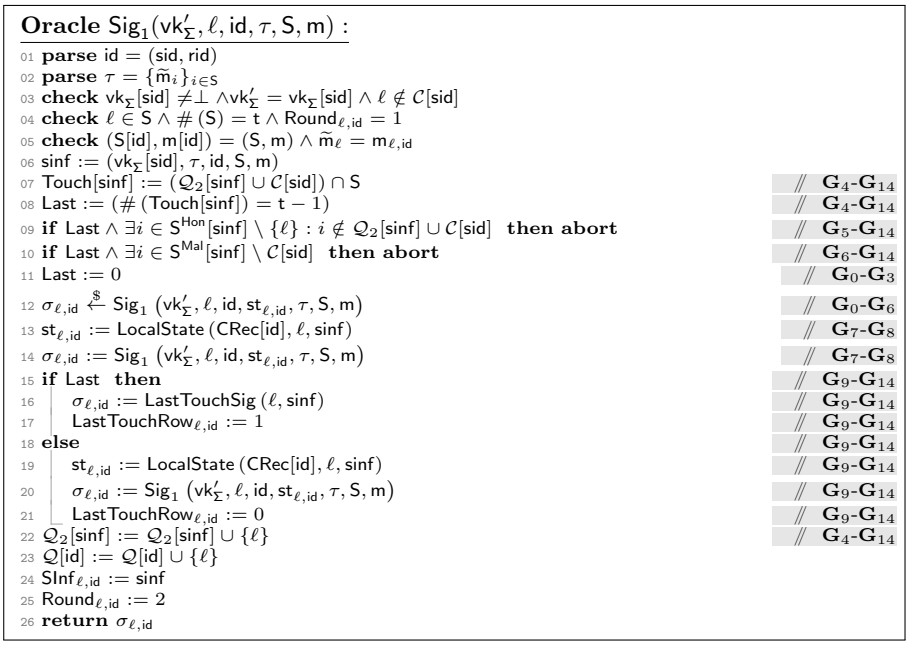


Fig. 17. Hybrid second-round signing oracle.

Proof. $\mathbf{G}_4$ only records $\mathcal{Q}_2[\mathsf{sinf}]$ and $\mathsf{Touch}[\mathsf{sinf}]$ and marks whether the current query is last-touch. It returns exactly the same answers as $\mathbf{G}_3$.

Consider a legal last-touch query by signer ℓ on transcript sinf. Legality gives $\mathsf{m}_\ell = \mathsf{m}_{\ell,\mathsf{id}}$, $\ell \notin \mathcal{C}[\mathsf{sid}]$, and $\mathsf{Round}_{\ell,\mathsf{id}} = 1$. Hence $\ell \notin \mathsf{Touch}[\mathsf{sinf}]$ before the current query is answered. If $\#(\mathsf{Touch}[\mathsf{sinf}]) = \mathsf{t}-1$, then ℓ is the unique untouched signer in the threshold set. Every other honest coordinate in S is already touched and, by the definition of Touch, lies in $\mathcal{Q}_2[\mathsf{sinf}] \cup \mathcal{C}[\mathsf{sid}]$. This excludes $\mathsf{BadHonTouch}$.

If a non-current malicious coordinate i were uncorrupted, then $i \neq \ell$ and the last-touch condition would put i in $\mathsf{Touch}[\mathsf{sinf}]$. Since $i \notin \mathcal{C}[\mathsf{sid}]$, the definition of Touch gives $i \in \mathcal{Q}_2[\mathsf{sinf}]$. The earlier legal second-round answer on the exact transcript sinf required $\mathsf{m}_i = \mathsf{m}_{i,\mathsf{id}}$, so $i \in \mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}]$. This contradicts $i \in \mathsf{S}^{\mathsf{Mal}}[\mathsf{sinf}]$ and excludes $\mathsf{BadMalTouch}$. □

Lemma 14 (Lazy completion and direct posterior sampling). *The lazy completion records of $\mathbf{G}_7$ and the direct first-round posterior sampler of $\mathbf{G}_8$ are information-theoretic reorganizations of the same conditional distribution. Hence*

$$|pr_7 - pr_6| = 0 \qquad \textit{and} \qquad |pr_8 - pr_7| = 0.$$

Proof. Fix a registered identifier and a legal signer set S. The record stores the registered row and the hidden branch state from $\mathbf{G}_3$. Local first-round variables are left latent until a signing answer or corruption reveals them. The mask seed tuple in a local signing state is the already-fixed session seed for that signer

Alg $\mathsf{LastTouchSig}\,(\ell, \mathsf{sinf})$:

