

# Levis: Extension-Free Multi-Key Fully Homomorphic Encryption in the Plain Model

Baoyu Li<sup>1,3</sup> and Binwu Xiang<sup>2,3</sup> Kang Yang<sup>3</sup>

<sup>1</sup> Shanghai Jiao Tong University

<sup>2</sup> East China Normal University

<sup>3</sup> State Key Laboratory of Cryptology

**Abstract.** Multi-key fully homomorphic encryption (MKFHE) enables computation over ciphertexts encrypted under keys independently generated by different parties. Recently, Min, Park, and Song (MPS, CRYPTO 2026) constructed the first RLWE-based MKFHE scheme in the plain model, i.e., without trusted or interactive setup. Their construction requires each local compact Gentry–Sahai–Waters (GSW) ciphertext to be extended to a joint-key GSW-like ciphertext before evaluation, incurring  $O(dk^2)$  gadget products per input for  $k$  parties and gadget length  $d$ .

In this paper, we first construct an extension-free MKFHE scheme (called **Levis-fast**) in the plain model, eliminating the  $O(dk^2)$  gadget products required for joint-key ciphertext extension before homomorphic evaluation. Our core technique is a compatibility technique, termed *Cofactor Compatibility*, which establishes a common projective decryption relation across ciphertexts encrypted under independently generated local keys, enabling their direct homomorphic evaluation. We further present a compact variant (named as **Levis-small**) that reduces the communication of each fresh ciphertext from  $O(dk^2)$  to  $O(d)$  ring elements with only a lightweight lifting operation. A proof-of-concept implementation shows that, compared with MPS, for  $k \in \{2, 4, 8\}$ , our construction **Levis-fast** achieves evaluation speedups of  $67.7\times$ ,  $52.8\times$ , and  $139.4\times$ , respectively, while our compact variant **Levis-small** reduces communication by a factor of  $2.5\times$ ,  $8.7\times$  and  $14.2\times$ , and achieves evaluation speedups of  $11.8\times$ ,  $19.7\times$ , and  $54.4\times$ , respectively.

## 1 Introduction

Multi-key fully homomorphic encryption (MKFHE), introduced by López-Alt, Tromer, and Vaikuntanathan [22], is a powerful primitive that enables homomorphic computation on ciphertexts encrypted under different keys. An interesting direction is to realize this primitive in the plain model, i.e., without interactive setup or reliance on a trusted third party, which is useful for real-world applications, such as decentralized blockchains [21], and for building low-round MPC protocols in the plain model [25].

Earlier constructions face a trade-off between setup requirements and efficiency. The first MKFHE scheme, proposed by López-Alt, Tromer, and Vaikuntanathan [22], requires neither a common reference string (CRS) nor interactive

setup. However, it relies on the decisional small polynomial ratio (DSPR) assumption and uses a multiplicative joint secret key, resulting in multi-round decryption and noise growth exponential in the number of parties. This noise growth makes it difficult to instantiate the scheme with secure NTRU parameters [2,12,16,17,19]. Subsequent (R)LWE-based MKFHE constructions support one-round decryption but rely on the CRS model [13,26,8,29,10,9,31].

Recently, Min, Park, and Song (MPS) [25] constructed the first RLWE-based MKFHE scheme in the plain model with only one-round decryption. Their construction heavily relies on so-called compact Gentry–Sahai–Waters (cGSW) ciphertexts, which require a *ciphertext extension* to ring GSW (RGSW) ciphertexts under a joint key before the subsequent homomorphic evaluation. For  $k$  parties and gadget length  $d$ , this extension requires  $O(dk^2)$  gadget products, which scales quadratically with the number of parties and is much more expensive than the homomorphic operation itself, which requires only  $O(1)$  gadget products. Moreover, each fresh cGSW ciphertext contains  $O(dk^2)$  ring elements, incurring a communication cost that also grows quadratically with the number of parties. This motivates the following fundamental question:

*Can we construct an MKFHE scheme in the plain model with one-round decryption, without costly ciphertext extension and with per-input communication independent of the number of parties?*

## 1.1 Our Contributions

In this paper, we propose a new MKFHE scheme (called **Levis-fast**) *without* ciphertext extension in the plain model, which enables independently generated local GSW ciphertexts to be evaluated straightforwardly. We further present a compact variant (referred to as **Levis-small**), which reduces the per-input communication to be *independent* of the number  $k$  of parties. Our constructions can also be combined with the generic transformation by Ananth et al. [4], as in MPS [25], to obtain one-round decryption. Another useful feature is that, unlike MPS [25, Section 4.1], both ciphertexts and evaluation keys of our constructions are independent of the multiplication order among the participating parties, and so changing the evaluation order requires neither ciphertext modification nor key switching. Specifically, our results are summarized below.

