Balsamic: One-round Threshold UOV from GPU-Accelerated FHE

Jaehyung Kim[1,†], Charles Gouert[2], Jean-Luc Watson[2], and G. Edward Suh[2]

[1]Stanford University
jaehk@cs.stanford.edu

[2]NVIDIA
{cgouert,jwatson,esuh}@nvidia.com

Abstract

Universal thresholdizers offer one-round threshold signatures, but require evaluating an entire signer under fully homomorphic encryption (FHE). We present Balsamic, a complete homomorphic Unbalanced Oil and Vinegar (UOV) signer and its application to one-round threshold UOV signature. UOV's short salted target hash shifts the encrypted workload to finite-field linear solving, which we integrate with coin derivation and hashing in a packed discrete-CKKS circuit. Our paired-pivot solver selects both pivots before first-pivot normalization, then applies their coupled update with one inversion stage per pair and cubic field work. Pivot-success certificates enable encrypted retry selection without residual evaluation. In 2-out-of-3 semihonest threshold UOV signing, each partial signing (online) takes 10.7 seconds on NVIDIA RTX 6000 Ada and 7.77 seconds on NVIDIA GB300, with 5.25 MB per response. These measurements demonstrate that complete homomorphic signing can provide the computational core of one-round threshold signatures without intermediate openings or per-signature interactive preprocessing.

1 Introduction

Universal thresholdization offers a general route to one-round post-quantum threshold signatures: encrypt an ordinary signing key, evaluate signing homomorphically, and let each party return one decryption response [BGG+18, EY24]. Parties answer independently, and any authorized set yields an ordinary signature. The obstacle is evaluating the complete signer under fully homomorphic encryption (FHE). THED implements FHE-based Dilithium signing but decrypts its secret-derived commitment before challenge hashing, requiring two communication rounds [PPS26]. Retaining one round requires affordable encrypted hashing as well as signing.

Unbalanced Oil and Vinegar (UOV) offers a favorable computational structure. Fiat–Shamir schemes hash an internally generated commitment, and hash-based signatures repeatedly hash secret-derived values through chains and trees [DKL+18, HBD+22]. UOV instead hashes the message and a short salt; fixing the vinegar variables then reduces preimage sampling to a small-field linear solve [KPG99, BCD+24]. This limits encrypted hashing and shifts the main workload to batched field arithmetic, provided retries and hidden pivot choices can be evaluated efficiently.

We present Balsamic, a complete homomorphic UOV signer and its one-round threshold realization. Discrete CKKS combines hashing, coin derivation, field arithmetic, and encrypted control in one packed circuit. Packing and refresh costs motivate this choice beyond native exact-arithmetic support: published finite-field refresh benchmarks favor CKKS over BGV in

[†]Most of this work was conducted during an internship at NVIDIA.

their tested configurations [SS26]. Our solver discovers paired pivots before applying their coupled row transform; the sole threshold decryption releases the signature-or-abort output.

To our knowledge, BALSAMIC is one of the first implementations of a complete homomorphic signer for a post-quantum signature scheme within a one-round threshold-signature protocol.[1] Across three independent setups per platform, the online cost of each partial signing takes 10.7 seconds on NVIDIA RTX 6000 Ada and 7.77 seconds on NVIDIA GB300, including coin derivation, hashing, signing, and response generation. These results make complete homomorphic signing an executable computational core for one-round universal thresholdization.

1.1 Technical Overview

For a public message m, independently keyed derivations supply a salt s and K candidate vinegar vectors. The hash of $m\|s$ is their common target; each candidate induces an oil-linear system $A_i z_i = b_i$ of dimension o over $\mathbb{F}_q$. Corresponding coordinates occupy SIMD lanes. After parallel solving, an encrypted prefix computation selects the lowest-index successful candidate, or aborts if all K attempts fail.

Constant-time UOV signers obtain pivots by conditionally accumulating later rows, with each addition depending on the preceding pivot value [BCH+23]. Our encrypted prefix scan selects the first available row with a nonzero current-column entry. This row anchors $O(o)$ parallel two-row determinant tests on the unmodified two-column panel; a second scan selects the first available nonzero determinant. Thus second-pivot selection does not wait for first-pivot normalization or propagation. The same hidden selectors drive a coupled Gauss–Jordan update using the 2×2 adjugate formula and one inversion stage per pair. Inverse-or-zero masking makes a failed step an identity update and records failure.

Updates act through selected rows and carry the right-hand side, directly yielding the oil solution on success. Pivot-success bits certify nonsingularity, eliminating back substitution and residual evaluation. Section 3.3 proves these properties for general public pivot widths; the implementation uses pairs over a binary extension field.

A dealer encrypts the private signing state once and distributes threshold states. Parties independently evaluate the same tagged circuit and return responses that any authorized set can combine, without intermediate decryption or per-signature interactive preprocessing. Section 4 gives generic semi-honest and malicious compilers under their stated threshold-primitive assumptions, supporting general access structures and post-setup static corruption. A classical random-oracle reduction supplies bounded-signer UF-CMA from randomized UOV under the stated coin-derivation and key-expansion assumptions (Appendix A.2).

1.2 Contributions

Complete homomorphic signing We integrate UOV's short target hash, coin derivation, candidate-parallel field arithmetic, and hidden retry selection in a discrete-CKKS circuit. Reusable encrypted state supports one-round threshold signing without intermediate openings or per-signature interactive preprocessing.

Paired-pivot organization Encrypted first-nonzero selection anchors linearly many determinant tests and a coupled update, reducing sequential block-inversion stages from o to $\lceil o/2 \rceil$ while retaining cubic field work. The complete paired design is 4.77× faster than the parallel scalar baseline on RTX 6000 Ada, including their respective packing, refresh, and workspace schedules (Section 5.2).

[1] A very recent concurrent work [CCJK26] explores hash-based signatures.

Evaluated one-round threshold realization Our GPU semi-honest realization produces ordinary UOV signatures from independent party responses. Experiments on two platforms quantify complete warm party computation, initialization, and communication costs, demonstrating the computational feasibility of homomorphic signing within the universal-thresholdizer architecture.

1.3 Related Work

Threshold signatures Universal thresholdizers establish one-round feasibility from threshold FHE [BGG+18, ASY22, EY24]; THED makes Dilithium practical by moving challenge hashing between two encrypted phases [PPS26]. Specialized lattice protocols instead optimize scheme-specific interaction [GKS24, DKM+24]. For multivariate signatures, Cozzo–Smart thresholdize LUOV through interactive LSSS, while Celi–Escudero–Niot and Vinaigrette distribute UOV/MAYO solution sampling [CS19,CEN25,ABB+26]. Vinaigrette reports one message-dependent semi-honest round after preprocessing. These distributed solvers use masked openings and account for rank leakage; our FHE circuit keeps pivot choices and candidate outcomes encrypted through final selection.

Discrete and accelerated FHE Discrete CKKS provides packed arithmetic and functional bootstrapping [CKKS17, DMPS24, BCKS24, BKSS24, AKP25]. We combine these operations with hashing and secret-dependent elimination using Grafting's RNS-CKKS organization and GPU execution techniques [CCK+25,JKA+21,FWX+23,CKA25]. Advances in BGV bootstrapping, subfield packing, and CKKS-assisted BFV offer alternative backends [MHWW24, PP25, KSS24].

Homomorphic linear solving Nissim–Weinreb give encrypted block elimination with inverse-or-zero blocks, singular-block no-ops, and deferred corrections, using randomized mixing and interaction with the decryption-key holder [NW06]. Kluczniak implements exact finite-field Gaussian elimination under FHE [Klu22]; Hong–Yoo–Yoon implement encrypted scalar pivots and conditional row exchanges for fixed-point Gauss–Jordan inversion [HYY20]. Bouman–de Vreede give deterministic block-recursive LEU decomposition and, separately, a randomized preconditioned protocol with one final inversion [BdV18]. Division-free determinant/adjugate circuits provide another exact alternative [Ber84], whereas packed CKKS iterative inversion targets numerical systems [ALYY23, KCL25].

Fixed physical rows and deferred permutations have a GPU Gauss–Jordan antecedent in Anzt et al. [ADFQO19]; identity-plus-thin-factor composition appears in Bischof–Van Loan's Householder transformations [BV87]. Our contribution is the linked circuit organization: an encrypted first pivot anchors linear determinant testing, the second is selected before first-pivot normalization, and both drive a coupled update. Section 3.3 develops this mechanism and its generalization.

2 Preliminaries

We specify the signature and threshold interfaces, UOV algebra, and discrete-CKKS semantics used below. Appendices A.1 and A.3 give the security games; Appendix A.2 analyzes the bounded signer.

2.1 Threshold Signatures

Let λ be the security parameter and $\mathfrak{C} \in \{\mathrm{PPT}, \mathrm{QPT}\}$ the adversary class. Scheme algorithms are classical PPT and protocol queries are classical. The efficiently recognizable message domain $\mathcal{M}_\lambda$ has polynomially bounded encodings; all spaces and negligible bounds depend on λ.

Definition 1 (Digital signature)**.** *A signature scheme* $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *generates* $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, signs* $\sigma \leftarrow \mathsf{Sign}(sk, m) \in \mathsf{Sig}_\lambda \cup \{\perp\}$*, and deterministically checks* $\mathsf{Verify}(vk, m, \sigma) \in \{0, 1\}$*, rejecting* $\perp$*. For fixed* m*, let* $\varepsilon^{\mathrm{corr}}_{\Sigma,m} = \Pr[\mathsf{Verify}(vk, m, \sigma) = 0]$*, over key generation and signing. Correctness requires negligible*

$$\varepsilon^{\mathrm{corr}}_{\Sigma} = \max_{m \in \mathcal{M}_\lambda} \varepsilon^{\mathrm{corr}}_{\Sigma,m}. \tag{1}$$

UF-CMA bounds the probability that an adversary given vk and a signing oracle outputs a verifying $(m^\star, \sigma^\star)$ with $m^\star$ unqueried. Every legal query is recorded before execution, including aborts. Definition 5 adapts ordinary UF-CMA only by making this bookkeeping explicit [GMR88, BGG+18].

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ and let $\Gamma \subseteq 2^{\mathcal{P}}$ be an efficiently recognizable nontrivial monotone access structure, with unauthorized sets $\mathcal{U}_\Gamma = 2^{\mathcal{P}} \setminus \Gamma$.

Definition 2 (Threshold signature)**.** *A threshold signature* Π *has trusted-dealer setup*

$$(pp, vk, \{sk_i\}_{i \in \mathcal{P}}) \leftarrow \mathsf{TS.KeyGen}(1^\lambda, \mathcal{P}, \Gamma),$$

partial signing $\mu_i \leftarrow \mathsf{PartSign}(pp, sk_i, m)$*, deterministic partial verification* $\mathsf{PartVerify}(pp, i, m, \mu_i)$*, combination* $\sigma \leftarrow \mathsf{Combine}(pp, m, \{(i, \mu_i)\}_{i \in I})$*, and ordinary verification* Verify*. Public parameters contain* $(\mathcal{P}, \Gamma, vk)$*; private states may persist across invocations. Partial verification binds* (i, m) *and rejects* $\perp$*. For fixed* $I \in \Gamma, m \in \mathcal{M}_\lambda$*, let* $\varepsilon^{\mathrm{corr}}_{\Pi,I,m}$ *be the probability, over honest setup and one invocation per* $i \in I$*, that any response fails partial verification or their combination fails ordinary verification. Correctness requires* $\varepsilon^{\mathrm{corr}}_{\Pi} = \max_{I,m} \varepsilon^{\mathrm{corr}}_{\Pi,I,m}$ *negligible uniformly over supported polynomial-size instances.*

This combines general-access syntax with persistent signing states [BGG+18, BCK+22, EY24]; DKG can optionally replace the dealer. One round means that each party responds without another party's online message; asynchrony means its response distribution is independent of the eventual authorized set and arrival order. Identities are authenticated, and neither property guarantees delivery.

Both security games allow adaptive signing queries after an unauthorized coalition S is chosen once, after public setup. In the *semi-honest* game, the adversary receives S's complete states and prescribed future coins; every legal honest query (i, m) records $m \in \mathcal{Q}$ before execution. A win requires $\mathsf{Verify}(vk, m^\star, \sigma^\star) = 1$ and $m^\star \notin \mathcal{Q}$. Definition 6 adapts TS-UF-0 to general access structures and post-setup corruption [BCK+22, BGG+18], using THED's honest-service abstraction [PPS26].