parse $\mathsf{sinf} = (\mathsf{vk}_\Sigma[\mathsf{sid}], \tau, \mathsf{id}, \mathsf{S}, \mathsf{m})$
parse $\tau = \{\widetilde{\mathsf{m}}_i\}_{i\in\mathsf{S}}$
$\mathcal{H} := \mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}]$; $\mathcal{M} := \mathsf{S}^{\mathsf{Mal}}[\mathsf{sinf}]$
$x_{\mathcal{M}} := \sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i} x_{i,\mathsf{sid}}$; $\mathsf{vk}_\Sigma^{\mathcal{H}} := \mathsf{vk}_\Sigma[\mathsf{sid}] - \mathsf{F}\,(x_{\mathcal{M}})$
for $b \in \{0,1\}$:
  for $i \in \mathsf{S}$:
    parse $\widetilde{\mathsf{m}}_i[b] = \widehat{\mathsf{c}}_{i,b}$
  $\mathsf{c}_b := \sum_{i\in\mathsf{S}} \lambda_{\mathsf{S},i}\widehat{\mathsf{c}}_{i,b}$
  parse $\mathsf{c}_b = (\mathsf{x}_b, \mathsf{r}_b)$
  $\mathsf{c}_b^{\mathcal{H}} := \sum_{i\in\mathcal{H}} \lambda_{\mathsf{S},i}\widehat{\mathsf{c}}_{i,b} + \sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i}\Delta_i^{\mathsf{c}}\,(\mathsf{id}, \mathsf{S}, b)$
  parse $\mathsf{c}_b^{\mathcal{H}} = \left(\mathsf{x}_b^{\mathcal{H}}, \mathsf{r}_b^{\mathcal{H}}\right)$
for $b \in \{0,1\}$:
  for $Y \in \{X, R, \rho\}$:
    $\Delta_b^{Y,\mathcal{H}} := -\sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i}\Delta_i^{Y}\,(\mathsf{sinf}, b)$
  $\Delta_b^{\omega,\mathcal{H}} := -\sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i}\Delta_i^{\omega}\,(\mathsf{sinf}, b)$
$\mathsf{h} := \mathsf{H}_{\mathbf{H}}\,(\mathsf{vk}_\Sigma[\mathsf{sid}], \mathsf{x}_0, \mathsf{r}_0, \mathsf{x}_1, \mathsf{r}_1, \mathsf{id}, \mathsf{S}, \mathsf{m})$
$\left(a_{b_{\mathsf{id}}}^{\mathcal{H}}, \left\{X_b^{\mathcal{H}}, \omega_b^{X,\mathcal{H}}, \omega_b^{X,\mathcal{H}}\right\}_{b\in\{0,1\}}\right)$
  $:= \mathsf{OpenXAgg}\left(\mathsf{CRec}[\mathsf{id}], \mathsf{vk}_\Sigma^{\mathcal{H}}, \left\{\mathsf{x}_b^{\mathcal{H}}\right\}_{b\in\{0,1\}}, b_{\mathsf{id}}\right)$
$\left(z_{b_{\mathsf{id}}}^{\mathcal{H}}, R_{b_{\mathsf{id}}}^{\mathcal{H}}, \omega_{b_{\mathsf{id}}}^{R,\mathcal{H}}\right) := \mathsf{OpenRAgg}\left(\mathsf{CRec}[\mathsf{id}], \mathsf{r}_{b_{\mathsf{id}}}^{\mathcal{H}}, \perp, b_{\mathsf{id}}\right)$
$\rho_{b_{\mathsf{id}}}^{\mathcal{H}} := z_{b_{\mathsf{id}}}^{\mathcal{H}} - a_{b_{\mathsf{id}}}^{\mathcal{H}}\,\mathsf{h}$
$\rho_{1-b_{\mathsf{id}}}^{\mathcal{H}} := \mathsf{Resp}\,(\mathsf{CRec}[\mathsf{id}], 1-b_{\mathsf{id}}, \mathsf{sinf})$
$R_{1-b_{\mathsf{id}}}^{\mathcal{H}} := \mathsf{F}\left(\rho_{1-b_{\mathsf{id}}}^{\mathcal{H}}\right) + X_{1-b_{\mathsf{id}}}^{\mathcal{H}}\,\mathsf{h}$
$\omega_{1-b_{\mathsf{id}}}^{R,\mathcal{H}} := \mathsf{OpenRAgg}\left(\mathsf{CRec}[\mathsf{id}], \mathsf{r}_{1-b_{\mathsf{id}}}^{\mathcal{H}}, R_{1-b_{\mathsf{id}}}^{\mathcal{H}}, 1-b_{\mathsf{id}}\right)$
for $b \in \{0,1\}$:
  $\omega_b^{\mathcal{H}} := \left(\omega_b^{X,\mathcal{H}}, \omega_b^{R,\mathcal{H}}\right)$
for $b \in \{0,1\}$:
  $\widehat{X}_{\ell,b} := \lambda_{\mathsf{S},\ell}^{-1}\left(X_b^{\mathcal{H}} + \Delta_b^{X,\mathcal{H}} - \sum_{i\in\mathcal{H}\setminus\{\ell\}} \lambda_{\mathsf{S},i}\widehat{X}_{i,b}\right)$
  $\widehat{R}_{\ell,b} := \lambda_{\mathsf{S},\ell}^{-1}\left(R_b^{\mathcal{H}} + \Delta_b^{R,\mathcal{H}} - \sum_{i\in\mathcal{H}\setminus\{\ell\}} \lambda_{\mathsf{S},i}\widehat{R}_{i,b}\right)$
  $\widehat{\rho}_{\ell,b} := \lambda_{\mathsf{S},\ell}^{-1}\left(\rho_b^{\mathcal{H}} + \Delta_b^{\rho,\mathcal{H}} - \sum_{i\in\mathcal{H}\setminus\{\ell\}} \lambda_{\mathsf{S},i}\widehat{\rho}_{i,b}\right)$
  $\widehat{\omega}_{\ell,b} := \lambda_{\mathsf{S},\ell}^{-1}\left(\omega_b^{\mathcal{H}} + \Delta_b^{\omega,\mathcal{H}} - \sum_{i\in\mathcal{H}\setminus\{\ell\}} \lambda_{\mathsf{S},i}\widehat{\omega}_{i,b}\right)$
return $\left\{\widehat{X}_{\ell,b}, \widehat{R}_{\ell,b}, \widehat{\rho}_{\ell,b}, \widehat{\omega}_{\ell,b}, \mathsf{c}_b\right\}_{b\in\{0,1\}}$

Fig. 18. Last-touch helper used by the hybrid second-round signing oracle.

and is copied into $\mathsf{LocalState}$ rather than sampled by the posterior record. Already visible values are cached and are never resampled. A latent commitment record contains the visible commitment string and the conditional constraints on the still-hidden committed value and opening. It does not attach a TCom trace at the time the first-round message is returned. When a trace is first needed, the completion sampler fixes a source representative ω_R and opening ω such that the already cached string equals $\mathsf{Com}\,(\mathsf{ck}, \mathsf{Rep}_{\mathsf{LFF}}\,(\omega_R)\,; \omega)$, and then runs $\mathsf{TCom}\,(\mathsf{ck}, \mathsf{tk}, \mathsf{Rep}_{\mathsf{LFF}}\,(\omega_R)\,, \omega_R; \omega)$ to attach a trace to that same public commitment string. The source representative and trace are cached for all later openings. The public first-round commitment string itself is never changed after it has been returned or used in $\mathsf{H}_{\mathbf{H}}$.

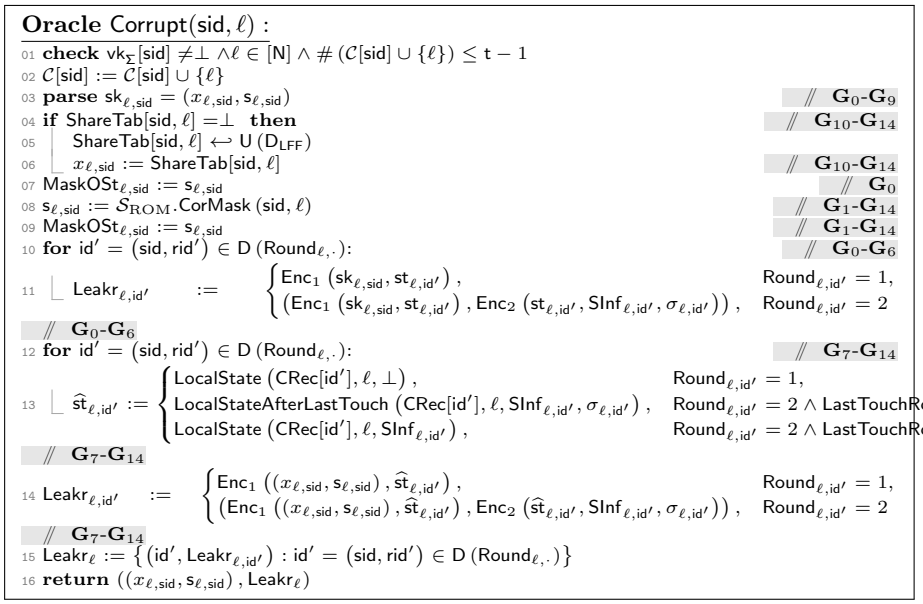


Fig. 19. Hybrid corruption oracle.

The only samplers used after a record is created are those listed in section C.1. FirstRoundVisible returns exactly the masked commitment pair returned by Sig_0 and records only the visible equations from lemma 9. It does not fix unmasked branch variables, long-term shares, first-round mask coordinates, or commitment traces. LocalState opens those decompositions only when the proof must explain an ordinary Sig_1 call or corruption. It is not used on a last-touch row before LastTouchSig returns its residual share. If such a row is corrupted later, LocalStateAfterLastTouch opens the current signer state from the joint fiber of lemma 10. OpenXAgg fixes the honest aggregate X values and the one image preimage $a^{\mathcal{H}}_{b_{\mathsf{id}}}$ needed on the hidden branch. It opens the hidden aggregate X commitment in source mode and retargets only the complementary aggregate X commitment with $\mathsf{tk}^{X}_{\mathsf{id},1-b_{\mathsf{id}}}$ while preserving the commitment strings that were already visible. OpenRAgg has two disjoint modes. In source mode it opens a cached R commitment to an image value $R^{\mathcal{H}}_b = \mathsf{F}\left(z^{\mathcal{H}}_b\right)$ and then the last-touch procedure sets the unique response $\rho^{\mathcal{H}}_b = z^{\mathcal{H}}_b - a^{\mathcal{H}}_b \mathsf{h}$. In target mode it opens a cached simulation trace to $\mathsf{F}\left(\rho^{\mathcal{H}}_b\right) + X^{\mathcal{H}}_b \mathsf{h}$. These modes are selected explicitly in fig. 17 and fig. 19, so there is no additional implicit sampler in the proof.