**Cofactor compatibility.** Our core technique is a compatibility technique called *Cofactor Compatibility*. See Section 1.2 for technical details. It establishes a common projective decryption relation across cGSW ciphertexts encrypted under independently generated local keys. As a result, it enables us to evaluate local cGSW ciphertexts straightforwardly, without converting them to the joint-key RGSW representation. Furthermore, existing ciphertexts remain unchanged when the number of parties grows, and no extension key is required when a previously joined party later supplies a fresh, local ciphertext.

**Compact ciphertexts with fused evaluation.** We further present a ciphertext-compression technique to reduce the communication cost of each fresh ciphertext.

**Table 1.** Comparison of MKFHE schemes in the plain model between the state-of-the-art construction MPS [25] and our constructions, where  $k$  is the number of parties and  $d$  denotes the gadget length.

|                         | MPS [25]  | Levis-fast (ours) | Levis-small (ours)   |
|-------------------------|-----------|-------------------|----------------------|
| Post-join computation   | $O(dk^2)$ | $O(1)$            | $O(1 + \frac{d}{k})$ |
| Post-join communication | $O(dk^2)$ | $O(dk^2)$         | $O(d)$               |
| Size of evaluation keys | $O(dk^4)$ | –                 | $O(dk^3)$            |
| Free join               | ×         | ✓                 | ✓                    |

In particular, each party generates an independent auxiliary key and publishes a reusable local hint in the key-generation phase. It can then encrypt each fresh message as an  $O(d)$ -element gadget-level encryption (GLEV) ciphertext, which is independent of the number  $k$  of parties. In the evaluation phase, we let the evaluator fuse the lifting operation which transforms the gadget-level ciphertext into a cGSW ciphertext with the subsequent vector external product, which enables us to reduce the computational cost. See Section 1.2 for more details.

**Theoretical comparison.** Table 1 compares Levis-fast and Levis-small with MPS [25] in terms of computation and communication costs in the *post-join* setting. In this table, the computation counts the number of polynomial multiplications divided by the cost of one  $k$ -party vector gadget product, and the communication counts the number of ring polynomials. We also compare the total size of the evaluation keys from all  $k$  parties, which counts the total number of ring polynomials. We say that an MKFHE scheme supports “free join”, if a party can join the system without the need to perform computation, when the party provides no input message.

**Implementation and evaluation.** We implemented Levis-fast and Levis-small in C++ for the 128-bit security level, and report the performance for  $k \in \{2, 4, 8\}$  in the *post-join* setting, where one party joins a fresh ciphertext into the homomorphic evaluation after all  $k$  parties have generated ciphertexts in the initialization phase. Compared to MPS [25], Levis-fast improves the evaluation time by a factor of  $52.8 \sim 139.4\times$ , and Levis-small reduces the communication cost by a factor of  $2.5 \sim 14.2\times$  and speeds up the runtime by  $11.8 \sim 54.4\times$ .

## 1.2 Technical Overview

We focus on the MKFHE in the plain model. Consider a system with at most  $K_{\max}$  parties and an evaluator. Initially, there are no parties in the system. The parties join the system one by one, starting from party  $i_1$ . After joining, a party can upload ciphertexts to the evaluator. The evaluator uses them for further homomorphic computation. After the computation is completed, the parties jointly perform distributed decryption to recover the plaintext. Let  $\mathcal{K} = (i_1, \dots, i_k)$  be

the current party set, where  $k = |\mathcal{K}|$ , and let  $n = K_{\max} + 1$  for system capacity  $K_{\max}$ .