In the *malicious* game, the adversary receives S's states and may deviate arbitrarily. Let T_m contain all distinct honest parties queried on m, including aborting requests. A win requires a verifying forgery with $S \cup T_{m^\star} \notin \Gamma$, so unauthorized target-message queries are permitted. Definition 7 generalizes TS-UF-1 [BCK+22, EY24]. We also require *partial-signature robustness*: even given all party states, the adversary cannot produce a different accepted evaluation value for a fixed party and message. Definition 10 compares canonical values used by combination, permitting different valid proof certificates for the same value [BGG+18, EY24].

2.2 Threshold FHE and the Universal Thresholdizer

Let $\mathcal{C}_\lambda$ be a recognizable family of canonically described deterministic total circuits $C : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$.

Definition 3 (Threshold FHE)**.** *Threshold setup produces* $(pk, evk, \{dk_i\}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{P}, \Gamma)$*. Encryption gives* $c_x \leftarrow \mathsf{Enc}_{pk}(x)$*; canonical deterministic evaluation gives* $c_C := \mathsf{Eval}_{evk}(C, c_x)$*; party* i *returns* $d_i \leftarrow \mathsf{PartDec}(dk_i, c_C)$*; and an authorized set obtains* $y := \mathsf{FinDec}(c_C, \{(i, d_i)\}_{i \in I})$*. Here* evk *is public and* dk_i *is private. Correctness requires negligible* $\varepsilon^{\mathrm{dec}}_{\mathsf{ThFHE}} = \max_{C,x,I} \Pr[y \neq C(x)]$ *over supported* C, x*, authorized* I*, and honest setup, encryption, and share randomness, including approximate-arithmetic and decoding errors.*

We require *transcript security*: exposed-state threshold IND-CPA after a post-setup coalition choice, adaptive correctness with prescribed corrupt-side coins exposed, and stateful joint simulation of honest shares from exact outputs $C(x)$. Simulation includes correlated public auxiliary data and repeated circuits and overlapping requested sets (Definitions 11–13). These extend threshold-FHE security to our exposure and multi-session model [BGG+18, ASY22, PS24]. Perfect evaluation correctness is available for suitable bounded-noise FHE constructions [Hal17]; our realization assumes the discrete-CKKS correctness condition below.

For malicious security, a universal thresholdizer provides $(pp, \{st_i\}) \leftarrow \mathsf{UT.Setup}(1^\lambda, 1^D, \mathcal{P}, \Gamma, x)$, $y_i \leftarrow \mathsf{UT.Eval}(pp, st_i, C)$, $\mathsf{UT.Verify}(pp, i, C, y_i)$, and $\mathsf{UT.Combine}(pp, C, \{(i, y_i)\}_{i\in I})$, for circuits of depth at most D [BGG+18, EY24]. Honest responses verify and combine to $C(x)$, except with errors $\varepsilon^{\mathrm{ver}}_{\mathsf{UT}}$ and $\varepsilon^{\mathrm{eval}}_{\mathsf{UT}}$. Verification binds (i, C) and rejects malformed responses and $\perp$; combination also rejects duplicate indices. The public parser $\mathsf{val}_{\mathrm{ut}}$ extracts deterministic canonical evaluation values, excluding potentially randomized proof certificates; combination uses these values after checking the complete responses.

Strong simulation covers setup, exposed states, and complete responses, including certificates, with correlated auxiliary data and post-setup static corruption. Its stateful simulator receives no $C(x)$ while S together with the honest parties queried on C is unauthorized, and only $C(x)$ once authorized. *Response robustness* excludes a different accepted value even given all evaluation states. Definitions 14 and 15 adapt the invalid/valid simulator split and canonical-value robustness of [BGG+18, EY24] to this access and corruption model.

2.3 UOV Signatures

UOV removes oil–oil monomials using a hidden linear change of coordinates [KPG99, BCD+24]. Fix $\mathbb{F} = \mathbb{F}_q$ of prime-power order and positive vinegar/oil dimensions v, o, with $v, o, \log q$ polynomially bounded; security is assumed for the selected parameter family. The verification key defines $\mathcal{P}_{vk} : \mathbb{F}^{v+o} \to \mathbb{F}^o$ and $q_{V,vk}(v') = \sum_{1\le a\le b\le v} v'_a v'_b P_{a,b}$, where $P_{a,b} \in \mathbb{F}^o$. The block trapdoor $\vartheta = (O, \{D_j\}_{j=1}^v)$ has $O \in \mathbb{F}^{v\times o}$, $D_j \in \mathbb{F}^{o\times o}$, and $\mathcal{L}_\vartheta(v') = \sum_j v'_j D_j$, satisfying

$$\mathcal{P}_{vk}((v' + Oz)\|z) = q_{V,vk}(v') + \mathcal{L}_\vartheta(v')z. \tag{2}$$

Writing $q_V = q_{V,vk}$ and $\mathcal{L} = \mathcal{L}_\vartheta$, signing solves

$$\mathcal{L}(v')z = \tau - q_V(v'), \qquad \tau = \mathsf{XOF}(m\|s) \in \mathbb{F}^o, \tag{3}$$

and returns the canonical encoding of $((v' + Oz)\|z, s)$ when $\mathcal{L}(v')$ is nonsingular. The XOF includes the fixed hash-to-field conversion. These identities hold in every characteristic; subtraction becomes addition in characteristic two.

Our deterministic bounded scheme $\mathsf{UOV}^{\mathrm{bd}} = (\mathsf{UOV.KeyGen}, \mathsf{Sign}_{\mathrm{bd}}, \mathsf{Verify})$ retains ordinary UOV verification. For a positive public polynomially bounded integer K, signing derives $s = \mathsf{Salt}_{\boldsymbol{k}_{\mathrm{s}}}(m) \in \mathcal{S}_\lambda = \{0,1\}^\ell$ and $v_i = \mathsf{Vinegar}_{k_{\mathrm{v}}}(m, s, i) \in \mathbb{F}^v$, returning the first nonsingular candidate or $\perp$ (Definition 4). The canonical counter space $\mathcal{I}_\lambda$ contains $0, \dots, K-1$; K counts all attempts, independently of batching.

For positive polynomially bounded integers b, h, ℓ, set $L = bh$ and require $2^\ell/3^L$ negligible. A bank of b independent uniform keys for $\mathsf{G}_k : \mathcal{M}_\lambda \to \mathbb{F}_3^h$ supplies L trits, converted to ℓ salt bits by Equation (5). The bank $\boldsymbol{k}_{\mathrm{s}}$, uniform vinegar key k_{v}, and ordinary key-generation coins are mutually independent. The private state is $\theta = (\boldsymbol{k}_{\mathrm{s}}, k_{\mathrm{v}}, \vartheta) \in \Theta_\lambda$, with $sk_{\mathrm{bd}} = (vk, \theta)$. Its canonical injective encoding $x_\theta = \mathsf{enc}_\theta(\theta) \in \{0,1\}^{\ell_\theta}$ has polynomial length and a total left-inverse parser dec_θ, using a fixed default on nonencodings. For s_σ-bit signatures, define $\mathsf{Encode}(\sigma) = (1, \sigma)$ and $\mathsf{Encode}(\perp) = (0, \sigma_{\mathrm{dum}})$ for fixed $\sigma_{\mathrm{dum}} \in \{0,1\}^{s_\sigma}$.

Security of bounded signing Theorem 5 establishes classical UF-CMA from random-coins UOV with full-rank rejection, published Toeplitz-CDM weak-PRF security, the independent vinegar PRF, and ordinary-key expansion pseudorandomness, in the public-preprocessing

random-oracle model. Together with Corollary 6, it supplies the base premises for thresholdization; Appendix A.2 gives the distributions, reductions, and bounds.

2.4 Discrete CKKS

CKKS offers packed approximate arithmetic, rescaling, and bootstrapping [CKKS17, CHK+18]; discrete CKKS assigns discrete values decoding basins and uses cleaning or functional bootstrapping to restore canonical representatives [DMPS24, BCKS24, BKSS24, AKP25]. For $\mathbb{F}_{2^k} = \mathbb{F}_2[X]/(p)$ with irreducible p of degree $k \geq 1$, our realization encodes coefficient bits by evaluation at the complex roots of the monic integer lift of p. Componentwise multiplication and the inverse transform compute products modulo that lift; coefficientwise reduction modulo two recovers the field product. Functional maps implement reduction, nonzero tests, and inversion; related finite-field CKKS arithmetic appears in [SS26].

Let $\mathsf{GoodDec}_C$ mean that every discrete wire stays in its decoding basin and every cleanup returns its intended representative, making the field computation exact. We assume this correctness adaptively for supported circuits, including bootstrapping, with negligible failure (zero for perfect correctness), counted in $\varepsilon^{\mathrm{dec}}_{\mathsf{ThFHE}}$. Section 5.1 separately states the coefficient-error and no-wrap premises for threshold decryption.

2.5 Zero-Knowledge Proofs

A noninteractive zero-knowledge proof can bind an evaluation value p_i to (pp, i, C) and a committed state producing it. The malicious compiler assumes the resulting complete-response simulation and value robustness, including the realization's completeness, zero knowledge, and soundness or extraction requirements [BGG+18, EY24].

3 Homomorphic UOV Signing

Our homomorphic signer batches $K \geq 1$ UOV attempts over $\mathbb{F} = \mathbb{F}_q$, constructs their oil systems, and solves them with encrypted pivot selectors and product-form row transforms. The construction supports arbitrary dimensions (v, o) and public pivot widths summing to o; the discrete-CKKS realization uses binary fields and paired pivots.

The total circuit $C_{vk,m} : \{0,1\}^{\ell_\theta} \to \{0,1\}^{1+s_\sigma}$ parses $\theta = \mathsf{dec}_\theta(x_\theta)$ and incorporates the public verification key vk and message m. We suppress ciphertext notation: state-dependent wires remain encrypted, while dimensions, indices, schedules, and routing are public.

3.1 Oblivious Batch Signing Circuit

Let $\mathsf{EncSig} : \mathbb{F}^{v+o} \times \{0,1\}^\ell \to \{0,1\}^{s_\sigma}$ denote canonical UOV signature encoding. The signing contract fixes the salt, ordered candidates, and first-success rule independently of packing.

Definition 4 (Bounded UOV signer)**.** *Let (vk, sk_{bd}) lie in the support of $\mathsf{UOV.KeyGen}(1^\lambda)$, with $sk_{\mathrm{bd}} = (vk, \theta)$. On input m, $\mathsf{Sign}_{\mathrm{bd}}$ derives the salt, target, and ordered vinegar vectors by the maps in Section 3.1, and forms (A_i, b_i) by Equations (4) and (6). If every A_i is singular, it returns $\perp$. Otherwise it sets*

$$i^\star = \min\{i : A_i \text{ is nonsingular}\}, \qquad z = A_{i^\star}^{-1} b_{i^\star},$$

and returns the canonical signature $\mathsf{EncSig}((v_{i^\star} + Oz) \,\|\, z\,;\, salt)$. The UOV trapdoor identity in Equation (2) implies that every non-$\perp$ output verifies under vk.

For each candidate i, the circuit constructs the oil system

$$A_i z_i = b_i, \qquad A_i = \mathcal{L}(v_i), \quad b_i = \tau - q_V(v_i), \tag{4}$$

where $z_i \in \mathbb{F}^o$ is the oil vector [KPG99, BCD+24]; subtraction equals addition in characteristic two. An encrypted output bit distinguishes a signature from $\perp$.

Arithmetic framework Our binary-field realization uses discrete CKKS to connect coefficient-bit hashing, ternary coin derivation, spectral field products, and functional reduction/inversion in one accelerated SIMD pipeline [DMPS24, BCKS24, BKSS24, SS26]. Representation, packing, and refresh jointly determine its cost. Other fields require suitable encodings and their own realization-correctness analysis.