For ordinary image-key source openings, the source representative is $\mathsf{Emb}_{\mathsf{LFF}}(r)$ for the first-round value $R = \mathsf{F}(r)$. For the simulated complementary branch in the last-touch rule, the aggregate traces are still latent when h is computed. The target value $X^{\mathcal{H}}_{1-b_{\mathsf{id}}}$ is sampled together with a proof-only representative $\omega^{X,\mathcal{H}}_{1-b_{\mathsf{id}}}$. This remains true after the public-key hop because the complementary branch absorbs the represented residual of the uniform codomain key. The later

R opening uses TCol with $\omega^{X,\mathcal{H}}_{1-b_{\mathsf{id}}}$, h, and $\rho^{\mathcal{H}}_{1-b_{\mathsf{id}}}$ to open the cached commitment to $\mathsf{F}\left(\rho^{\mathcal{H}}_{1-b_{\mathsf{id}}}\right) + X^{\mathcal{H}}_{1-b_{\mathsf{id}}}\mathsf{h}$.

Before a completion record is fully opened, legality of corruptions and the touch checks imply that fewer than t local coordinates of each affine profile have been fixed. Therefore the affine completion rule of lemma 1 samples the conditional completion by choosing non-pivot local coordinates uniformly and setting a pivot by the Lagrange equation. The affine rule is applied only to vector-space values: split preimages, Schnorr randomness, X images, and commitment openings. For each latent commitment, the sampler first fixes the underlying value, proof-only representative, and opening and then runs TCom under the registered key, storing the resulting trace as proof state. Thus commitments and traces are generated jointly from the same sampled opening and are never interpolated as independent affine coordinates. The profile-wise masking functionality of definition 4 gives the conditional completion law for every visible masked coordinate and preserves the zero-sum mask equations. The current signer in a last-touch row is handled by lemma 10. The local unmasked variables and the mask coordinates are sampled as one assignment satisfying the cached residual signature, rather than deriving either side from the other.

In $\mathbf{G}_7$, fig. 16 calls $\mathsf{LocalState}\,(\mathsf{CRec}[\mathsf{id}], \ell, \bot)$ immediately after $\mathsf{FirstRoundVisible}$. Thus the internal state is materialized at the same point as in the eager execution, and the game remains distributionally equivalent to that eager view. In $\mathbf{G}_8$, fig. 16 omits the immediate $\mathsf{LocalState}$ call. Only the visible first-round masked commitments are sampled, from the exact visible marginal of lemma 9, while the unmasked state remains latent. The conditional state for a later corruption includes the exact first-round message already returned to the adversary and, when $\mathsf{Round}_{\ell,\mathsf{id}} = 2$, the exact second-round share already returned on its transcript. Since $\mathsf{FirstRoundVisible}$ samples exactly the visible marginal and $\mathsf{LocalState}$ samples exactly the corresponding posterior, the $\mathbf{G}_7/\mathbf{G}_8$ hop is zero-distance. If $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 0$, opening the latent state later uses $\mathsf{LocalState}$ and gives the same law as the eager sampling in $\mathbf{G}_7$. If $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$, the state was never sampled before the residual answer and is opened by $\mathsf{LocalStateAfterLastTouch}$ from the joint fiber fixed by that answer. Finally, lemma 11 converts this opened proof state back to the random-tape leakage syntax of the security experiment, so the corruption oracle in fig. 19 does not reveal an extra object. Thus both hops are zero-distance. □

Lemma 15 (Aggregate last-touch signing). *The last-touch replacement in* $\boldsymbol{G}_9$ *changes the winning probability by at most*

$$2\mathsf{Q}_{\mathsf{Sig}_1}\varepsilon_{\mathsf{t}}.$$

Proof. Fix a legal last-touch query on $\mathsf{sinf} = (\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}], \tau, \mathsf{id}, \mathsf{S}, \mathsf{m})$ and write $b^\star = b_{\mathsf{id}}$ and $\bar{b}^\star = 1 - b_{\mathsf{id}}$. Let

$$\mathcal{H} := \mathsf{S}^{\mathsf{Hon}}[\mathsf{sinf}], \qquad \mathcal{M} := \mathsf{S}^{\mathsf{Mal}}[\mathsf{sinf}].$$

The touch checks of lemma 13 imply that ℓ is the unique untouched signer in $\mathcal{H}$. Every signer in $\mathcal{H} \setminus \{\ell\}$ is already touched by either a prior legal second-

round answer on the exact transcript sinf or by corruption. Every malicious non-current coordinate lies in $\mathcal{M} \subseteq \mathcal{C}[\mathsf{sid}]$ and is already opened by corruption. Those malicious coordinates remain adversarial transcript coordinates. The last-touch rule does not try to complete them.

In $\mathbf{G}_9$-$\mathbf{G}_{14}$ the current signer ℓ enters $\mathsf{LastTouchSig}$ without a preceding call to $\mathsf{LocalState}\,(\mathsf{CRec}[\mathsf{id}], \ell, \mathsf{sinf})$. The local branch variables of ℓ therefore remain latent while the residual second-round answer is returned. The row is marked $\mathsf{LastTouchRow}_{\ell,\mathsf{id}} = 1$, so a later corruption opens those local variables only with $\mathsf{LocalStateAfterLastTouch}$.

For each $i \in \mathcal{H} \setminus \{\ell\}$, the values denoted $\widehat{X}_{i,b}$, $\widehat{R}_{i,b}$, $\widehat{\rho}_{i,b}$, and $\widehat{\omega}_{i,b}$ are fixed under the current conditioning. If i was touched by a prior second-round answer on sinf, they are the cached returned coordinates. If i was touched only by corruption, then the corrupted view contains $\mathsf{sk}_{i,\mathsf{sid}}$ and the first-round state for id, and the mask opening state fixes the profile coordinates for (sinf, b) when that profile is initialized. Since $\widetilde{\mathsf{m}}_i = \mathsf{m}_{i,\mathsf{id}}$ for $i \in \mathcal{H}$, these data determine the canonical honest second-round coordinates on the exact transcript sinf without sampling a new value.

The malicious long-term shares are fixed under the current conditioning. Set

$$x_{\mathcal{M}} := \sum_{i \in \mathcal{M}} \lambda_{\mathsf{S},i} x_{i,\mathsf{sid}}, \qquad \mathsf{vk}_{\Sigma}^{\mathcal{H}} := \mathsf{vk}_{\Sigma}[\mathsf{sid}] - \mathsf{F}\,(x_{\mathcal{M}})\,.$$

The proof keeps a representative of $\mathsf{vk}_{\Sigma}^{\mathcal{H}}$ under the commitment basis. In the image-key games this representative is obtained from $x^{\mathcal{H}} := x[\mathsf{sid}] - x_{\mathcal{M}}$. After the public-key hop, the uniform residual public key is sampled together with such a representative. This representative is proof-only and is not returned to the adversary. Parse every supplied first-round coordinate as $\widetilde{\mathsf{m}}_i[b] = \widehat{\mathsf{c}}_{i,b}$ and set

$$\begin{aligned} \mathsf{c}_b &:= \sum_{i \in \mathsf{S}} \lambda_{\mathsf{S},i} \widehat{\mathsf{c}}_{i,b}, \\ \mathsf{c}_b^{\mathcal{H}} &:= \sum_{i \in \mathcal{H}} \lambda_{\mathsf{S},i} \widehat{\mathsf{c}}_{i,b} + \sum_{i \in \mathcal{M}} \lambda_{\mathsf{S},i} \Delta_i^{\mathsf{c}}\,(\mathsf{id}, \mathsf{S}, b)\,. \end{aligned}$$