**Prior solution.** In MPS [25], each party  $i_j \in \mathcal{K}$  independently samples a secret vector  $\mathbf{s}_{i_j} \in \mathcal{R}^j$ . When a party wants to join the system, it encrypts its message  $\mu_{i_j}$  as a local cLEV ciphertext  $L_{i_j} \in \mathcal{R}_Q^{(j+1) \times d_j}$  satisfying  $A_{i_j} L_{i_j} = \mu_{i_j} G + pE_{i_j}$ , where  $A_{i_j} = [I_j \mid \mathbf{s}_{i_j}]$  is the local decryption map. The evaluator has a trivial accumulator ciphertext  $\text{acc}^0 = 1$  and a trivial joint key  $\mathbf{s}^0 = 1$ . The parties join the system one by one in a fixed order  $(i_1 \rightarrow \dots \rightarrow i_k), k \leq K_{\max}$ . When a new party  $i_j$  joins, it uploads a cLEV ciphertext  $L_{i_j}$  which encrypts  $\mu_{i_j}$  under the decryption map  $A_{i_j}$ . The evaluator uses a vector gadget product to update both the accumulator  $\text{acc}^j \leftarrow L_{i_j} \text{D}(\text{acc}^{j-1})$  explicitly and the joint key  $\mathbf{s}^j \leftarrow \mathbf{s}^{j-1} \cdot A_{i_j}$  implicitly, which gives  $\mathbf{s}^j \text{acc}^j = \mathbf{s}^{j-1} A_{i_j} L_{i_j} \text{D}(\text{acc}^{j-1}) = \mu_{i_j} \mathbf{s}^{j-1} G \text{D}(\text{acc}^{j-1}) = \mu_{i_j} \mu_{i_{j-1}} \dots \mu_{i_1}$ .

Once the joint key has been established, suppose an existing party  $i_j$  with  $j < k$  wants to perform an external product with the accumulator. Since party  $i_j$  can only generate a local cGSW ciphertext  $C_{i_j}$  satisfying  $A_{i_j} C_{i_j} = \mu_{i_j} A_{i_j} G + pE_{i_j}$  under its decryption map  $A_{i_j}$ , this ciphertext cannot be interpreted under the joint key. Moreover, its dimension differs from the accumulator, so it cannot be used directly in the computation. To resolve this issue, MPS introduces an RGSW extension procedure, which homomorphically extends the local cGSW ciphertext along the party order into an RGSW ciphertext under the joint key before performing the external product. After party  $i_j$  uploads a cGSW ciphertext, the evaluator uses secret-key-dependent evaluation keys provided by parties  $i_{j+1}, \dots, i_k$  to extend the cGSW ciphertext into an RGSW ciphertext under the joint key through vector gadget products recursively.

However, this design has two main bottlenecks. To extend a local cGSW ciphertext of party  $i_j$  to the current joint-key domain, the evaluator uses the secret-key-dependent evaluation keys of the subsequently joined parties and repeatedly applies vector gadget products along the party order. In MPS, this extension requires  $O(dk^2)$  vector gadget products in the worst case, where  $d$  is the gadget decomposition length. These operations, which involve gadget decomposition and ring-polynomial multiplications, must be performed before the actual external product and therefore introduce a significant pre-product cost. The communication cost is also substantial. A local cGSW ciphertext of party  $i_j$  has dimension  $(j+1) \times d(j+1)$ , and thus contains  $d(j+1)^2$  ring elements. In the worst case, where  $j = \Theta(k)$ , each fresh cGSW ciphertext therefore requires  $O(dk^2)$  ring elements to be transmitted to the evaluator. Hence, both the computation required for RGSW extension and the communication of fresh cGSW ciphertexts grow quadratically with the number of parties.

**Our solution.** In our construction, each party  $i$  independently samples a secret vector  $\mathbf{s}_i \in R^{K_{\max}}$  and defines the local decryption map  $A_i = [I_{K_{\max}} \mid \mathbf{s}_i]$ . It encrypts a message  $\mu_i$  as a local cGSW ciphertext  $C_i \in \mathcal{R}_{QP}^{n \times d_n}$ , satisfying  $A_i C_i = \mu_i A_i G + pE_i$ .