Coin derivation Use the state $\theta = (\boldsymbol{k}_{\mathrm{s}}, k_{\mathrm{v}}, O, \{D_j\}_{j=1}^{v})$ and lengths $b, h, L = bh, \ell$ from Section 2.3. Let d be the CDM input dimension, $d_{\mathrm{s}} = 2d - 1$ the salt-key length, and d_{v} the vinegar-key length. Fix public $M \in \mathbb{F}_3^{h \times d}$ independently of signing keys, and publicly prepare $p_m = H_0(m) \in \{0,1\}^{\kappa}$ and $x_m = H_1(p_m) \in \mathbb{F}_2^d$. The Toeplitz-key CDM candidate with fixed random compression [BIP+18, DGH+21] computes $(T_{k_{\mathrm{s},a}})_{r,c} = (k_{\mathrm{s},a})_{d-1+r-c}$, $y^{(a)} = T_{k_{\mathrm{s},a}} x_m$ over $\mathbb{F}_2$, and $u^{(a)} = M\iota(y^{(a)}) \in \mathbb{F}_3^h$, where $0 \le r, c < d$ and ι embeds bits. This avoids the cyclic-key specialization targeted in [HLR+26]. Set $t_{ah+j} = u_j^{(a)}$ in key-major order and

$$\begin{aligned} \mathit{salt} &= \mathsf{SaltDerive}(\boldsymbol{k}_{\mathrm{s}}; x_m, M) \\ &= \mathsf{LE}_{\ell}\left(\sum_{j=0}^{L-1} t_j 3^j \bmod 2^{\ell}\right) \in \{0,1\}^{\ell}. \end{aligned} \tag{5}$$

Here LE_{ℓ} is little-endian encoding and $\mathsf{Radix}_{L,\ell}$ denotes the trit-to-bit map. Writing $t_j = a_j + 2b_j$ with $a_j, b_j \in \{0,1\}$ and $a_j b_j = 0$, a fixed carry network sums the selected public bit vectors of 3^j and $2 \cdot 3^j$ modulo 2^{ℓ}. This total conversion introduces no abort.

Toeplitz multiplication takes coordinates $d-1, \ldots, 2d-2$ of the length-$2d$ circular convolution of zero-padded k and x: each contributing index $d - 1 + r - c$ lies in $\{0, \ldots, 2d-2\}$, so no wraparound occurs. Public preprocessing enters as constants; all keyed operations are encrypted.

Fixed circuit-realizable maps $\mathsf{XOF} : \{0,1\}^* \to \mathbb{F}^o$ and $\mathsf{V} : \{0,1\}^{d_{\mathrm{v}}} \times \{0,1\}^{\kappa} \times \{0,1\}^{\ell} \times \{0, \ldots, K-1\} \to \mathbb{F}^v$ derive

$$\begin{aligned} \tau &= \mathsf{XOF}(m \parallel \mathit{salt}), \\ v_i &= \mathsf{V}(k_{\mathrm{v}}, p_m, \mathit{salt}, i), \qquad 0 \le i < K, \end{aligned}$$

These instantiate Salt and $\mathsf{Vinegar}$ from Section 2.3. Every candidate is evaluated, possibly across several ciphertexts or public batches, with its global order preserved and no secret-dependent stopping.

For $x, y \in \mathbb{F}_q$, use $\mathsf{NZ}(x) = x^{q-1}$, $\mathsf{Inv}(0) = 0$, $\mathsf{Inv}(x) = x^{-1}$ for $x \neq 0$, and $\mathsf{Eq}(x,y) = 1 - \mathsf{NZ}(x-y)$. Total inversion is x^{q-2} for $q > 2$ and x for $q = 2$. These fixed arithmetic circuits are exact in our algebraic model, with predicate outputs embedded as field bits.

3.2 Packed Oil-System Construction

For vinegar $x \in \mathbb{F}^v$, evaluate

$$\begin{aligned} \mathcal{L}(x) &= \sum_{j=1}^{v} x_j D_j, \\ q_V(x) &= \sum_{1 \le a \le b \le v} x_a x_b P_{a,b}. \end{aligned} \tag{6}$$

The coefficients $D_j \in \mathbb{F}^{o\times o}$ and $P_{a,b} \in \mathbb{F}^o$ are defined in Section 2.3; substituting $x = v_i$ constructs (A_i, b_i) in $O(vo^2 + v^2o)$ field operations. The binary-field realization uses coefficient/spectral representations, SIMD basis changes and Frobenius powers, and functional predicate/inversion maps [DMPS24, BCKS24, BKSS24, AKP25].

To define the logical SIMD layout, write $[K] = \{0, \dots, K-1\}$ and $[o] = \{0, \dots, o-1\}$. For $r, c \in [o]$, corresponding coordinates across candidates form the vectors

$$\mathbf{a}_{r,c} = \big((A_i)_{r,c}\big)_{i\in[K]} \in \mathbb{F}^K, \qquad \mathbf{b}_r = \big((b_i)_r\big)_{i\in[K]} \in \mathbb{F}^K. \tag{7}$$

Public positive widths $\boldsymbol{w} = (w_1, \dots, w_{N_{\text{step}}})$ summing to o partition columns into consecutive sets J_p, with $|J_{j-1}| = w_j$. Write $\mathcal{C}_p = ((A_i)_{*,J_p})_{i\in[K]}$ and $\mathcal{B} = (b_i)_{i\in[K]}$ for the panel and right-hand-side workspaces. Each operation acts componentwise across candidates; public routing must preserve these coordinates and arithmetic dependencies. Packing and refresh determine physical costs (Section 5).

3.3 Product-Form Block Gauss–Jordan

The solver returns encrypted $(1, A^{-1}b)$ for invertible $A \in \mathbb{F}^{o\times o}$ and $(0, 0^o)$ otherwise. A constant-time UOV baseline accumulates later rows into the pivot row until its diagonal becomes nonzero, then normalizes and eliminates [BCH+23]. This creates dependent full-row updates. We instead parallelize predicates and use first-nonzero prefix selection, keeping physical rows fixed and recording their logical order as in implicit pivoting [ADFQO19]. For a column $c \in \mathbb{F}^o$ and encrypted availability bits a_r, let $s_r = a_r\mathsf{NZ}(c_r)\prod_{u<r}(1 - a_u\mathsf{NZ}(c_u))$ and $\gamma = s^{\mathsf{T}}c$. With I_o the identity, the scalar transform is

$$G_1 = I_o + (s - c)\mathsf{Inv}(\gamma)s^{\mathsf{T}}. \tag{8}$$

On success $G_1 c = s$, but scalar elimination still inverts and propagates after every column. Paired pivots select both rows from one unchanged panel before a coupled update.

For a public width d, let $C \in \mathbb{F}^{o\times d}$ contain the next columns of the transformed system and $U = \{r : a_r = 1\}$ its unselected rows. Set $e_0 = 1$. At scan $j \in \{1, \dots, d\}$, let $D_r^{(j)}$ be the determinant with first $j-1$ rows $(s^{(h)})^{\mathsf{T}}C_{*,1:j}$, $1 \le h < j$, and last row $C_{r,1:j}$. Select

$$\begin{aligned} g_r^{(j)} &= e_{j-1}\left(a_r - \sum_{h<j} s_r^{(h)}\right)\mathsf{NZ}(D_r^{(j)}), \\ s_r^{(j)} &= g_r^{(j)}\prod_{u<r}(1 - g_u^{(j)}), \qquad e_j = \sum_{r\in[o]} s_r^{(j)}. \end{aligned} \tag{9}$$

Here $D_r^{(1)} = C_{r,1}$; each selector is one-hot or zero, and e_{j-1} gates off scans after failure. All determinants and adjugates use fixed polynomial-size division-free circuits, including on singular inputs.

For width two, let the first selector choose r_0 with $p = C_{r_0,1} \neq 0$. Scalar elimination would give each other available row the second entry

$$C_{r,2} - C_{r,1}p^{-1}C_{r_0,2} = p^{-1}\det\begin{bmatrix} C_{r_0,1} & C_{r_0,2} \\ C_{r,1} & C_{r,2}\end{bmatrix}. \tag{10}$$

Thus the same second-pivot predicate follows from a determinant test before first-pivot inversion, normalization, or propagation. Anchoring on the first selected row leaves $O(o)$ candidate determinants, rather than $\binom{o}{2}$ pair tests; both scans use the unchanged panel.

For each scan, prefixes $P_r^{(j)} = \prod_{u<r}(1 - g_u^{(j)})$ give $s_r^{(j)} = P_r^{(j)} - P_{r+1}^{(j)}$. A parallel prefix network has depth $\lceil \log_2 o \rceil$ after predicate formation; the d scans remain sequential. Set $S = [s^{(1)} \ \cdots \ s^{(d)}]$ and $a'_r = a_r - \sum_j s_r^{(j)}$.

One *pivot step* selects and eliminates these d columns. Define

$$\begin{aligned} B &= S^{\mathsf{T}} C, & \delta &= e_d = \mathsf{NZ}(\det B), \\ B^{\dagger} &= \mathsf{Inv}(\det B)\,\mathrm{adj}(B). \end{aligned} \tag{11}$$

Inverse-or-zero blocks and singular-block no-ops also appear in interactive encrypted elimination [NW06]; here selection is deterministic within a fixed circuit. For $d = 1$, $\mathrm{adj}([\xi]) = [1]$; for $d = 2$,

$$B^{\dagger} = \mathsf{Inv}(\Delta) \begin{bmatrix} b_{2,2} & -b_{1,2} \\ -b_{2,1} & b_{1,1} \end{bmatrix}, \qquad \Delta = b_{1,1}b_{2,2} - b_{1,2}b_{2,1}. \tag{12}$$

This uses one total field inversion; signs disappear in characteristic two. SIMD evaluation may invert candidate determinants before gathering, since one-hot selection and $\mathsf{Inv}(0) = 0$ preserve the result. Define

$$T = (S - C)B^{\dagger}, \qquad G = I_o + TS^{\mathsf{T}}. \tag{13}$$

Here $B \in \mathbb{F}^{d\times d}$ and $T, S \in \mathbb{F}^{o\times d}$. A singular B gives $G = I_o$; its failure bit will mask the final output despite any partial availability update. On success,

$$GC = C + (S - C)B^{-1}B = S. \tag{14}$$

Moreover, the matrix determinant lemma and $S^{\mathsf{T}}S = I_d$ give

$$\det(G) = \det(I_d + S^{\mathsf{T}}T) = \det(B^{-1}) \neq 0. \tag{15}$$

For a panel or right-hand-side workspace $W \in \mathbb{F}^{o\times w}$, apply

$$W \longmapsto GW = W + T(S^{\mathsf{T}}W), \tag{16}$$

The same hidden selectors gather pivot rows for this coupled update, without forming G, exchanging rows, or applying an intervening scalar transform.

Lemma 1 (Product-form pivot step)**.** *Let $1 \le d \le o$ be the current public pivot width, let $a_r = 1$ exactly for $r \in U$, let C_U be the restriction of C to U, and suppose field operations, NZ, and Inv are exact. The selectors in Equation (9) produce $\delta = 1$ if and only if $\mathrm{rank}(C_U) = d$. When $\delta = 1$, $S^{\mathsf{T}}S = I_d$, G is invertible, and $GC = S$. Moreover, if S_{pre} consists of distinct standard-basis columns supported on previously selected rows outside U, then $GS_{\mathrm{pre}} = S_{\mathrm{pre}}$, and multiplying an augmented system by G preserves its solution set. When $\delta = 0$, $G = I_o$.*

Proof. Suppose $\mathrm{rank}(C_U) = d$. Every first-j column restriction then has rank j. Inductively, the previously selected rows have a nonsingular leading $(j-1)\times(j-1)$ minor. Eliminate the first $j-1$ entries of each remaining row using that minor. If all resulting jth entries vanished, the first j columns would have rank at most $j-1$, a contradiction. Hence some remaining row extends the minor, and the first-true scan selects one. All d scans therefore succeed. Conversely, success gives a nonsingular selected $d \times d$ minor; a failed scan leaves a zero row in B and gates off later scans. This proves $\delta = e_d = \mathsf{NZ}(\det B)$ and, on success, $S^{\mathsf{T}}S = I_d$. Equations (14) and (15) give $GC = S$ and invertibility of G, which preserves the solution set. Disjoint supports give $S^{\mathsf{T}}S_{\mathrm{pre}} = 0$ and thus $GS_{\mathrm{pre}} = S_{\mathrm{pre}}$. On failure, $B^{\dagger} = 0$ gives $G = I_o$. □

Public schedule Independently of $\boldsymbol{w}$, positive public counts $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{N_{\mathrm{blk}}})$ partition the pivot steps into *composition blocks*. Starting with all rows available, $\mathsf{PF\text{-}Solve}_{\boldsymbol{w},\boldsymbol{\nu}}(A, b)$ updates later local pivot panels immediately and combines the ordered thin factors before updating distant panels and the right-hand side. Appendix A.5 specifies the recurrence and eager-execution invariant; the algebra parallels WY aggregation [BV87]. Singleton blocks apply each transform separately. Record $\eta = \prod_{j=1}^{N_{\mathrm{step}}} \delta_j$ and $S_{\mathrm{all}} = [S_1 \ \cdots \ S_{N_{\mathrm{step}}}]$ in original column order.