Write $\mathsf{c}_b = (\mathsf{x}_b, \mathsf{r}_b)$ and $\mathsf{c}_b^{\mathcal{H}} = \left(\mathsf{x}_b^{\mathcal{H}}, \mathsf{r}_b^{\mathcal{H}}\right)$. The values c_0 and c_1 are the full first-round commitments used in

$$\mathsf{h} := \mathsf{H}_{\mathbf{H}}\,(\mathsf{vk}_{\Sigma}[\mathsf{sid}], \mathsf{x}_0, \mathsf{r}_0, \mathsf{x}_1, \mathsf{r}_1, \mathsf{id}, \mathsf{S}, \mathsf{m})\,.$$

The challenger implements the aggregate step by the proof procedures $\mathsf{OpenXAgg}$, $\mathsf{OpenRAgg}$, and Resp from section C.1. By corollary 1, the aggregate X-opening step is simulatable without the hidden X trapdoor. The procedure samples $a_{b^\star}^{\mathcal{H}}$ together with $\left\{X_b^{\mathcal{H}}, \omega_b^{X,\mathcal{H}}, \omega_b^{X,\mathcal{H}}\right\}_{b \in \{0,1\}}$. The sampled assignment opens $\mathsf{x}_{b^\star}^{\mathcal{H}}$ in source mode under $\mathsf{ck}_{\mathsf{id},b^\star}^{X}$, opens $\mathsf{x}_{\bar{b}^\star}^{\mathcal{H}}$ by TCol under $\mathsf{tk}_{\mathsf{id},\bar{b}^\star}^{X}$ with $\rho = 0$ and $c = 1$, and satisfies $X_{b^\star}^{\mathcal{H}} = \mathsf{F}\left(a_{b^\star}^{\mathcal{H}}\right)$, and satisfies $X_0^{\mathcal{H}} + X_1^{\mathcal{H}} = \mathsf{vk}_{\Sigma}^{\mathcal{H}}$. On branch $b^\star$, the R commitment is opened in its ordinary source mode. Namely,

$\mathsf{OpenRAgg}\left(\mathsf{CRec}[\mathsf{id}], \mathsf{r}^{\mathcal{H}}_{b^\star}, \perp, b^\star\right)$ samples $\left(z^{\mathcal{H}}_{b^\star}, R^{\mathcal{H}}_{b^\star}, \omega^{R,\mathcal{H}}_{b^\star}\right)$ from the conditional first-round fiber satisfying

$$R^{\mathcal{H}}_{b^\star} = \mathsf{F}\left(z^{\mathcal{H}}_{b^\star}\right), \qquad \mathsf{Com}\left(\mathsf{ck}^{R}_{\mathsf{id},b^\star}, R^{\mathcal{H}}_{b^\star}; \omega^{R,\mathcal{H}}_{b^\star}\right) = \mathsf{r}^{\mathcal{H}}_{b^\star}.$$

It then sets

$$\rho^{\mathcal{H}}_{b^\star} := z^{\mathcal{H}}_{b^\star} - a^{\mathcal{H}}_{b^\star}\mathsf{h}.$$

Therefore branch $b^\star$ uses an ordinary opening of the commitment string that was fixed before h. It does not use a trapdoor opening.

On branch $\bar{b}^\star$, Resp samples $\rho^{\mathcal{H}}_{\bar{b}^\star}$ from the conditional response fiber for sinf and the current immutable view. If no immutable response constrains this branch, this fiber is $\mathsf{D}_{\mathsf{LFF}}$. The challenger defines

$$R^{\mathcal{H}}_{\bar{b}^\star} := \mathsf{F}\left(\rho^{\mathcal{H}}_{\bar{b}^\star}\right) + X^{\mathcal{H}}_{\bar{b}^\star}\mathsf{h},$$

The commitment string $\mathsf{r}^{\mathcal{H}}_{\bar{b}^\star}$ is already fixed before h is evaluated. The rule does not change this string. It materializes the still-latent aggregate trace with the source representative of the cached first-round R value. The target $X^{\mathcal{H}}_{\bar{b}^\star}$ has a proof-only representative $\omega^{X,\mathcal{H}}_{\bar{b}^\star}$ from the conditional sampler. The call to TCol with $\omega^{X,\mathcal{H}}_{\bar{b}^\star}$, h, and $\rho^{\mathcal{H}}_{\bar{b}^\star}$ therefore opens the same cached commitment to $X^{\mathcal{H}}_{\bar{b}^\star}\mathsf{h} + \mathsf{F}\left(\rho^{\mathcal{H}}_{\bar{b}^\star}\right)$. The proof-only trace used here is the first materialized trace of this aggregate commitment and is cached for all later openings. Thus the honest-transcript residual satisfies

$$X^{\mathcal{H}}_0 + X^{\mathcal{H}}_1 = \mathsf{vk}^{\mathcal{H}}_{\Sigma}, \qquad R^{\mathcal{H}}_b = \mathsf{F}\left(\rho^{\mathcal{H}}_b\right) + X^{\mathcal{H}}_b\mathsf{h} \quad \text{for } b \in \{0,1\}.$$

It remains to derive the current signer's masked contribution. For $Y \in \{X, R, \rho\}$, let

$$\Delta^{Y,\mathcal{H}}_b := -\sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i}\Delta^{Y}_i\left(\mathsf{sinf}, b\right), \qquad \Delta^{\omega,\mathcal{H}}_b := -\sum_{i\in\mathcal{M}} \lambda_{\mathsf{S},i}\Delta^{\omega}_i\left(\mathsf{sinf}, b\right).$$

These are the honest aggregate mask offsets forced by the zero-sum masking profiles. The challenger sets, for $Y \in \{X, R, \rho\}$,

$$\widehat{Y}_{\ell,b} := \lambda^{-1}_{\mathsf{S},\ell}\left(Y^{\mathcal{H}}_b + \Delta^{Y,\mathcal{H}}_b - \sum_{i\in\mathcal{H}\setminus\{\ell\}} \lambda_{\mathsf{S},i}\widehat{Y}_{i,b}\right),$$

and applies the same residual rule to $\omega^{\mathcal{H}}_b = \left(\omega^{X,\mathcal{H}}_b, \omega^{R,\mathcal{H}}_b\right)$ with offset $\Delta^{\omega,\mathcal{H}}_b$. The returned second-round answer contains these residual masked values together with the full aggregate first-round commitments c_b from the supplied transcript.