To address the extension bottleneck, we introduce *Cofactor Compatibility*. For  $\mathcal{K} = (i_1, \dots, i_k)$ , let  $\mathbf{s}_i^{(k)}$  contain the first  $k$  coordinates of  $\mathbf{s}_i$ ,  $(V_{\mathcal{K}})_{\hat{j}}$  denote the matrix obtained by deleting column  $j$ , and define

$$V_{\mathcal{K}} = \begin{pmatrix} (\mathbf{s}_{i_1}^{(k)})^\top & -1 \\ \vdots & \vdots \\ (\mathbf{s}_{i_k}^{(k)})^\top & -1 \end{pmatrix}, \quad \hat{h}_j = (-1)^j \det(V_{\mathcal{K}})_{\hat{j}}.$$

We denote  $\hat{\mathbf{h}}_{\mathcal{K}} = (\hat{h}_0, \dots, \hat{h}_k)^\top$ , and define  $\mathbf{h}_{\mathcal{K}} = (\hat{h}_0, \dots, \hat{h}_{k-1}, 0, \dots, 0, \hat{h}_k)^\top \in R^{K_{\max}+1}$ , and  $\mathbf{w}_{\mathcal{K}} = (\hat{h}_0, \dots, \hat{h}_{k-1}, 0, \dots, 0)^\top \in R^{K_{\max}}$ . The signed-minor identity yields  $V_{\mathcal{K}} \hat{\mathbf{h}}_{\mathcal{K}} = 0$  and hence  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K}}^\top A_i$  for every  $i \in \mathcal{K}$ . Applying this identity to the local cGSW relation gives  $\mathbf{h}_{\mathcal{K}}^\top C_i = \mu_i \mathbf{h}_{\mathcal{K}}^\top G + p \mathbf{w}_{\mathcal{K}}^\top E_i$  for every party. Although the local maps  $A_i$  differ, their ciphertexts therefore share one projective decryption relation.

Therefore, joining a new party does not require updating a joint key or converting existing ciphertexts. Existing ciphertexts remain unchanged, and the enlarged party set only induces a new common decryption direction for decryption. For an external product, suppose a vector ciphertext  $\mathbf{c}_x$  satisfies  $\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}_x = x h_{\mathcal{K},0} + p e_x$ , and party  $i \in \mathcal{K}$  uploads a local cGSW ciphertext  $C_i$  encrypting  $\mu_i$ . Since  $C_i$  already satisfies the common cofactor relation under  $\mathbf{h}_{\mathcal{K}}$ , the evaluator can directly compute  $\mathbf{c}_{x\mu_i} = \text{ExtProd}(C_i, \mathbf{c}_x)$ , where  $\text{ExtProd}(C, \mathbf{c}) = \text{ModDown}_P(\text{CD}(\mathbf{c}))$ . The resulting ciphertext satisfies  $\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}_{x\mu_i} = x \mu_i h_{\mathcal{K},0} + p e'$ . Thus, when a previously joined party later participates in an external product, no RGSW extension is needed, reducing the pre-product work from  $O(dk^2)$  vector gadget products to  $O(1)$ .

To address the second bottleneck, we introduce the *Compact-Input Technique*. Party  $i$  samples an independent bridge key  $t_i$  and publishes once a reusable hint  $H_i \in \mathcal{R}_{QP}^{n \times dn}$  satisfying  $A_i H_i = t_i A_i G + p E_i^H \pmod{QP}$ . A fresh encryption of a message  $\mu$  contains the  $d$  pairs  $(a_\ell, b_\ell) \in \mathcal{R}_Q^2$  satisfying  $b_\ell + t_i a_\ell = \mu \bar{g}_\ell + p e_\ell \pmod{Q}$ , where  $\bar{g}_\ell = Q/q_\ell$  and  $e_\ell$  is fresh error. Hence each fresh ciphertext contains  $2d$  ring elements instead of a  $dn^2$ -element cGSW matrix. The pair  $(a_\ell, b_\ell)$  and the reusable hint define the conceptual cGSW block

$$C_\mu^{(\ell)} = P b_\ell I_n + H_i D(a_\ell I_n), \quad A_i C_\mu^{(\ell)} = \mu g_\ell A_i + p \hat{E}_{i,\ell} \pmod{QP}.$$

The evaluator does not construct these blocks. Before modulus down, it reassociates the external product as  $C_\mu D(\mathbf{c}_x) = P \sum_\ell b_\ell \mathbf{v}_\ell + H_i \sum_\ell D(a_\ell I_n) \mathbf{v}_\ell$ , where  $D(\mathbf{c}_x) = (\mathbf{v}_0, \dots, \mathbf{v}_{d-1})$ . This fused computation has the same algebraic result without transmitting or materializing the full cGSW input. After the reusable hint is published, communicating each fresh ciphertext requires  $O(d)$  rather than  $O(dK_{\max}^2)$  ring elements.