Cost For a fixed field and bounded widths, singleton composition costs $O(o^3)$ field operations per candidate: each $o \times h$ update costs $O(odh)$, dominating o prefix scans of $O(o \log o)$ work each. Growing widths additionally incur o determinants of each size $1, \ldots, d$, a selected-block determinant, and its adjugate. Pairing keeps these constant-size and reduces sequential selected-block inversion stages from o to $\lceil o/2 \rceil$, with a final singleton for odd o. These count logical stages, not lane-wise inverse evaluations. Compared with Berkowitz's $O(\log^2 o)$-depth determinant/adjoint circuits [Ber84], we prioritize cubic work and thin workspaces. Composition trades wider factors for fewer distant updates; Section 5.2 evaluates paired pivots with singleton composition.

At termination, with H the ordered product of all transforms and $b' = Hb$, return η and

$$\widetilde{z} = S_{\text{all}}^{\mathsf{T}} b', \qquad \widehat{z} = \eta \widetilde{z}. \tag{17}$$

The gather restores oil-variable order; η masks every failed candidate to zero.

Theorem 1 (Exactness of product-form elimination)**.** *Let $\mathbb{F}$ be any finite field, $A \in \mathbb{F}^{o \times o}$, and $b \in \mathbb{F}^o$. For every public positive pivot-width sequence $\boldsymbol{w}$ summing to o and every public composition schedule $\boldsymbol{\nu}$ partitioning its steps, exact evaluation of $\mathsf{PF\text{-}Solve}_{\boldsymbol{w},\boldsymbol{\nu}}(A, b)$ returns $(\eta, \widehat{z})$ satisfying*

$$\eta = \mathsf{NZ}(\det A), \qquad \widehat{z} = \begin{cases} A^{-1}b, & \eta = 1, \\ 0^o, & \eta = 0. \end{cases} \tag{18}$$

Proof. Expanding the product gives Equation (41). Induction proves the block-boundary invariant and equivalence to eager execution. After a successful prefix of pivot steps covering h columns, let H_h be the accumulated invertible transform and Π_h the permutation placing the selected rows first in selector order. Lemma 1 gives $\Pi_h H_h A = \left[\begin{smallmatrix} I_h & * \\ 0 & R_h \end{smallmatrix}\right]$, where R_h restricts to unselected rows and unprocessed columns. If A is nonsingular, so is R_h; its next d columns have rank d, and the next pivot succeeds. Thus nonsingularity implies every $\delta_j = 1$. Conversely, success at every step gives $HA = S_{\text{all}}$, where H is invertible and availability makes S_{all} a permutation matrix, so A is nonsingular. Therefore $\eta = \mathsf{NZ}(\det A)$. On success, gathering $S_{\text{all}}^{\mathsf{T}} Hb$ gives $A^{-1}b$; on failure, the output mask gives 0°. □

3.4 Algebraic Certification and First-Success Selection

The solver-success bit already certifies the unique solution, so first-success selection can use η_i directly. To formalize this fact, let $\widehat{z}_i \in \mathbb{F}^o$ be the totalized oil vector returned for candidate i, let $\eta_i \in \{0, 1\}$ be the balanced product of its pivot-success bits, and define, for analysis only, the original-system residual predicate

$$r_i = \prod_{r \in [o]} \mathsf{Eq}\big((A_i \widehat{z}_i)_r, (b_i)_r\big). \tag{19}$$

Theorem 1 gives $\widehat{z}_i = A_i^{-1} b_i$ whenever $\eta_i = 1$, and therefore

$$\eta_i r_i = \eta_i. \tag{20}$$

The signing circuit therefore uses η_i without recomputing residuals. Equation (20) shows that a residual gate cannot change acceptance, although r_i need not equal η_i for a singular but consistent system. Omitting the gate saves o^2 field multiplications, o equality tests, their conjunction, and storage of the original system per candidate. Packing and refresh determine the latency benefit.

The winner bits and aggregate-success bit are

$$\alpha_i = \eta_i \prod_{j<i} (1 - \eta_j), \qquad \beta = 1 - \prod_{j<K} (1 - \eta_j). \tag{21}$$

Lemma 2 (First success)**.** *For input bits η_i, Equation* (21) *produces output bits α_i and β such that $\beta = \sum_{i=0}^{K-1} \alpha_i$. If all η_i are zero then $\beta = 0$ and every $\alpha_i = 0$; otherwise $\beta = 1$ and $\alpha_i = 1$ exactly at $i = \min\{j : \eta_j = 1\}$.*

Proof. If all $\eta_i = 0$, the formulas give zero outputs. Otherwise, α_i vanishes before the first success because $\eta_i = 0$, equals one at that success, and vanishes afterward because its prefix contains a zero factor; also $\beta = 1$. □

Define the exclusive-failure prefixes $Q_i = \prod_{j<i}(1 - \eta_j)$ for $0 \le i \le K$. A balanced parallel-prefix network computes all Q_i and the aggregate success $\beta = 1 - Q_K$ at depth $\lceil \log_2 K \rceil$; the circuit then computes each winner as $\alpha_i = \eta_i Q_i$ in one terminal multiplication level. Define the selected oil and vinegar vectors by $z_{\mathrm{sel}} = \sum_{i=0}^{K-1} \alpha_i \widehat{z}_i \in \mathbb{F}^o$ and $v_{\mathrm{sel}} = \sum_{i=0}^{K-1} \alpha_i v_i \in \mathbb{F}^v$; all candidates share the same salt. With the secret linear oil map O, the circuit assembles

$$\sigma_{\mathrm{sel}} = \mathsf{EncSig}\big((v_{\mathrm{sel}} + O z_{\mathrm{sel}}) \,\|\, z_{\mathrm{sel}}\,;\, \mathit{salt}\big), \tag{22}$$

and canonicalizes its discrete encoding. For a fixed dummy string $\sigma_{\mathrm{dum}} \in \{0,1\}^{s_\sigma}$, the total circuit output is

$$C_{vk,m}(x_\theta) = \big(\beta,\ \beta\sigma_{\mathrm{sel}} + (1-\beta)\sigma_{\mathrm{dum}}\big)\,, \tag{23}$$

where the selector is applied coordinatewise to the bit encoding. Thus an unsuccessful batch has the fixed output $\mathsf{Encode}(\perp) = (0, \sigma_{\mathrm{dum}})$.

Theorem 2 (Bounded-signer equivalence)**.** *Fix $\mathbb{F}_q$, UOV dimensions (v, o), a total attempt count $K \ge 1$, the derivation maps, a pivot-width sequence summing to o, and a public composition and candidate-batching schedule that evaluates all attempts and preserves their global order and arithmetic dependencies. For every bounded key pair sampled by $\mathsf{UOV.KeyGen}(1^\lambda)$, write $sk_{\mathrm{bd}} = (vk, \theta)$. For every $m \in \mathcal{M}_\lambda$, if salt derivation, target and vinegar derivations, field encodings and operations, predicates, inversions, and canonical encodings are evaluated exactly, then*

$$C_{vk,m}(\mathsf{enc}_\theta(\theta)) = \mathsf{Encode}(\mathsf{Sign}_{\mathrm{bd}}(sk_{\mathrm{bd}}, m)). \tag{24}$$

The circuit topology and gate schedule are independent of the first successful counter and of whether the bounded signer aborts.

Proof. The parser recovers θ, so the circuit and signer derive identical salts, targets, vinegars, and systems. Theorem 1 makes η_i exactly the signer's acceptance bit and supplies $A_i^{-1} b_i$ on success. Lemma 2 selects the least accepted counter, yielding the same signature by Equation (22). If none exists, Equation (23) yields $\mathsf{Encode}(\perp)$. Every candidate and selector gate is evaluated in either case, proving the topology claim. □

The threshold-FHE correctness property in Section 2.2 connects this algebraic identity to encrypted evaluation and decoding.

4 Threshold UOV from FHE

Complete homomorphic signing supplies the expensive computational step in one-round thresholdization: parties evaluate the signer on encrypted key material and release responses only for its final output. We apply the existing thresholdizer framework [BGG+18, PPS26, EY24] to the circuit of Section 3, using transcript-secure threshold FHE for semi-honest security and a strongly simulatable, robust universal thresholdizer for malicious security. The parties, access structure, field, dimensions, attempt count, and public pivot/batching/composition schedules remain parameters.

4.1 Semi-Honest Security

The dealer encrypts the signing state once; each party evaluates the public-message circuit and returns one decryption share for its final output, following the thresholdizer blueprint [BGG+18, PPS26]. Fix a uniform polynomial-time generator $(vk, m) \mapsto C_{vk,m}$ using a fixed circuit-realizable derivation/XOF suite. Theorem 2 gives

$$C_{vk,m}(x_\theta) = \mathsf{Encode}(\mathsf{Sign}_{\mathrm{bd}}(sk_{\mathrm{bd}}, m)). \tag{25}$$

The *semi-honest support condition* requires, for every supported UOV key and legal message, supported equal-length plaintexts $x_\theta, 0^{\ell_\theta}$, a circuit $C_{vk,m} \in \mathcal{C}_\lambda$ of public polynomially bounded size s_C and depth D_C, and output support $\{0,1\}^{1+s_\sigma}$. Zero-plaintext support is used only in the privacy hybrid.

Let Can be a fixed injective, efficiently decodable encoding with type/arity tags. Define $\mathsf{Filter}(vk, m, (b, s)) = s$ if the input is canonical, $b = 1$, and $\mathsf{Verify}(vk, m, s) = 1$, and $\perp$ otherwise. The dealer samples $(vk, (vk, \theta)) \leftarrow \mathsf{UOV.KeyGen}(1^\lambda)$ and $(pk, evk, \{dk_i\}) \leftarrow \mathsf{ThFHE.Setup}(1^\lambda, \mathcal{P}, \Gamma)$, encrypts $c_\theta \leftarrow \mathsf{Enc}_{pk}(x_\theta)$, publishes $pp = (\mathcal{P}, \Gamma, pk, evk, c_\theta, vk)$, and gives immutable state dk_i to party i. That party returns

$$\begin{aligned} c_m &= \mathsf{Eval}_{evk}(C_{vk,m}, c_\theta), \\ d_{i,m} &\leftarrow \mathsf{PartDec}(dk_i, c_m), \\ \mu_{i,m} &= \mathsf{Can}(i, C_{vk,m}, c_m, d_{i,m}). \end{aligned} \tag{26}$$

Partial verification checks the canonical party, request, and ciphertext tags; the semi-honest model supplies prescribed share generation. For distinct consistent responses from $I \in \Gamma$, combination returns

$$\mathsf{Filter}\big(vk, m, \mathsf{FinDec}(c_m, \{(i, d_{i,m})\}_{i \in I})\big). \tag{27}$$

Malformed or inconsistent response sets are rejected.

The bounded-signer theorem supplies classical base security in the public-preprocessing random-oracle model, shared by all algorithms and reductions; these oracles are not additional circuit gates. The following generic theorems also allow QPT adversaries when every premise holds in that model. This conditional QPT statement does not extend the classical CDM proof.

Theorem 3 (Semi-honest compiler)**.** *Fix $\mathfrak{C} \in \{\mathrm{PPT}, \mathrm{QPT}\}$ and the semi-honest support condition. Suppose $\mathsf{UOV}^{\mathrm{bd}}$ is correct and UF-CMA secure against $\mathfrak{C}$, and ThFHE_Γ is transcript secure (Definitions 11–13) against the corresponding adversaries, distinguishers, and schedulers, with uniform output simulator $\mathcal{S}_{\mathrm{out}} \in \mathfrak{C}$. Then Π_{sh} is a correct asynchronous one-round threshold signature satisfying fresh-message unforgeability under post-setup static semi-honest corruption (Definition 6). It preserves ordinary UOV verification keys and signatures. Appendix A.4 gives its correctness and security bounds and proof.*

4.2 Malicious Security

The malicious compiler replaces evaluation/decryption shares by verifiable universal-thresholdizer responses [BGG+18, EY24]. Its *active support condition* requires the same polynomial-length state encoding and uniform circuit generator, with supported plaintext x_θ, output space $\{0,1\}^{1+s_\sigma}$, and circuits of public polynomial size at most s_C and depth at most D for every supported key and legal message. The tagged map $m \mapsto C_{vk,m}$ must be injective.