This construction preserves all values already visible in τ, every corruption opening, and every canonical touched honest share on sinf. It changes only latent current-signer second-round coordinates and honest aggregate commitment

openings. If the current signer is corrupted later, lemma 10 samples a local state and mask coordinates that reproduce the cached residual answer and all cached first-round commitments. The statistical cost comes from one targetable opening for the complementary aggregate X commitment and one targetable opening for the simulated complementary R commitment. A union bound over these two calls and at most $\mathsf{Q}_{\mathsf{Sig}_1}$ second-round queries gives the stated bound. □

Lemma 16 (Uniform public keys and trapdoor setup). *The public-key replacement in $\boldsymbol{G}_{10}$ costs at most $\varepsilon_{\mathsf{ki}}(\lambda, \mathsf{Q}_{\mathsf{Session}})$, and the parameter-generation replacement in $\boldsymbol{G}_{11}$ is exact.*

Proof. At the $\mathbf{G}_9$ point, each registered row has branch-specific X-commitment keys. For a last-touch row, the hidden aggregate branch b_{id} is maintained in source mode with $X^{\mathcal{H}}_{b_{\mathsf{id}}} = \mathsf{F}\left(a^{\mathcal{H}}_{b_{\mathsf{id}}}\right)$, while the complementary branch remains simulation-openable. After $\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}]$ is replaced by a uniform codomain point, the complementary branch absorbs the residual $X^{\mathcal{H}}_{1-b_{\mathsf{id}}} = \mathsf{vk}^{\mathcal{H}}_{\mathbf{\Sigma}} - X^{\mathcal{H}}_{b_{\mathsf{id}}}$ through $\mathsf{OpenXAgg}$. Hence the aggregate first-round X commitments do not force the simulator to maintain an image-key relation $X^{\mathcal{H}}_0 + X^{\mathcal{H}}_1 = \mathsf{F}\left(x^{\mathcal{H}}\right)$ after the public-key hop.

The master witness $x[\mathsf{sid}]$ is visible only through $\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}] = \mathsf{F}(x[\mathsf{sid}])$ and through below-threshold corrupted shares. The completion records and aggregate last-touch state use the registered public key, registered branch state, and below-threshold local conditional laws. They do not read the master witness itself. Replacing at most $\mathsf{Q}_{\mathsf{Session}}$ independent image public keys by uniform points in $\mathsf{CD}_{\mathsf{LFF}}$ is therefore bounded by LFF key-indistinguishability.

At the $\mathbf{G}_9/\mathbf{G}_{10}$ boundary, local long-term shares, local branch preimages, local X values, and local first-round commitment decompositions are latent unless they have already been opened by corruption or by an ordinary Sig_1 query. A cached first-round masked commitment is only a visible equation $\widehat{\mathsf{c}}_{\ell,b} = \mathsf{c}_{\ell,b} + \Delta^{\mathsf{c}}_{\ell}(\mathsf{id}, \mathsf{S}, b)$. It is not a binding opening to $\mathsf{F}(a_{\ell,b})$ that is already tied to a complete Shamir reconstruction. In $\mathbf{G}_{10}$ the table $\mathsf{ShareTab}[\mathsf{sid}, \ell]$ is sampled the first time a local share handle is needed, either by a local completion sampler or by a corruption, and the same entry is reused thereafter. The corruption bound ensures that fewer than t entries of this table are ever exposed to the adversary for a fixed session. This exposed marginal is identical in the image-key and uniform-key worlds by Shamir privacy. The hybrid therefore samples only the marginal distribution of opened shares and proof-only local handles, and never asserts that the uniform point $\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}] \in \mathsf{CD}_{\mathsf{LFF}}$ has a global witness in $\mathsf{D}_{\mathsf{LFF}}$.

The aggregate values used by signing are handled separately. In image-key games, an honest aggregate public-key residual may be represented by $x[\mathsf{sid}] - x_{\mathcal{M}}$. After $\mathbf{G}_{10}$, key generation samples the uniform public key together with a proof-only representative $\omega_{\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}]}$. The same proof role is played by the linear representative $\omega_{\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}]} - \mathsf{Emb}_{\mathsf{LFF}}(x_{\mathcal{M}})$ of the already fixed residual $\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}] - \mathsf{F}(x_{\mathcal{M}})$. This representative is proof-only. It is used by $\mathsf{OpenXAgg}$ and targetable opening calls, and it is never decoded into all t local long-term shares. Thus every later corruption is explained by the below-threshold Shamir marginal,

while every aggregate commitment and response is explained by the completion record's public-codomain conditional law.

Aggregate last-touch answers fixed before a later corruption are explained by the same completion record. Already visible values are reused. If the corrupted coordinate is the current signer of a last-touch row, $\mathsf{LocalStateAfterLastTouch}$ samples its local state and mask coordinates from lemma 10, conditioned on both the cached first-round masked equations and the cached residual second-round equations. Otherwise $\mathsf{LocalState}$ samples the ordinary conditional law. Neither procedure requires a global witness for the uniform session key.

For corruptions, at most $\mathsf{t}-1$ long-term shares are exposed in each session. Shamir privacy makes these shares uniform over $\mathsf{D}_{\mathsf{LFF}}$ even conditioned on an image public key, so the same lazy share handles can be used in the image and uniform-key worlds.

The $\mathbf{G}_{11}$ setup distribution has the same exposed $\mathsf{pp}_{\mathsf{LFF}}$ marginal as Setup, and the trapdoor $\mathsf{td}_{\mathsf{im}}$ is hidden and unread until $\mathbf{G}_{12}$. Hence this hop has distance zero. □

The branch-specific X keys are essential for this hop. The hidden branch remains a source image point, and the complementary branch is the only equivocal X branch. Thus public keys can become uniform without changing any public commitment string and without forcing corruptions to see more than the below-threshold Shamir marginal.

Lemma 17 (Forged-branch image abort). *Let* $\mathsf{Bad}_{\mathsf{avk}}$ *denote the event that the final forgery's session key is in* $\mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$. *The abort in* $\boldsymbol{G}_{12}$ *satisfies*

$$pr_{11} - pr_{12} \leq \frac{pr_{11}}{2} + \frac{\mathsf{Q}_{\mathsf{Session}} \#(\mathsf{F}(\mathsf{D}_{\mathsf{LFF}}))}{\#(\mathsf{CD}_{\mathsf{LFF}})}.$$

Moreover,

$$\Pr[\mathsf{Bad}_{\mathsf{avk}}] \leq \frac{\mathsf{Q}_{\mathsf{Session}} \#(\mathsf{F}(\mathsf{D}_{\mathsf{LFF}}))}{\#(\mathsf{CD}_{\mathsf{LFF}})},$$

and conditioned on $\neg\mathsf{Bad}_{\mathsf{avk}}$ *the image-branch abort loses at most a factor two.*

Proof. $\mathbf{G}_{12}$ only adds the final abort. From $\mathbf{G}_{10}$ onward, every session public key is sampled uniformly from $\mathsf{CD}_{\mathsf{LFF}}$, which gives the displayed union bound over sessions.

Condition on $\neg\mathsf{Bad}_{\mathsf{avk}}$ and on final verification accepting. The aggregate-key check gives

$$X_0^\star + X_1^\star = \mathsf{vk}_\Sigma[\mathsf{sid}^\star], \qquad \mathsf{vk}_\Sigma[\mathsf{sid}^\star] \notin \mathsf{F}(\mathsf{D}_{\mathsf{LFF}}).$$

Since $\mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$ is a linear subspace, at most one of $X_0^\star$ and $X_1^\star$ lies in the image. The hidden bit $b_{\mathsf{id}^\star}$ is uniform, and the execution is invariant under swapping the two branch labels for the final identifier together with replacing $b_{\mathsf{id}^\star}$ by

$1-b_{\mathsf{id}^\star}$. This invariance is preserved under the prior conditioning. Registered row generation in $\mathbf{G}_3$ samples the two X simulation keys and the two R simulation keys symmetrically, and the $\mathsf{Good}_{\mathsf{sim}}$ abort from lemma 12 checks both branch labels symmetrically. The hidden bit only determines which branch is source-mode and which branch is equivocal. For the fresh final identifier, the freshness argument below rules out an earlier last-touch query, and the remaining games before $\mathbf{G}_{12}$ do not expose $b_{\mathsf{id}^\star}$ or apply branch-asymmetric conditioning to that identifier. Therefore conditioning on no prior abort preserves the branch-swap invariance needed for the factor-two image-branch argument. Consequently

$$\Pr_{\mathbf{G}_{11}}\left[\mathbf{G}_{11}\text{ wins} \wedge \mathsf{Bad}_{\mathsf{Forg}} \wedge \neg\mathsf{Bad}_{\mathsf{avk}}\right] \leq \frac{\mathsf{pr}_{11}}{2}.$$

Therefore

$$\mathsf{pr}_{11} - \mathsf{pr}_{12} = \Pr_{\mathbf{G}_{11}}\left[\mathbf{G}_{11}\text{ wins} \wedge \mathsf{Bad}_{\mathsf{Forg}}\right] \leq \frac{\mathsf{pr}_{11}}{2} + \Pr\left[\mathsf{Bad}_{\mathsf{avk}}\right],$$

and the first displayed bound follows from the bound on $\Pr\left[\mathsf{Bad}_{\mathsf{avk}}\right]$.