### 1.3 Related Work

*Multi-key FHE in the plain model.* MKFHE was introduced by López-Alt, Tromer, and Vaikuntanathan [22], followed by constructions based on learn-

ing with errors (LWE) and RLWE that support dynamic and multi-hop evaluation [13,26,8,29,11]. More recent works have focused on reducing setup assumptions and ciphertext-extension costs. Daza et al. [15] proposed a leveled RLWE-based MKFHE scheme without ciphertext expansion in the CRS model, while Dai et al. [14] replaced conventional expansion with an interactive key-lifting procedure. Luo et al. [23] further studied RLWE-based MKFHE without a CRS.

Most closely related to our work, MPS [25] gave the first RLWE-based MKFHE with non-interactive setup in the plain model. Their construction introduces compact leveled ciphertexts (cLEV) and cGSW ciphertexts and extends local cGSW ciphertexts to RGSW ciphertexts under a nonlinear joint key for multi-hop evaluation. Wang, Huang, and Tang [30] subsequently obtained dynamic multi-key GSW without a CRS, while still retaining a ciphertext-extension procedure. In contrast, our construction leaves local cGSW ciphertexts unchanged and obtains cross-key compatibility through a common projective decryption direction, eliminating ciphertext extension and extension key size.

*Efficient MKFHE.* A complementary line of work improves the concrete cost of multi-key evaluation and bootstrapping. Chen et al. [9] introduced multi-key Torus fully homomorphic encryption (TFHE), while Kim et al. [18] improved multi-key homomorphic evaluation through homomorphic gadget decomposition. Kwak et al. [20] later reduced the complexity of multi-key TFHE bootstrapping to  $O(dkn)$ . Recent NTRU-based constructions, including FINALLY [28] and Xiang et al. [32], further improved multi-key bootstrapping and key-scaling costs. Most recently, HERDS [31] reduced the serial bootstrapping complexity to  $O(kn)$  and to  $O(\sqrt{kn})$  under parallelization, together with an efficient distributed-decryption protocol. Afshar and Goyal [1] pursued a different direction and obtained fully succinct MKFHE from LWE. These works mainly optimize bootstrapping, ciphertext size, or key scaling, whereas our work focuses on the compatibility of independently generated GSW ciphertexts before such higher-level evaluation procedures.

## 2 Preliminaries

For a positive integer  $r$ , let  $[r] = \{1, \dots, r\}$ . We write vectors and matrices in bold and use  $(\cdot)^\top$  for transpose. The symbol  $I_r$  denotes the  $r \times r$  identity matrix. We write  $x \leftarrow \mathcal{D}$  for sampling from a distribution  $\mathcal{D}$  and  $U(T)$  for the uniform distribution over a finite set  $T$ . Let  $N$  be a power of two,  $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ , and  $\mathcal{R}_Q = \mathcal{R}/Q\mathcal{R}$ , where  $Q$  is odd. Let  $p$  be a prime plaintext modulus satisfying  $\gcd(p, Q) = 1$ , and let  $\mathcal{R}_p = \mathcal{R}/p\mathcal{R}$  be the plaintext ring. Every  $a \in \mathcal{R}$  (and likewise every residue class) is represented by  $a(X) = \sum_{i=0}^{N-1} a_i X^i$ . For  $u \in \mathcal{R}$ ,  $\|u\|_p$  is the coefficient  $\ell_p$  norm. For a ring-valued matrix,  $\|E\|_\infty$  is the largest coefficient magnitude among all entries. Unless stated otherwise, norms are taken before reduction modulo  $Q$ , and  $[z]_Q$  denotes the coefficientwise centered lift of  $z \in \mathcal{R}_Q$ . We use  $\|uv\|_\infty \leq \|u\|_1 \|v\|_\infty$  and  $\|uv\|_1 \leq \|u\|_1 \|v\|_1$ . Let  $\chi$  and  $\psi$  be

key and error distributions over  $\mathcal{R}$ . Let  $K_{\max}$  denote the maximum number of parties supported by the public parameters. For a computation,  $\mathcal{K} \subseteq [K_{\max}]$  denotes its current party set and  $k = |\mathcal{K}|$  its number of parties. Ciphertext and key dimensions depend on  $K_{\max}$  and remain fixed when  $\mathcal{K}$  grows.