The dealer samples the UOV key and $(pp_{\mathrm{ut}}, \{st_i\}) \leftarrow \mathsf{UT.Setup}(1^\lambda, 1^D, \mathcal{P}, \Gamma, x_\theta)$, publishes $(\mathcal{P}, \Gamma, D, pp_{\mathrm{ut}}, vk)$, and distributes st_i. Writing $C_m = C_{vk,m}$, party i returns $\mu_{i,m} = \mathsf{Can}(i, C_m, y_{i,m})$ for $y_{i,m} \leftarrow \mathsf{UT.Eval}(pp_{\mathrm{ut}}, st_i, C_m)$. Partial verification checks the tags and $\mathsf{UT.Verify}$; the extracted value is $\mathsf{val}_{\mathrm{ts}}(\mu_{i,m}) = \mathsf{Can}(i, C_m, \mathsf{val}_{\mathrm{ut}}(y_{i,m}))$. Combination applies Filter to $\mathsf{UT.Combine}$ on an authorized set of accepted responses.

Theorem 4 (Malicious compiler)**.** *Fix* $\mathfrak{C} \in \{\mathrm{PPT}, \mathrm{QPT}\}$ *and the active support condition. Suppose* $\mathsf{UOV}^{\mathrm{bd}}$ *is correct and UF-CMA secure against* $\mathfrak{C}$*, and* UT_Γ *is correct, strongly simulatable with uniform* $\mathcal{S} \in \mathfrak{C}$*, and response robust with deterministic canonical evaluation values, against the corresponding* $\mathfrak{C}$ *distinguishers and adversaries. Then* Π_{act} *is a correct asynchronous one-round threshold signature satisfying target-query unforgeability under post-setup static malicious corruption and partial-signature robustness (Definitions 7 and 10). It preserves ordinary UOV verification keys and signatures. Appendix A.4 gives its correctness and security bounds and proof.*

5 Experimental Results

We evaluate the complete signer and one-round semi-honest realization on RTX 6000 Ada and GB300, measuring warm responses, component improvements, communication, and initialization.

5.1 Experimental Design

Our implementation combines Grafting's discrete-CKKS organization with the Cheddar GPU backend [CCK+25, CKA25]. We use the UOV Ip′ dimensions $(q, v, o) = (256, 76, 44)$ [FI26] and $K = 17$ total attempts. For at most 200 fresh requests, the ideal retry term is below 2^{-128}; this is a retry bound, separate from computational security (Appendix A.2). Three independent 511-bit Toeplitz-CDM keys produce 192 trits, converted to a 128-bit salt, while independent 256-bit seeds supply vinegar derivation and complete SHAKE256 key expansion. The CDM dimensions follow the published classical target with the public-query qualification $q_1 + Q \leq 2^{40}$, including offline CDM-input-hash queries [DGH+21]. The instantiated security reduction is classical.

Packing and execution Signing uses RLWE ring degree 2^{15} (N15), followed by a public bridge to degree 2^{16} (N16) for threshold release. Eight coefficient slots represent each $\mathbb{F}_{256}$ element, giving 2,048 field positions per N15 ciphertext. A two-column panel for all candidates occupies $2Ko = 1,496$ positions; width three would require $3Ko = 2,244$. We therefore use 22 paired pivot stages with singleton composition and 334 functional cleanups. Coefficient refresh and bit restoration precede broadcasts in the selected-row update. Both platforms use the same signing and FHE parameters; GB300 additionally batches transforms, hoists rotations, and replays the solver as a CUDA graph.

Threshold release The evaluated configuration uses replicated additive sharing for two-out-of-three signing and standard coefficientwise noise flooding [BGG+18, PPS26]. We assume adaptive correctness of discrete-CKKS evaluation and bootstrapping, including terminal conversion and final decoding without modular wraparound, except with negligible probability. For transcript privacy, the standard flooding premise is a common coefficient-error lift of norm at most B across CRT limbs, except with negligible probability, and independent flooding whose aggregate shift distance over the supported lifetime is negligible. Both conditions hold uniformly over adaptive request prefixes and exposed corrupt-party tapes. We also assume input privacy for the complete evaluation-key bundle, pseudorandom service coins under state exposure, and collision-resistant request bindings. The service supports one no-fault cryptographic epoch of 100 fresh responses per party, including warm-ups and diagnostics.

Platforms and sampling We use GCC 13.3 Release builds with CUDA 13.0 on Ada and 13.2 on GB300. Each homogeneous platform configuration has three independent setups. In each setup, three party services run sequentially on one GPU, retain their imported keys, perform one counted but untimed full-call warm-up each, and answer the same three fresh 86-byte messages.

Table 1: Warm per-party average PartSign latency in seconds, pooled over 27 sequential local RPCs per platform: three setups, three parties per setup, and three shared fresh 86-byte messages per party. The last two rows partition each measured call into its nested N15 interval and the remaining work.

Component	RTX 6000 Ada	GB300
Nested N15	10.0	6.96
Remaining	0.715	0.813
Total	10.7	7.77

Table 2: RTX 6000 Ada component ablations. Baseline and ours give mean seconds over 18 evaluations per variant (three fixtures, six repetitions each). Speedup is the mean of the 18 matched baseline/ours ratios. Baselines are parallel scalar solving and residual-enabled selection, respectively.

Component	Baseline	Ours	Speedup
Linear solve	34.1	7.15	4.77×
Candidate selection	0.248	0.0740	3.36×

Thus each platform contributes 27 timed responses over nine messages and three independent keys. The interface supports up to 86 bytes, keeping each secret-dependent SHAKE256 input within one block.

The timer spans local request labeling through response return and final synchronization. The nested N15 component includes public preparation and output processing as well as encrypted signing. Setup, startup, key loading, and warm-up precede these measurements; coordinator validation, combination, and final signature verification follow them.

5.2 Signer Cost and Bottlenecks

Complete responses average 10.7 s on Ada and 7.77 s on GB300 (Table 1). Their nested signing stages average 10.0 s and 6.96 s, accounting for 93.3% and 89.5% of response time. Encrypted signing and its associated preparation dominate both execution paths.

Component ablations The solver variants evaluate the same 17 44×44 systems with matched FHE parameters and backend. Both perform 44 ordered row-selection scans. Pairing selects each second pivot before first-pivot normalization and propagation, reducing sequential inversion and update stages from 44 to 22. The 4.77× measured gain includes the complete designs' packing, refresh, and workspace schedules.

The selection variants share the same solver output and downstream first-success, cleanup, and extraction circuit. Replacing residual evaluation and its acceptance gate by the solver certificate saves 0.174 s (70.2%).

5.3 Threshold-Service Costs and Functionality

Each party returns 5.25 MB, so an authorized pair communicates 10.5 MB for a 136-B ordinary UOV signature. The public UOV key occupies 319 KB; reusable FHE evaluation state is additional setup material. Threshold responses dominate per-request communication.

Startup through readiness, including key loading, takes 81.4–81.7 s on Ada and 81.1–81.7 s on GB300; full-call warm-ups average 10.8 s and 37.7 s. GB300's separately measured N15 setup averages 316 s; complete joint dealer setup and deployment are outside these intervals.

Every timed and warm-up response validates, and every authorized pair yields a signature accepted by the reference UOV verifier. Candidate outputs agree with unrestricted finite-field elimination. Encrypted fixtures cover singular candidates and delayed success; all-candidate abort is checked at selection level. Repeated-message tests check deterministic salt and signature equality.

The resulting organization exchanges local FHE computation and multi-megabyte responses for one-round signing without intermediate openings or per-signature interactive preprocessing. THED retains an intermediate decryption for challenge hashing, while Vinaigrette uses one-use interactive preprocessing [PPS26, ABB+26]. Their reported timing boundaries differ from our full party RPCs. Our measurements establish the feasibility of the complete encrypted signer on both platforms and quantify the benefits of its paired solver and certified selection.

Acknowledgments

We thank Dan Boneh for helpful discussions on research directions related to this work.

AI Usage

AI-based tools were used to assist with portions of the writing and implementation in this work.

References

[ABB+26] Diego F. Aranha, Ward Beullens, Giacomo Borin, Fabio Campos, Sofía Celi, Basil Hess, Matthias J. Kannwischer, Lisa Kohl, and Guilhem Niot. Vinaigrette: An MPC approach for threshold UOV and MAYO. Preview writeup for the NIST First Call for Multi-Party Threshold Schemes, January 2026. Version 1.0, January 21, 2026.

[ADFQO19] Hartwig Anzt, Jack Dongarra, Goran Flegar, and Enrique S. Quintana-Ortí. Variable-size batched Gauss–Jordan elimination for block-Jacobi preconditioning on graphics processors. *Parallel Computing*, 81:131–146, 2019.

[AKP25] Andreea Alexandru, Andrey Kim, and Yuriy Polyakov. General functional bootstrapping using CKKS. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part III*, volume 16002 of *Lecture Notes in Computer Science*, pages 304–337. Springer, Cham, August 2025.

[ALYY23] Tae Min Ahn, Kang Hoon Lee, Joon Soo Yoo, and Ji Won Yoon. Cheap and fast iterative matrix inverse in encrypted domain. In Gene Tsudik, Mauro Conti, Kaitai Liang, and Georgios Smaragdakis, editors, *ESORICS 2023: 28th European Symposium on Research in Computer Security, Part I*, volume 14344 of *Lecture Notes in Computer Science*, pages 334–352. Springer, Cham, September 2023.

[ASY22] Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Round-optimal lattice-based threshold signatures, revisited. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *ICALP 2022: 49th International Colloquium on Automata, Languages and Programming*, volume 229 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:20. Schloss Dagstuhl, July 2022.

[BCD+24] Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. UOV — Unbalanced Oil and Vinegar. Technical report, National Institute of Standards and Technology, 2024. available at `https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures`.

[BCH+23] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(3):321–365, 2023.

[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, *Advances in Cryptology – CRYPTO 2022, Part IV*, volume 13510 of *Lecture Notes in Computer Science*, pages 517–550. Springer, Cham, August 2022.

[BCKS24] Youngjin Bae, Jung Hee Cheon, Jaehyung Kim, and Damien Stehlé. Bootstrapping bits with CKKS. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 94–123. Springer, Cham, May 2024.

[BdV18] Niek J. Bouman and Niels de Vreede. New protocols for secure linear algebra: Pivoting-free elimination and fast block-recursive matrix decomposition. Cryptology ePrint Archive, Report 2018/703, 2018.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147–150, 1984.

[BGG+18] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 565–596. Springer, Cham, August 2018.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018: 16th Theory of Cryptography Conference, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729. Springer, Cham, November 2018.

[BKSS24] Youngjin Bae, Jaehyung Kim, Damien Stehlé, and Elias Suvanto. Bootstrapping small integers with CKKS. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part I*, volume 15484 of *Lecture Notes in Computer Science*, pages 330–360. Springer, Singapore, December 2024.

[BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science*, pages 112–117. IEEE Computer Society Press, November 1982.

[BV87] Christian Bischof and Charles Van Loan. The WY representation for products of Householder matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(1):s2–s13, 1987.

[CCJK26] Jung Hee Cheon, Kevin Choi, Hyeoncheol Joo, and Taeseong Kim. One-round threshold XMSS and SPHINCS+ from fully homomorphic encryption. Cryptology ePrint Archive, Paper 2026/2360, 2026.

[CCK+25] Jung Hee Cheon, Hyeongmin Choe, Minsik Kang, Jaehyung Kim, Seonghak Kim, Johannes Mono, and Taeyeong Noh. Grafting: Decoupled scale factors and modulus in RNS-CKKS. In Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh,

David Lie, and Véronique Cortier, editors, *ACM CCS 2025: 32nd Conference on Computer and Communications Security*, pages 1098–1112. ACM Press, October 2025.

[CEN25] Sofía Celi, Daniel Escudero, and Guilhem Niot. Share the MAYO: Thresholdizing MAYO. In Ruben Niederhagen and Markku-Juhani O. Saarinen, editors, *Post-Quantum Cryptography - 16th International Workshop, PQCrypto 2025, Part I*, pages 165–198. Springer, Cham, April 2025.