The same freshness condition also implies that no last-touch second-round query has occurred for $\mathsf{id}^\star$. Otherwise the ordinary update to $\mathcal{Q}[\mathsf{id}^\star]$ after that query would put all t signers of the registered set inside $\mathcal{Q}[\mathsf{id}^\star] \cup \mathcal{C}[\mathsf{sid}^\star]$, contradicting freshness. Thus the simulation trapdoors stored for the final identifier have not been used by the aggregate last-touch rule of $\mathbf{G}_9$ and are not part of the final verification predicate. □

Lemma 18 (Binding-key switches and lossy endpoint). *Let* $Q_{\mathsf{reg}} := \mathsf{Q}_{\mathsf{Sig}_0} + Q_{\mathsf{H}} + 1$. *The X-key switch in $\boldsymbol{G}_{13}$ and the R-key switch in $\boldsymbol{G}_{14}$ satisfy*

$$|\mathit{pr}_{13} - \mathit{pr}_{12}| \leq Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + \varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right), \qquad |\mathit{pr}_{14} - \mathit{pr}_{13}| \leq Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}} + \varepsilon_{\mathsf{key}}\left(\lambda, Q_{\mathsf{reg}}\right).$$

The endpoint also satisfies

$$\mathit{pr}_{14} \leq 2\varepsilon_{\mathsf{b}}\left(\lambda, Q_{\mathsf{reg}}\right) + \left(Q_{\mathsf{H}} + \mathsf{Q}_{\mathsf{Sig}_1} + 1\right)\varepsilon_{\mathsf{I}}.$$

Proof. The $\mathbf{G}_{13}$ switch is global over all registered rows. For each row it replaces the hidden X branch key $\mathsf{ck}^X_{\mathsf{id},b_{\mathsf{id}}}$ from an SGen public component by a BGen key, and it keeps the complementary branch key $\mathsf{ck}^X_{\mathsf{id},1-b_{\mathsf{id}}}$ simulation-openable with $\mathsf{tk}^X_{\mathsf{id},1-b_{\mathsf{id}}}$. No final-row guessing is involved. The executable invariant from corollary 1 is that later $\mathsf{LastTouchSig}$ calls source-open the hidden X branch and call TCol only on the complementary X branch. Hence removing $\mathsf{tk}^X_{\mathsf{id},b_{\mathsf{id}}}$ from the row state does not change later signing.

The hidden X branch keys in $\mathbf{G}_{12}$ were sampled by SGen and also survived the $\mathsf{Good}_{\mathsf{sim}}$ abort introduced in $\mathbf{G}_3$. Introduce an intermediate game that no longer aborts on the hidden X branch key, while keeping the complementary X branch and both R branch keys subject to their previous good-key checks. A union bound over at most Q_{reg} registered rows bounds this conditioning-removal step by $Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}}$. From that intermediate game, the vector of unconditioned hidden

X public keys over at most Q_{reg} rows is switched from SGen public components to BGen keys by one application of the multi-key indistinguishability property of definition 10.

In $\mathbf{G}_{14}$, the hidden-branch public key $\mathsf{ck}^{R}_{\mathsf{id},b_{\mathsf{id}}}$ is switched to BGen and no longer carries or uses a simulation trapdoor. The complementary branch $1-b_{\mathsf{id}}$ remains simulation-openable and good, which is the only R trapdoor still needed by later TCol calls. The executable invariant is that every later $\mathsf{LastTouchSig}$ call uses $\mathsf{OpenRAgg}\left(C, \mathsf{r}^{\mathcal{H}}_{b_{\mathsf{id}}}, \bot, b_{\mathsf{id}}\right)$ on the hidden branch. This source mode opens the already fixed commitment as an ordinary image value $R^{\mathcal{H}}_{b_{\mathsf{id}}} = \mathsf{F}\left(z^{\mathcal{H}}_{b_{\mathsf{id}}}\right)$ and sets $\rho^{\mathcal{H}}_{b_{\mathsf{id}}} = z^{\mathcal{H}}_{b_{\mathsf{id}}} - a^{\mathcal{H}}_{b_{\mathsf{id}}}\mathsf{h}$. It does not invoke TCol. The only target-mode call is on $1-b_{\mathsf{id}}$, whose key remains simulation-openable. This switch is also global over all registered rows and does not guess the final row. This does not introduce branch-asymmetric conditioning before the $\mathbf{G}_{12}$ image-branch abort.

As above, the hidden R branch keys in $\mathbf{G}_{13}$ are still conditioned on $\mathsf{Good}_{\mathsf{sim}}$. First remove this hidden-branch good-key conditioning, paying at most $Q_{\mathsf{reg}}\varepsilon_{\mathsf{g}}$ by a union bound over the registered rows. Then replace the resulting unconditioned hidden R public components from SGen by BGen using the multi-key indistinguishability property of definition 10. There are at most Q_{reg} rows in this switch.

It remains to bound pr_{14}. Let $b^{\star} := b_{\mathsf{id}^{\star}}$ for an accepting fresh forgery in $\mathbf{G}_{14}$. Since $\mathbf{G}_{12}$ has already aborted on $\mathsf{Bad}_{\mathsf{Forg}}$,

$$X^{\star}_{b^{\star}} \notin \mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right).$$

At the endpoint, the hidden branch keys $\mathsf{ck}^{X}_{\mathsf{id}^{\star},b^{\star}}$ and $\mathsf{ck}^{R}_{\mathsf{id}^{\star},b^{\star}}$ are generated by BGen. Define

$$X^{\star}_{\mathsf{ext}} := \mathsf{Ext}\left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^{X}_{\mathsf{id}^{\star},b^{\star}}, \mathsf{x}^{\star}_{b^{\star}}\right), \qquad R^{\star}_{\mathsf{ext}} := \mathsf{Ext}\left(\mathsf{pp}_{\mathsf{LFF}}, \mathsf{ck}^{R}_{\mathsf{id}^{\star},b^{\star}}, \mathsf{r}^{\star}_{b^{\star}}\right).$$

The final row is selected adaptively from the registered rows initialized before final verification. Fix any enumeration of these rows, so that $\mathsf{id}^{\star}$ determines the selected index in the binding experiment. Apply the adaptive Q-key statistical binding property of definition 10 to the vector of hidden-branch X binding keys and the adaptively selected pair $(\mathsf{id}^{\star}, \mathsf{x}^{\star}_{b^{\star}})$. Except with probability $\varepsilon_{\mathsf{b}}(\lambda, Q_{\mathsf{reg}})$, the accepted opening of $\mathsf{x}^{\star}_{b^{\star}}$ opens to $X^{\star}_{\mathsf{ext}}$. Apply the same property to the vector of hidden-branch R binding keys and the adaptively selected pair $(\mathsf{id}^{\star}, \mathsf{r}^{\star}_{b^{\star}})$. Except with another probability $\varepsilon_{\mathsf{b}}(\lambda, Q_{\mathsf{reg}})$, the accepted opening of $\mathsf{r}^{\star}_{b^{\star}}$ opens to $R^{\star}_{\mathsf{ext}}$. Outside these binding errors, $X^{\star}_{b^{\star}} = X^{\star}_{\mathsf{ext}}$ and $R^{\star}_{b^{\star}} = R^{\star}_{\mathsf{ext}}$. In particular, $X^{\star}_{\mathsf{ext}} \notin \mathsf{F}\left(\mathsf{D}_{\mathsf{LFF}}\right)$.

The values $X^{\star}_{\mathsf{ext}}$ and $R^{\star}_{\mathsf{ext}}$ are fixed when the eventual final $\mathsf{H}_{\mathbf{H}}$ input is first initialized. Indeed, from $\mathbf{G}_3$ onward a direct $\mathsf{H}_{\mathbf{H}}$ query whose input names a registered $(\mathsf{vk}_{\mathbf{\Sigma}}[\mathsf{sid}], \mathsf{id})$ first initializes the corresponding registered H row if it has not already been initialized. Thus the keys used by the deterministic extractors are fixed before the $\mathsf{H}_{\mathbf{H}}$ answer is sampled, and $\mathsf{x}^{\star}_{b^{\star}}$ and $\mathsf{r}^{\star}_{b^{\star}}$ are part of that input. Let $\mathsf{h}^{\star}$ be the corresponding random-oracle value. Final verification checks

$$R^{\star}_{\mathsf{ext}} = \mathsf{F}\left(\rho^{\star}_{b^{\star}}\right) + X^{\star}_{\mathsf{ext}}\mathsf{h}^{\star}.$$

Equivalently, with $c^\star := -\mathsf{h}^\star$ and $s^\star := \rho^\star_{b^\star}$,

$$\mathsf{F}(s^\star) - X^\star_{\mathsf{ext}} c^\star = R^\star_{\mathsf{ext}}.$$

Partition the remaining accepting executions by the first initialization of this final $\mathsf{H}_{\mathbf{H}}$ input. This first initialization can occur in at most Q_{H} direct $\mathsf{H}_{\mathbf{H}}$ queries, at most $\mathsf{Q}_{\mathsf{Sig}_1}$ internal second-round signing queries, or during final verification. For each fixed first-initialization event, before the random-oracle value $\mathsf{h}^\star$ is sampled, the extracted values $X^\star_{\mathsf{ext}}$ and $R^\star_{\mathsf{ext}}$ are fixed because $\mathsf{x}^\star_{b^\star}$ and $\mathsf{r}^\star_{b^\star}$ are part of the $\mathsf{H}_{\mathbf{H}}$ input and Ext is deterministic. Since $\mathbf{G}_{12}$ ensures $X^\star_{b^\star} \notin \mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$ and binding gives $X^\star_{\mathsf{ext}} = X^\star_{b^\star}$, the lossy-soundness predicate applies with

$$X := X^\star_{\mathsf{ext}}, \qquad R := R^\star_{\mathsf{ext}}, \qquad c := -\mathsf{h}^\star, \qquad s := \rho^\star_{b^\star}.$$

A union bound over the at most $Q_{\mathsf{H}} + \mathsf{Q}_{\mathsf{Sig}_1} + 1$ first-initialization events gives the displayed $\left(Q_{\mathsf{H}} + \mathsf{Q}_{\mathsf{Sig}_1} + 1\right)\varepsilon_{\mathsf{I}}$ term. This is an analysis partition and introduces no inverse guessing loss. □

D Proofs for the MDDH Commitment Instantiation

Proof. We prove lemma 3. Linearity follows coordinatewise. The image-membership trapdoor has the same public marginal as ordinary setup, and the displayed test is exact because a codomain point lies in the image exactly when each pair is a DDH pair with respect to the public base $[\alpha_j]$.

Key indistinguishability is precisely the standard DDH distribution applied independently to the k coordinate pairs of each sampled LFF image. A hybrid over the kQ pairs converts any LFF key distinguisher into a distinguisher for the corresponding multi-instance DDH distribution.

For lossy soundness, fix any parameters, any $X \notin \mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$, and any $R \in \mathsf{CD}_{\mathsf{LFF}}$. Let π be the quotient map from $\mathsf{CD}_{\mathsf{LFF}}$ to $\mathsf{CD}_{\mathsf{LFF}}/\mathsf{F}(\mathsf{D}_{\mathsf{LFF}})$. Since $\pi(X) \neq 0$, the equation

$$\mathsf{F}(s) - c \cdot X = R$$

implies $-c \cdot \pi(X) = \pi(R)$ in the quotient space. There is at most one scalar $c \in \mathbb{Z}_q$ satisfying this equation. Therefore a uniformly sampled challenge is answerable with probability at most $1/q$, even for an unbounded adversary that chooses s after seeing c. □

Proof. We prove lemma 4. Homomorphism follows by expanding the two commitment components. For any scalars α_i,

$$\sum_i \alpha_i \mathsf{Com}\left(\mathsf{ck}, [\mathbf{R}]_i; \mathbf{U}_i, \mathbf{V}_i\right) = \mathsf{Com}\left(\mathsf{ck}, \sum_i \alpha_i [\mathbf{R}]_i; \sum_i \alpha_i \mathbf{U}_i, \sum_i \alpha_i \mathbf{V}_i\right).$$

Trace correctness follows from the representative equation $[\mathbf{R}] = [\mathbf{A}]\,\mathbf{D}$ and

$$[\mathbf{R}] + [\mathbf{A}]\,\mathbf{U} = [\mathbf{A}]\,(\mathbf{D} + \mathbf{U}).$$

Fix a good simulation key with $[\mathbf{Z}] = [\mathbf{Y}]\,\mathbf{T}$ and write

$$(\mathsf{c}, \mathbf{D}) = \mathsf{TCom}\left(\mathsf{ck}, \mathsf{tk}, [\mathbf{R}], \mathbf{D}_{\mathsf{src}}; \mathbf{U}, \mathbf{V}\right), \qquad \mathbf{L} := \mathbf{U} + \mathbf{T}\mathbf{V}.$$

Then $\mathbf{D} = \mathbf{D}_{\mathsf{src}} + \mathbf{U}$ and $\mathsf{c} = \left([\mathbf{A}]\,\mathbf{D}, [\mathbf{Y}]\,\mathbf{L}\right)$. For a target representative $\mathbf{D}_X$, challenge c, and response ρ, set

$$\mathbf{D}_{\mathsf{tar}} := \widehat{\rho} + \mathbf{D}_X c.$$

Since

$$\mathsf{Rep}_{\mathsf{LFF}}\left(\mathbf{D}_{\mathsf{tar}}\right) = \mathsf{F}\left(\rho\right) + \mathsf{Rep}_{\mathsf{LFF}}\left(\mathbf{D}_X\right) c,$$

the targetable-opening algorithm sets

$$\mathbf{U}' = \mathbf{D} - \mathbf{D}_{\mathsf{tar}}, \qquad \mathbf{V}' = \mathbf{T}^{-1}\left(\mathbf{L} - \mathbf{U}'\right).$$