[CFGM24] Benoît Cogliati, Pierre-Alain Fouque, Louis Goubin, and Brice Minaud. New security proofs and techniques for hash-and-sign with retry signature schemes. Cryptology ePrint Archive, Report 2024/609, 2024.

[CHK+18] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 360–384. Springer, Cham, April / May 2018.

[CKA25] Wonseok Choi, Jongmin Kim, and Jung Ho Ahn. Cheddar: A swift fully homomorphic encryption library designed for gpu architectures. In *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ASPLOS '26, page 35–49, New York, NY, USA, 2025. Association for Computing Machinery.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, Cham, December 2017.

[CS19] Daniele Cozzo and Nigel P. Smart. Sharing the LUOV: Threshold post-quantum signatures. In Martin Albrecht, editor, *17th IMA International Conference on Cryptography and Coding*, volume 11929 of *Lecture Notes in Computer Science*, pages 128–153. Springer, Cham, December 2019.

[DGH+21] Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. MPC-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 517–547, Virtual Event, August 2021. Springer, Cham.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, Feb. 2018.

[DKM+24] Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part II*, volume 14652 of *Lecture Notes in Computer Science*, pages 219–248. Springer, Cham, May 2024.

[DMPS24] Nir Drucker, Guy Moshkowich, Tomer Pelleg, and Hayim Shaul. BLEACH: Cleaning errors in discrete computations over CKKS. *Journal of Cryptology*, 37(1):3, January 2024.

[EY24] Ehsan Ebrahimi and Anshu Yadav. Strongly secure universal thresholdizer. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part III*, volume 15486 of *Lecture Notes in Computer Science*, pages 207–239. Springer, Singapore, December 2024.

[FI26] Hiroki Furue and Yasuhiko Ikematsu. Key recovery attacks on UOV using p^ℓ-truncated polynomial rings. Cryptology ePrint Archive, Paper 2026/298, 2026.

[FWX+23] Shengyu Fan, Zhiwei Wang, Weizhi Xu, Rui Hou, Dan Meng, and Mingzhe Zhang. Tensorfhe: Achieving practical computation on encrypted data using gpgpu. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 922–934, 2023.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 464–479. IEEE Computer Society Press, October 1984.

[GKS24] Kamil Doruk Gür, Jonathan Katz, and Tjerand Silde. Two-round threshold lattice-based signatures from threshold homomorphic encryption. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Part II*, pages 266–300. Springer, Cham, June 2024.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, April 1988.

[Hal17] Shai Halevi. Homomorphic encryption. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, Information Security and Cryptography, pages 219–276. Springer, Cham, 2017.

[HBD+22] Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Jean-Philippe Aumasson, Bas Westerbaan, and Ward Beullens. SPHINCS+. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022`.

[HLR+26] Kai Hu, Gregor Leander, Håvard Raddum, Arne Sandrib, and Aleksei Udovenko. Cryptanalysis of two alternating moduli weak PRFs. *IACR Transactions on Symmetric Cryptology*, 2026(1):95–118, 2026.

[HYY20] Mi Yeon Hong, Joon Soo Yoo, and Ji Won Yoon. Homomorphic model selection for data analysis in an encrypted domain. *Applied Sciences*, 10(18):6174, 2020.

[JKA+21] Wonkyung Jung, Sangpyo Kim, Jung Ho Ahn, Jung Hee Cheon, and Younho Lee. Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(4):114–148, 2021.

[KCL25] Seunghu Kim, Seongbong Choi, and Hyung Tae Lee. Improved matrix inversion with packed ciphertexts using fully homomorphic encryption. Cryptology ePrint Archive, Report 2025/1274, 2025.

[Klu22] Kamil Kluczniak. NTRU-v-um: Secure fully homomorphic encryption from NTRU with small modulus. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022: 29th Conference on Computer and Communications Security*, pages 1783–1797. ACM Press, November 2022.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222. Springer, Berlin, Heidelberg, May 1999.

[KSS24] Jaehyung Kim, Jinyeong Seo, and Yongsoo Song. Simpler and faster BFV bootstrapping for arbitrary plaintext modulus from CKKS. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 2535–2546. ACM Press, October 2024.

[MHWW24] Shihe Ma, Tairong Huang, Anyu Wang, and Xiaoyun Wang. Faster BGV bootstrapping for power-of-two cyclotomics through homomorphic NTT. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part I*, volume 15484 of *Lecture Notes in Computer Science*, pages 143–175. Springer, Singapore, December 2024.

[NW06] Kobbi Nissim and Enav Weinreb. Communication efficient secure linear algebra. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 522–541. Springer, Berlin, Heidelberg, March 2006.

[PP25] Chris Peikert and Zachary Pepin. Vive galois! Part 1: Optimal SIMD packing and packed bootstrapping for FHE. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025: 23rd Theory of Cryptography Conference, Part I*, volume 16268 of *Lecture Notes in Computer Science*, pages 185–219. Springer, Cham, December 2025.

[PPS26] Jai Hyun Park, Alain Passelègue, and Damien Stehlé. THED: Threshold dilithium from FHE. Cryptology ePrint Archive, Paper 2026/638, 2026.

[PS24] Alain Passelègue and Damien Stehlé. Low communication threshold fully homomorphic encryption. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology – ASIACRYPT 2024, Part I*, volume 15484 of *Lecture Notes in Computer Science*, pages 297–329. Springer, Singapore, December 2024.

[SS26] Tim Seuré and Elias Suvanto. Finite-field arithmetic in CKKS. Cryptology ePrint Archive, Paper 2026/1102, 2026.

A Security and Algebraic Details

We give the security games, signer reduction, compiler proofs, and public composition schedule.

A.1 Signature-Security Games

Use the syntax and adversary class $\mathfrak{C}$ of Section 2. Query and runtime bounds are polynomial. Malformed oracle requests return $\perp$ without state change; invalid corruption choices lose. Honest states persist across invocations. Security means negligible winning probability for every adversary in $\mathfrak{C}$ and supported instance.

Definition 5 (UF-CMA)**.** *After $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, the adversary receives vk and a signing oracle. Initialize $\mathcal{Q} = \varnothing$; the oracle records each legal message in $\mathcal{Q}$ before signing, including aborts. The adversary wins with $(m^\star, \sigma^\star)$ exactly when $m^\star \in \mathcal{M}_\lambda \setminus \mathcal{Q}$ and $\mathsf{Verify}(vk, m^\star, \sigma^\star) = 1$. Its winning probability is $\mathrm{Adv}^{\text{uf-cma}}_{\Sigma,\mathcal{A}}$ [GMR88].*

Definition 6 (Semi-honest fresh-message threshold unforgeability)**.** *Run honest setup and independently sample prescribed polynomial-length future random tapes. After seeing (pp, vk), the adversary chooses any $S \in \mathcal{U}_\Gamma$ and receives its complete states and remaining tapes. Corrupt invocations follow the prescribed algorithm and coins. An oracle answers adaptive (i, m) requests for $i \notin S$ with $\mathsf{PartSign}(pp, sk_i, m)$, advancing the honest state and tape and recording m in $\mathcal{Q}$ before execution. The winning condition is that of Definition 5; denote its probability by $\mathrm{Adv}^{\text{sh-thuf}}_{\Pi,\mathcal{A}}$. This adapts TS-UF-0 message freshness to general access structures and post-setup static honest-but-curious corruption [BCK+22, BGG+18].*

Definition 7 (Target-query threshold unforgeability)**.** *After honest setup, the adversary chooses $S \in \mathcal{U}_\Gamma$ from (pp, vk) and receives its states. Initialize $T_m = \varnothing$. Each legal honest-party query (i, m) adds i to T_m before invoking* PartSign *and updating its state, including on abort. A forgery wins iff*

$$m^\star \in \mathcal{M}_\lambda, \quad S \cup T_{m^\star} \notin \Gamma, \quad \mathsf{Verify}(vk, m^\star, \sigma^\star) = 1. \tag{28}$$

Denote its probability by $\mathrm{Adv}^{\text{tq-thuf}}_{\Pi,\mathcal{A}}$. This is the access-structure adaptation of TS-UF-1 [BCK+22, EY24].

A.2 Bounded-Signer Security

We use classical public-preprocessing random oracles, with concrete target hashing, verification, and secret-dependent circuits. Independent keys and parameter-specific primitive assumptions underlie the bounds.

Key and coin laws The complete generator $\mathsf{KExp}(\rho_{\text{key}})$ supplies $O \in \mathbb{F}^{v\times o}$, upper-triangular $P^{(1)}_a \in \mathbb{F}^{v\times v}$, and $P^{(2)}_a \in \mathbb{F}^{v\times o}$ for $1 \le a \le o$. Set $(D_j)_{a,b} = (P^{(2)}_a + (P^{(1)}_a + (P^{(1)}_a)^\top)O)_{j,b}$ and choose the oil–oil block so that the public quadratic map vanishes on $\{(Oz, z) : z \in \mathbb{F}^o\}$. This gives Equation (2) in every characteristic. The ideal expansion law samples all free coefficients independently and uniformly; binary-extension fields have a bijective bit encoding, while other fields require a suitable sampler.

Definition 8 (PRG and PRF advantages)**.** *Write $\mathrm{Adv}^{\text{prg}}_{\mathsf{KExp},\mathcal{D}}$ for the absolute difference between $\mathcal{D}$'s output-one probabilities on the complete expansion of a uniform seed and on the ideal coefficient law [BM82]. For a keyed family $\mathsf{F} : \mathcal{X} \to \mathcal{Y}$, $\mathrm{Adv}^{\text{prf}}_{\mathsf{F},\mathcal{D}}(\lambda; Q)$ compares oracle access to a uniformly keyed F and one uniform function, consistently answering repeats, with at most Q classical queries [GGM84]. Security requires negligible advantage for every classical PPT distinguisher.*

CDM and public preprocessing The published Toeplitz CDM family is $\mathsf{W}_{M,k}(x) = M\iota(A_k x)$, where $k \in \mathbb{F}_2^{2d-1}$, $(A_k)_{i,j} = k_{d-1+i-j}$, and ι embeds bits into $\mathbb{F}_3$. The fixed public $M \in \mathbb{F}_3^{h\times d}$ is uniform and independent of keys [BIP+18, DGH+21]. Public independent random oracles $H_0 : \mathcal{M}_\lambda \to \{0,1\}^\kappa$ and $H_1 : \{0,1\}^\kappa \to \mathbb{F}_2^d$ give $p_m = H_0(m)$ and $x_m = H_1(p_m)$. A disjoint

public XOF stream at a fixed suite input generates M by rejecting byte 255 and reducing other bytes modulo three. The concrete namespaces use domain-separated SHAKE256. Bank a evaluates $\mathsf{G}_{M,k_a}(m) = \mathsf{W}_{M,k_a}(x_m)$; the independent concrete vinegar PRF $\mathsf{V}_{k_\mathrm{v}}(p, s, i)$ takes the full digest, salt, and counter. Neither vinegar derivation nor KExp queries these public namespaces.

Definition 9 (Random-input weak PRF)**.** *Sample uniform M, k once and give M to a classical distinguisher. Each sample request returns an independent uniform $x \in \mathbb{F}_2^d$ paired with either $\mathsf{W}_{M,k}(x)$ or $R(x)$ for one uniform function $R : \mathbb{F}_2^d \to \mathbb{F}_3^h$. Repeated inputs have consistent outputs. The absolute output-probability difference after at most N samples is $\mathrm{Adv}^{\mathrm{wprf}}_{\mathsf{W},\mathcal{D}}(\lambda; N)$; it must be negligible for all polynomial N and classical PPT $\mathcal{D}$. This is the random-example game with explicit public parameters [BIP+ 18, Definition 2.1].*

Lemma 3 (Hash-then-evaluate)**.** *For b independent keys sharing M, replacing the message-indexed banks by independent uniform functions costs at most $\sum_{a<b} \mathrm{Adv}^{\mathrm{wprf}}_{\mathsf{W},\mathcal{D}_a}(\lambda; N_1) + \varepsilon_{\mathrm{pre}}$, where $N_j = q_j + Q$, Q bounds family queries, q_j bounds distinct public H_j queries, and*

$$\varepsilon_{\mathrm{pre}} = \binom{N_0}{2} 2^{-\kappa} + \binom{N_1}{2} 2^{-d}. \tag{29}$$

Each reduction uses at most N_1 samples and polynomial overhead.