The two commitment components for the target message are

$$\begin{aligned} \mathsf{Rep}_{\mathsf{LFF}}\left(\mathbf{D}_{\mathsf{tar}}\right) + [\mathbf{A}]\,\mathbf{U}' &= [\mathbf{A}]\,\mathbf{D}, \\ [\mathbf{Y}]\,\mathbf{U}' + [\mathbf{Z}]\,\mathbf{V}' &= [\mathbf{Y}]\,\mathbf{L}. \end{aligned}$$

Hence the returned opening is in the posterior fiber. Conversely, any $(\bar{\mathbf{U}}, \bar{\mathbf{V}})$ in the fiber satisfies

$$[\mathbf{A}]\,\bar{\mathbf{U}} = [\mathbf{A}]\left(\mathbf{D} - \mathbf{D}_{\mathsf{tar}}\right).$$

The full rank of $\mathbf{A}$ gives $\bar{\mathbf{U}} = \mathbf{D} - \mathbf{D}_{\mathsf{tar}} = \mathbf{U}'$. The second commitment component and full column rank of $\mathbf{Y}$ then give $\bar{\mathbf{U}} + \mathbf{T}\bar{\mathbf{V}} = \mathbf{L}$. Since $\mathbf{T}$ is invertible, $\bar{\mathbf{V}} = \mathbf{V}'$. The fiber is therefore the singleton output by TCol. □

Proof. We prove lemma 5. The matrix $\bar{\mathbf{A}}$ has rank k. The setup samples $\mathbf{A}_\perp$ uniformly, so the projection of its columns to the quotient by the span of $\bar{\mathbf{A}}$ is a uniform $k \times k$ matrix. Hence $\operatorname{rank}(\mathbf{A}) \neq 2k$ with probability at most k/q. Under SGen, the matrix $\mathbf{Y}$ is uniform in $\mathbb{Z}_q^{4k\times 2k}$ and $\mathbf{T}$ is uniform in $\mathbb{Z}_q^{2k\times 2k}$. The standard rank bound and a union bound give

$$\Pr\left[\operatorname{rank}(\mathbf{A}) \neq 2k \vee \operatorname{rank}(\mathbf{Y}) \neq 2k \vee \operatorname{rank}(\mathbf{T}) \neq 2k\right] \leq 5k/q.$$

Let $\mathsf{ck}_j = \left([\mathbf{A}], [\mathbf{Y}], [\mathbf{Z}]_j\right)$ be binding keys. If

$$\operatorname{rank}\left(\mathbf{Y} \mid \mathbf{Z}_j\right) = 4k,$$

then the second commitment component uniquely determines the opening pair $(\mathbf{U}, \mathbf{V})$ for any commitment under ck_j. The first component then uniquely determines the committed group message. Hence an unbounded extractor solves this linear system and returns $[\mathbf{C}] - [\mathbf{A}]\,\mathbf{U}$. The binding error is contained in the event that the displayed rank condition fails for some $j \in [Q]$. For each fixed j, the concatenated matrix is uniform in $\mathbb{Z}_q^{4k\times 4k}$ under BGen. A union bound gives $4kQ/q$. □

Proof. We prove lemma 6. In binding mode, each $[\mathbf{Z}]_j$ is uniform in $\mathbb{G}^{4k\times 2k}$. In simulation mode, each $[\mathbf{Z}]_j$ is $[\mathbf{Y}]\,\mathbf{T}_j$ for a uniform $\mathbf{T}_j \in \mathbb{Z}_q^{2k\times 2k}$. Grouping the $2kQ$ challenge columns of a $2kQ$-instance $(2k, 4k)$-MDDH distribution into Q matrices gives exactly these two key distributions, with the independent $[\mathbf{A}]$ carried as side information. □

Proof. We prove theorem 4. Binding keys are uniform because BGen samples $\mathbf{Z}$ uniformly over $\mathbb{Z}_q^{4k\times 2k}$. The representative-sampling property follows from the full-row-rank event for $\mathbf{A}$, since a uniform $\mathbf{D} \in \mathbb{Z}_q^{2k\times k}$ maps to a uniform $\mathsf{Rep}_{\mathsf{LFF}}(\mathbf{D}) \in \mathsf{CD}_{\mathsf{LFF}}$. Linearity of $\mathsf{Rep}_{\mathsf{LFF}}$ gives the stated proof-only representatives for later linear combinations. Combine lemmas 4 to 6. □

E MDDH (Matrix Decisional Diffie-Hellman) Assumption

Let $\mathbb{G}$ be a cyclic group of prime order q with generator g.

Definition 14 ((k,ℓ)-MDDH). The (k,ℓ)-MDDH assumption with $\ell > k$ holds in $\mathbb{G}$ if, for every uniformly random $\mathbf{A} \leftarrow\!\!\$\ \mathsf{U}\left(\mathbb{Z}_q^{\ell\times k}\right)$ and every PPT adversary $\mathcal{A}$, the following advantage is negligible in the security parameter λ:

$$\mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{A},\mathbb{G}}(\lambda) := \left|\Pr\left[\mathcal{A}([\mathbf{A}], [\mathbf{A}\mathbf{w}]) = 1 \;\middle|\; \mathbf{w} \leftarrow\!\!\$\ \mathsf{U}\left(\mathbb{Z}_q^{k}\right)\right] - \Pr\left[\mathcal{A}([\mathbf{A}], [\mathbf{u}]) = 1 \;\middle|\; \mathbf{u} \leftarrow\!\!\$\ \mathsf{U}\left(\mathbb{Z}_q^{\ell}\right)\right]\right|.$$

Definition 15 (Q-instance (k,ℓ)-MDDH). The Q-instance (k,ℓ)-matrix decisional Diffie-Hellman assumption with $\ell > k$ holds in $\mathbb{G}$ if, for every PPT adversary $\mathcal{A}$, the following advantage is negligible in the security parameter λ:

$$\begin{aligned}&\mathsf{Adv}^{Q\text{-}\mathcal{D}_{k,\ell}\text{-}\mathsf{MDDH}}_{\mathcal{A},\mathbb{G}}(\lambda)\\ &:= \left|\Pr\left[\mathcal{A}([\mathbf{A}], \{[\mathbf{A}\mathbf{w}_j]\}_{j\in[Q]}) = 1 \;\middle|\; \mathbf{A} \leftarrow\!\!\$\ \mathsf{U}\left(\mathbb{Z}_q^{\ell\times k}\right),\ \{\mathbf{w}_j\}_{j\in[Q]} \leftarrow\!\!\$\ \mathsf{U}\left((\mathbb{Z}_q^{k})^{Q}\right)\right]\right.\\ &\left. - \Pr\left[\mathcal{A}([\mathbf{A}], \{[\mathbf{u}_j]\}_{j\in[Q]}) = 1 \;\middle|\; \mathbf{A} \leftarrow\!\!\$\ \mathsf{U}\left(\mathbb{Z}_q^{\ell\times k}\right),\ \{\mathbf{u}_j\}_{j\in[Q]} \leftarrow\!\!\$\ \mathsf{U}\left((\mathbb{Z}_q^{\ell})^{Q}\right)\right]\right|.\end{aligned}$$

The above multi-instance variant admits a tight random self-reduction to the standard single-instance assumption.

Theorem 5 (Random Self-Reducibility of (k,ℓ)-MDDH [EHK+13]). *For all integers k, ℓ, Q with $\ell > k$ and $Q > 1$, and every PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)(1^\lambda, k, \ell)$ and*

$$\mathsf{Adv}^{Q\text{-}\mathcal{D}_{k,\ell}\text{-}\mathsf{MDDH}}_{\mathcal{A},\mathbb{G}}(\lambda) \le \mathsf{Adv}^{\mathsf{MDDH}}_{\mathcal{B},\mathbb{G}}(\lambda) + \frac{1}{q-1}.$$