Proof. Replace banks one at a time. Take M from the weak-PRF challenge and program every new $H_1(p)$ with the x from a fresh challenge pair (x, y), retaining y for that bank. Independently sampled keys and lazy tables answer the other banks, including repeats. To expose the matrix-generation stream conditional on M, emit 255 with probability $1/256$; otherwise choose uniformly among the 85 byte values with the next prescribed trit's residue. After all matrix entries, use uniform bytes. This is exactly the conditional stream law. Until an H_0 or H_1 collision, distinct messages index distinct random outputs. A union bound gives Equation (29); the same hybrids hold when stopped at the first collision. □

Random-coins reference For a public polynomial $R \geq K$ whose counters are supported, $\Sigma^R_{\mathcal{S}}$ uses ideal block-key coefficients, a fresh uniform ℓ-bit salt, and R independent uniform vinegars per signing call, returning the first nonsingular solution or $\perp$. The reference Σ^R_{UOV} instead samples uniform $T \in \mathrm{GL}_{v+o}(\mathbb{F})$ and an independent central homogeneous quadratic map $\mathcal{F}$ with uniform vinegar–vinegar and vinegar–oil coefficients and zero oil–oil coefficients, publishes $\mathcal{F} \circ T$, and transforms the first full-rank solution by T^{-1}. This is random-coins UOV with full-rank rejection, the base UF-CMA premise [KPG99, BCD+24, CFGM24].

Lemma 4 (Coordinate-chart reduction)**.** *For*

$$\alpha_{q,v,o} = \frac{\prod_{j=1}^{o}(1 - q^{-j})}{\prod_{j=v+1}^{v+o}(1 - q^{-j})} > \frac{1}{4},$$

every block-reference forger induces a reference-UOV forger with the same query bound and polynomial overhead, and $\mathrm{Adv}^{\text{uf-cma}}_{\Sigma^R_{\mathcal{S}},\mathcal{A}} \leq \alpha^{-1}_{q,v,o}\, \mathrm{Adv}^{\text{uf-cma}}_{\Sigma^R_{\mathrm{UOV}},\mathcal{B}}$.

Proof. The uniform oil space is transverse to $\mathbb{F}^v \times \{0\}^o$ with probability $\alpha_{q,v,o}$: exactly q^{vo} oil spaces are graphs of matrices O. Pullback by T bijects the corresponding homogeneous quadratic coefficient spaces. Conditional on transversality, O and the free coefficients $P^{(1)}, P^{(2)}$ therefore have the ideal block-key law. Uniform vinegars sample uniform cosets of the oil space independently of its basis and complement. On each coset, restriction is affine linear; full-rank acceptance and its unique preimage are therefore intrinsic. The entire R-trial transcript has the block-reference law under this conditioning. Running the forger unchanged gives the factor α^{-1}. Finally $\alpha > \prod_{j=1}^{o}(1 - 2^{-j}) > 1/4$, by the product lower bound for the factors after the first. □

Lemma 5 (Radix conversion)**.** *For L independent uniform trits, let $X = \sum_{j<L} t_j 3^j$, $D = 3^L$, $W = 2^\ell$, and $r = D \bmod W$. The total variation distance between $\mathsf{LE}_\ell(X \bmod W)$ and a uniform ℓ-bit string is*

$$\delta_{L,\ell} = \frac{r(W-r)}{DW} \le \frac{W}{4D}. \tag{30}$$

Proof. Modulo W, r residues have $\lfloor D/W \rfloor + 1$ preimages and the others have $\lfloor D/W \rfloor$. Summing their probability discrepancies gives the equality; $r(W-r) \le W^2/4$ gives the bound. □

Lemma 6 (Ideal retry bound)**.** *Let $D_j \in \mathbb{F}_q^{o\times o}$ and $v_i \in \mathbb{F}_q^v$ be mutually independent uniform matrices and vectors, with $A_i = \sum_j (v_i)_j D_j$. Put $p = 1 - \prod_{j=1}^{o}(1-q^{-j})$ and*

$$\begin{aligned} B_t &= p^t + 1 - \prod_{j=0}^{t-1}(1-q^{j-v}), \\ \beta_K &= \min\{1, \min_{1\le t\le \min(K,v)} B_t\}. \end{aligned} \tag{31}$$

Then all K matrices are singular with probability at most β_K, and $B_t < (q-1)^{-t} + q^{t-v}/(q-1)$.

Proof. The first t vinegar vectors have full row rank except with probability $1 - \prod_{j<t}(1-q^{j-v})$. Conditional on full rank, independent matrix coordinates map uniformly onto $\mathbb{F}_q^t$, making those t systems independent uniform matrices. Their joint singularity probability is p^t. All-attempt failure implies prefix failure. Finally $p < 1/(q-1)$ and the rank-deficiency probability is below $q^{t-v}/(q-1)$. □

Adaptive retry accounting With uniform signing coins, let $\varepsilon^{\mathrm{ret}}_{K,Q,\mathcal{D}}$ be the chance that one of the first Q distinct adaptive messages exhausts its K attempts, caching repeats including aborts. Replacing complete key expansion by uniform coefficients costs one PRG advantage. Conditional on $O, P^{(1)}$, the map $P^{(2)} \mapsto P^{(2)} + (P^{(1)} + (P^{(1)})^\top)O$ is a translation, giving independent uniform D_j. For fixed D_j, let p_{sing} be the fraction of singular vinegars. The contribution of a potential fresh request is $\mathbb{E}[I\, p_{\mathrm{sing}}^K] \le \mathbb{E}[p_{\mathrm{sing}}^K] \le \beta_K$, where I indicates that the request occurs. Thus

$$\varepsilon^{\mathrm{ret}}_{K,Q,\mathcal{D}} \le \mathrm{Adv}^{\mathrm{prg}}_{\mathsf{KExp},\mathcal{D}_{\mathrm{kg}}} + Q\beta_K. \tag{32}$$

For fixed q, the ideal term is negligible when some admissible t and $v-t$ both grow faster than $\log\lambda$, since p is bounded below one. The prefix bound also applies when $K > v$.

Theorem 5 (Classical UF-CMA of the bounded deterministic signer)**.** *Work in the public-preprocessing random-oracle model above with a chosen finite field and dimension family, positive polynomially bounded total attempt counts $K \le R$, and the specified key and coin maps. For every classical PPT $\mathcal{A}$ making at most Q legal signing queries and q_0, q_1 public queries, let $N_j = q_j + Q$ and let $\varepsilon_{\mathrm{pre}}$ be as in Equation (29). There are classical PPT reductions $\mathcal{B}$, $\{\mathcal{D}_{\mathrm{s},a}\}_{a=0}^{b-1}$, $\mathcal{D}_{\mathrm{v}}$, and $\mathcal{D}^{\mathrm{uf}}_{\mathrm{kg}}$ such that*

$$\begin{aligned} \mathrm{Adv}^{\text{uf-cma}}_{\mathsf{UOV}^{\mathrm{bd}},\mathcal{A}}(\lambda) \le{}& \sum_{a=0}^{b-1} \mathrm{Adv}^{\mathrm{wprf}}_{\mathsf{W},\mathcal{D}_{\mathrm{s},a}}(\lambda; N_1) + \varepsilon_{\mathrm{pre}} \\ &+ Q\delta_{L,\ell} \\ &+ \mathrm{Adv}^{\mathrm{prf}}_{\mathsf{V},\mathcal{D}_{\mathrm{v}}}(\lambda; KQ) \\ &+ \mathrm{Adv}^{\mathrm{prg}}_{\mathsf{KExp},\mathcal{D}^{\mathrm{uf}}_{\mathrm{kg}}}(\lambda) + Q\beta_K \\ &+ \alpha^{-1}_{q,v,o}\, \mathrm{Adv}^{\text{uf-cma}}_{\Sigma^R_{\mathrm{UOV}},\mathcal{B}}(\lambda). \end{aligned} \tag{33}$$

$\mathcal{B}$ makes at most Q signing queries, each $\mathcal{D}_{\mathrm{s},a}$ uses at most N_1 random-input samples, and $\mathcal{D}_{\mathrm{v}}$ makes at most KQ queries. The reductions have polynomial overhead. Consequently, classical UF-CMA of Σ^R_{UOV}, weak-PRF security of W, PRF security of V, PRG security of KExp, and negligible preprocessing, conversion, and retry terms imply classical bounded-signer UF-CMA. The public-query bounds are separate from the signing-service lifetime.

Proof. Record every queried message before execution and cache repeats, including aborts. Stop at the first public-preprocessing collision, charging $\varepsilon_{\mathrm{pre}}$ once. Lemma 3's stopped hybrids replace the independent salt banks; Lemma 5 couples their converted outputs to uniform salts at cost $Q\delta_{L,\ell}$. Next replace the independent vinegar PRF by a lazy random function on (p, s, i), using at most KQ queries. Before a preprocessing collision these inputs are distinct for fresh messages and counters. Replace complete key expansion by uniform coefficients, charging its PRG advantage once.

Extend each fresh message's vinegar sequence from K to R while coupling its salt and first K trials. Responses differ only on a K-trial abort, costing at most $Q\beta_K$ by the uniform-key argument above. A block-reference signing oracle simulates this final game with one query per fresh message and cached repeats; a forgery therefore remains on an unqueried message. Lemma 4 supplies the reference-UOV factor. All public preprocessing is simulated locally, the target hash and verification relation are unchanged, and summing the hybrid gaps gives the claimed bound. □

Corollary 6 (Bounded-signer correctness)**.** *Fix positive polynomially bounded K and $m \in \mathcal{M}_\lambda$. There are distinguishers $\mathcal{D}^{\mathrm{corr}}_{\mathrm{v},m}$ and $\mathcal{D}^{\mathrm{corr}}_{\mathrm{kg},m}$ in $\mathfrak{C}$, the first making K oracle queries, such that*

$$\begin{aligned}\varepsilon^{\mathrm{corr}}_{\mathsf{UOV}^{\mathrm{bd}},m}(\lambda) \le{}& \mathrm{Adv}^{\mathrm{prf}}_{\mathsf{V},\mathcal{D}^{\mathrm{corr}}_{\mathrm{v},m}}(\lambda; K) \\ &+ \mathrm{Adv}^{\mathrm{prg}}_{\mathsf{KExp},\mathcal{D}^{\mathrm{corr}}_{\mathrm{kg},m}}(\lambda) + \beta_K.\end{aligned} \tag{34}$$

Correctness follows when these terms are negligible uniformly over messages.

Proof. For fixed m, sample its salt and replace the independent vinegar family's K distinct counter-tagged outputs by uniform vectors, then replace key expansion. Abort depends only on the oil matrices, so Lemma 6 bounds it by β_K. There is no prehash-collision loss for one message. Every nonabort output verifies by Equation (2). □

Evaluated statistical terms For $(q, v, o, K) = (256, 76, 44, 17)$ and $Q \le 200$,

$$\begin{aligned} Q\beta_K &< 200(255^{-17} + 256^{-59}/255) < 2^{-128}, \\ \delta_{192,128} &\le 2^{128}/(4 \cdot 3^{192}) < 2^{-178}, \\ Q\delta_{192,128} &< 2^{-170}, \qquad \alpha^{-1}_{256,76,44} < 255/254. \end{aligned}$$

Sixteen attempts do not meet the same retry bound, motivating 17. The CDM choice uses $d = 256$ and projects the published 81-trit output to 64 trits per independent key, without increasing weak-PRF advantage [DGH+21]. Its concrete sample qualification counts $q_1 + Q \le 2^{40}$. With $\kappa = 512$, public collisions contribute $\binom{q_0+Q}{2}2^{-512} + \binom{q_1+Q}{2}2^{-256}$ separately from the computational advantages.

A.3 Threshold-Primitive Security

These post-setup-static, exposed-state games adapt [BGG+18, ASY22, PS24, EY24]. Invalid inputs terminate with zero; malformed requests return $\perp$ without state change. Bounds are polynomial; negligibility ranges over supported instances and adversaries in $\mathfrak{C}$.

Definition 10 (Partial-signature robustness)**.** *Let $\nu_{i,m} = \mathsf{val}_{\mathrm{ts}}(\mathsf{PartSign}(pp, sk_i, m))$ be canonical and independent of invocation coins and prior requests. Given honest public parameters and every party state, an adversary wins by producing an accepted $(i, m, \widetilde{\mu})$ with $i \in \mathcal{P}$, $m \in \mathcal{M}_\lambda$, and $\mathsf{val}_{\mathrm{ts}}(\widetilde{\mu}) \neq \nu_{i,m}$. The winning probability $\mathrm{Adv}^{\text{part-rob}}_{\Pi,\mathcal{A}}$ must be negligible. Verification checks complete responses; combination depends on parsed values, allowing different valid proof certificates for the same value.*

Definition 11 (Adaptive threshold-FHE correctness)**.** *A scheduler chooses x before honest setup and encryption. Independently sample all prescribed future tapes before coalition selection. After seeing (pk, evk, c_x) it selects $S \in \mathcal{U}_\Gamma$ and receives its decryption states and prescribed future tapes. It makes at most Q adaptive queries $(C, I) \in \mathcal{C}_\lambda \times \Gamma$. Each returns the canonical evaluated ciphertext, the parties' shares using the next prescribed coins, and their combined value; synchronized tape copies advance also for corrupt parties. Let $\varepsilon^{\mathrm{adec}}_{\mathsf{ThFHE},\mathcal{D}}$ be the probability that any returned value differs from $C(x)$. It must be negligible for every scheduler and polynomial Q.*

Definition 12 (Exposed-state threshold IND-CPA)**.** *After public setup, $\mathcal{B}$ chooses equal-length supported x_0, x_1 and arbitrary auxiliary data z. It receives $c^\star \leftarrow \mathsf{Enc}_{pk}(x_b)$, selects any $S \in \mathcal{U}_\Gamma$ using the complete public view, then receives $\{dk_i\}_{i\in S}$ and outputs a bit. The absolute difference between output-one probabilities for $b = 0, 1$, denoted $\mathrm{Adv}^{\text{tind-cpa}}_{\mathsf{ThFHE},\mathcal{B}}$, must be negligible.*

Definition 13 (Adaptive output simulation)**.** *A distinguisher chooses (x, z) before honest setup and encryption, selects $S \in \mathcal{U}_\Gamma$ after seeing (pk, evk, c_x, z), and receives its exposed decryption states. It adaptively requests (C, I) with $C \in \mathcal{C}_\lambda$ and $\varnothing \neq I \subseteq \mathcal{P} \setminus S$, counting individual responses toward Q. The real world returns canonical c_C and honest partial decryptions. In the ideal world one uniform stateful simulator is initialized with $(pk, evk, c_x, S, \{dk_i\}_{i\in S})$ and answers each request from $(C, I, c_C, C(x))$ and its state. The simulator receives neither x nor z directly. The real/ideal distinguishing advantage $\mathrm{Adv}^{\text{out-sim}}_{\mathsf{ThFHE},\mathcal{D},\mathcal{S}_{\mathrm{out}}}$ must be negligible for every distinguisher and polynomial Q, including repeated circuits and overlapping sets.*

Thresholdizer correctness Write $\mathcal{C}^{\le D}_\lambda$ for supported circuits of depth at most D. For honest setup on x, evaluation of C, and authorized I, let $\varepsilon^{\mathrm{eval}}_{\mathsf{UT}}$ bound the probability of combination differing from $C(x)$ and $\varepsilon^{\mathrm{ver}}_{\mathsf{UT}}$ that of any honest response in I failing verification. Both must be negligible uniformly over supported inputs, circuits, sets, and depth bounds.

Definition 14 (Strong thresholdizer simulation)**.** *A distinguisher chooses (x, z) before setup, sees either real setup on x or simulated setup $\mathcal{S}_0(1^\lambda, 1^D, \mathcal{P}, \Gamma)$ independent of (x, z), then chooses $S \in \mathcal{U}_\Gamma$ and receives its real or simulated states. It adaptively queries (J, C) for nonempty honest sets $J \subseteq \mathcal{P} \setminus S$ and $C \in \mathcal{C}^{\le D}_\lambda$. The real world returns complete evaluation responses. The simulator maintains shared state and cumulative queried sets T_C, initially empty, updating $T_C \leftarrow T_C \cup J$ before answering. Its algorithm $\mathcal{S}_I$ answers while $S \cup T_C \notin \Gamma$ without x, z, or $C(x)$; once authorized, $\mathcal{S}_V$ additionally receives only $C(x)$. One uniform simulator must make the real/ideal advantage $\mathrm{Adv}^{\text{sut-sim}}_{\mathsf{UT},\mathcal{B},\mathcal{S}}$ negligible, including the first authorization transition and correlations across circuits.*

Definition 15 (Thresholdizer response robustness)**.** *For deterministic canonical honest values $\nu_{i,C} = \mathsf{val}_{\mathrm{ut}}(\mathsf{UT.Eval}(pp, st_i, C))$, an adversary chooses a supported x and receives honest public parameters and all evaluation states. It wins with an accepted $(i, C, \widetilde{y}_i)$ with $i \in \mathcal{P}$ and $C \in \mathcal{C}^{\le D}_\lambda$ whose parsed value differs from $\nu_{i,C}$. The winning probability $\mathrm{Adv}^{\text{sut-rob}}_{\mathsf{UT},\mathcal{B}}$ must be negligible.*

A.4 Compiler Bounds and Proofs

All bounds use the cited theorem's hypotheses and fixed game parameters.

Semi-honest compiler Under Theorem 3, for every $I \in \Gamma, m \in \mathcal{M}_\lambda$,

$$\varepsilon^{\mathrm{corr}}_{\Pi_{\mathrm{sh}},I,m} \le \varepsilon^{\mathrm{corr}}_{\mathsf{UOV}^{\mathrm{bd}},m} + \varepsilon^{\mathrm{dec}}_{\mathsf{ThFHE}}. \tag{35}$$

For every polynomial Q and $\mathcal{A} \in \mathfrak{C}$ making at most Q legal partial-signing queries, straight-line reductions in $\mathfrak{C}$ satisfy

$$\begin{aligned} \mathrm{Adv}^{\text{sh-thuf}}_{\Pi_{\mathrm{sh}},\mathcal{A}} &\le \mathrm{Adv}^{\text{uf-cma}}_{\mathsf{UOV}^{\mathrm{bd}},\mathcal{B}_{\mathrm{sig}}} \\ &\quad + \mathrm{Adv}^{\text{tind-cpa}}_{\mathsf{ThFHE}_\Gamma,\mathcal{B}_{\mathrm{fhe}}} \\ &\quad + \mathrm{Adv}^{\text{out-sim}}_{\mathsf{ThFHE}_\Gamma,\mathcal{B}_{\mathrm{out}},\mathcal{S}_{\mathrm{out}}}. \end{aligned} \tag{36}$$

The reductions respectively make at most Q signing queries (one per distinct honest-request message), one IND-CPA challenge, and Q singleton simulation queries/responses, and run in time $T_{\mathcal{A}} + \mathrm{poly}(\lambda, n, Q, s_C)$.

Proof of Theorem 3. The circuit identity and filter give the correctness bound and ordinary formats. Set-independent local response generation gives one-round asynchrony.

For unforgeability, replace all honest shares jointly by $\mathcal{S}_{\mathrm{out}}$ using exact signer outputs and correlated auxiliary value vk, at the output-simulation cost. Then replace c_θ by an encryption of 0^{ℓ_θ} at the exposed-state IND-CPA cost. Both reductions preserve post-setup coalition selection and expose independently sampled prescribed corrupt-party tapes; the stateful simulator preserves correlations across repeated and overlapping queries.

In the final game, $\mathcal{B}_{\mathrm{sig}}$ generates threshold setup and the zero encryption, obtains each message's first honest-request output from its signing oracle, and caches it, including $\perp$. Its queried-message set equals the threshold game's $\mathcal{Q}$, recorded before execution. Thus every winning fresh-message forgery is an ordinary forgery. These two hybrid gaps yield the bound; decoding error enters only correctness. □

Malicious compiler Under Theorem 4, for every $I \in \Gamma, m \in \mathcal{M}_\lambda$,

$$\varepsilon^{\mathrm{corr}}_{\Pi_{\mathrm{act}},I,m} \le \varepsilon^{\mathrm{corr}}_{\mathsf{UOV}^{\mathrm{bd}},m} + \varepsilon^{\mathrm{eval}}_{\mathsf{UT}} + \varepsilon^{\mathrm{ver}}_{\mathsf{UT}}. \tag{37}$$

For every polynomial Q, target-query adversary $\mathcal{A}_{\mathrm{uf}} \in \mathfrak{C}$ with at most Q legal partial-signing queries, and robustness adversary $\mathcal{A}_{\mathrm{rob}} \in \mathfrak{C}$, straight-line reductions in $\mathfrak{C}$ satisfy

$$\begin{aligned} \mathrm{Adv}^{\text{tq-thuf}}_{\Pi_{\mathrm{act}},\mathcal{A}_{\mathrm{uf}}} &\le \mathrm{Adv}^{\text{uf-cma}}_{\mathsf{UOV}^{\mathrm{bd}},\mathcal{B}_{\mathrm{sig}}} \\ &\quad + \mathrm{Adv}^{\text{sut-sim}}_{\mathsf{UT}_\Gamma,\mathcal{B}_{\mathrm{sim}},\mathcal{S}}, \end{aligned} \tag{38}$$

$$\mathrm{Adv}^{\text{part-rob}}_{\Pi_{\mathrm{act}},\mathcal{A}_{\mathrm{rob}}} \le \mathrm{Adv}^{\text{sut-rob}}_{\mathsf{UT}_\Gamma,\mathcal{B}_{\mathrm{rob}}}. \tag{39}$$

The first two reductions make at most Q signing and simulation-response queries, respectively, and run in time $T_{\mathrm{uf}} + \mathrm{poly}(\lambda, n, Q, s_C)$; the robustness reduction makes no oracle queries and runs in time $T_{\mathrm{rob}} + \mathrm{poly}(\lambda, n, s_C)$.

Proof of Theorem 4. The circuit identity and filter give correctness and ordinary formats outside the three bounded failure events. Set-independent local evaluation gives one-round asynchrony.

Strong simulation replaces setup, exposed states, and all complete responses by one joint ideal transcript, with auxiliary value vk. For each message, $\mathcal{S}_I$ answers without an output while $S \cup T_m$ is unauthorized. At its first authorized crossing, $\mathcal{B}_{\mathrm{sig}}$ queries its ordinary signing oracle, caches the result including $\perp$, and gives its encoding to $\mathcal{S}_V$. Circuit injectivity equates the thresholdizer's queried-party set with T_m. A winning target remains unauthorized and was never queried to the signing oracle, proving the unforgeability bound.

For robustness, canonical parsing binds (i, m) to C_m. A different accepted outer value therefore yields a different accepted thresholdizer value for the same (i, C_m), with exactly the same winning probability. Randomizing a valid certificate does not change either value. □

A.5 Public Composition Schedules

Fix public positive block lengths $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{N_{\mathrm{blk}}})$ partitioning the N_{step} pivot steps, independently of their widths. Within block r, let $d_1, \ldots, d_{\nu_r}$ be the local pivot widths, write $G_j = I_o + T_j S_j^{\mathsf{T}}$, and let $r_j = \sum_{u \le j} d_u$ be the accumulated width. Initialize $(\overline{T}_1, \overline{S}_1) = (T_1, S_1)$ and, for $2 \le j \le \nu_r$, set

$$\begin{aligned} X_j &= S_j^{\mathsf{T}} \overline{T}_{j-1}, \\ \overline{T}_j &= [\,\overline{T}_{j-1} + T_j X_j \;\; T_j\,], \\ \overline{S}_j &= [\,\overline{S}_{j-1} \;\; S_j\,]. \end{aligned} \tag{40}$$

The factors lie in $\mathbb{F}^{o \times r_j}$, and direct expansion gives

$$G_j \cdots G_1 = I_o + \overline{T}_j \overline{S}_j^{\mathsf{T}}. \tag{41}$$

Initialize (A, b) with all rows available. Each step uses eager elimination's selectors, success bit, and availability update; transform later local pivot panels and extend the factors. Distant panels and the right-hand side retain block-entry values until the boundary, then receive $W \mapsto W + \overline{T}_{\nu_r}(\overline{S}_{\nu_r}^{\mathsf{T}} W)$. Inductively, pivot panels equal their original values under all preceding transforms; Equation (41) maintains this invariant at block boundaries. Thus pivots, the transformed right-hand side, and the output defined in Section 3.3 match eager execution for every public schedule.