*Signed maximal minors.* For a matrix  $V \in \mathcal{R}^{k \times (k+1)}$  over a commutative ring, let  $V_{\hat{j}}$  denote the matrix obtained by deleting column  $j$ , and define  $c_j = (-1)^j \det(V_{\hat{j}})$  for  $0 \leq j \leq k$ . Laplace expansion gives  $V\mathbf{c} = 0$ . Permuting the rows of  $V$  multiplies every coordinate of  $\mathbf{c}$  by the same sign. Both properties follow from the determinant expansion and its alternating property [27].

## 2.1 Ring Learning with Error

**Definition 1 (Decisional RLWE).** For a modulus  $q$ , let  $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ . For  $s \leftarrow \chi$ , sample  $a \leftarrow U(\mathcal{R}_q)$  and  $e \leftarrow \psi$ , and output  $(b, a) \in \mathcal{R}_q^2$ , where  $b = -as + e \pmod{q}$ . The problem  $\text{RLWE}(N, q, \psi, \chi)$  is to distinguish this distribution from  $U(\mathcal{R}_q^2)$  [24].

**Definition 2 (Decisional shared-A RLWE).** For  $\mathbf{s} = (s_1, \dots, s_r) \leftarrow \chi^r$ , sample one common  $a \leftarrow U(\mathcal{R}_q)$  and independent  $e_i \leftarrow \psi$ , output  $(b_1, \dots, b_r, a) \in \mathcal{R}_q^{r+1}$ , where  $b_i = -as_i + e_i \pmod{q}$ . The shared-A RLWE problem ( $\text{sARLWE}$ ), denoted  $\text{sARLWE}(r, N, q, \psi, \chi)$ , is to distinguish this distribution from  $U(\mathcal{R}_q^{r+1})$  [25].

**Lemma 1 (RLWE implies shared-A RLWE).** If one-sample  $\text{RLWE}(N, q, \psi, \chi)$  has distinguishing advantage at most  $\varepsilon$ , then rank- $r$  shared-A RLWE has advantage at most  $r\varepsilon$  [5, 6, 25].

*Proof.* Replace the  $r$  responses one at a time. In the hybrid for coordinate  $j$ , a one-sample RLWE challenge supplies the common mask  $a$  and the  $j$ -th response. The reduction samples all other secrets and errors itself and forms their responses using the same  $a$ . Previously replaced responses are uniform. Thus sharing  $a$  does not obstruct the hybrid, and the triangle inequality gives the factor  $r$ .

## 2.2 Multi-Key Fully Homomorphic Encryption

We adapt the syntax used by MPS [25], which follows López-Alt et al. [22], to fixed-capacity leveled evaluation.

**Definition 3 (Leveled multi-key fully homomorphic encryption).** A leveled MKFHE scheme is a family of probabilistic polynomial-time (PPT) algorithms

$$\Pi^{(K_{\max}, L, M)} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec}), \quad K_{\max}, L, M > 0,$$

with the following syntax.

- $pp \leftarrow \text{Setup}(1^\lambda, 1^{K_{\max}}, 1^L, 1^M)$  generates public parameters for security parameter  $\lambda$ , party capacity  $K_{\max}$ , circuit depth  $L$ , and lifted-wire bound  $M$ .

- $(sk_i, ek_i, pk_i) \leftarrow \text{KeyGen}(pp)$  independently generates a secret key, public evaluation keys, and a public key for party  $i$ .
- $(ct, \{i\}) \leftarrow \text{Enc}(pp, pk_i, m)$  encrypts  $m \in \mathcal{R}_p$  and attaches the singleton party label  $\{i\}$ .
- A party-side compact-input extension may define  $(ct, \{i\}) \leftarrow \text{CEnc}(pp, sk_i, m)$ . Its output is supplied to **Eval** together with  $ek_i$ .
- $(ct_f, \mathcal{K}) \leftarrow \text{Eval}(pp, f, \{(pk_i, ek_i)\}_{i \in \mathcal{K}}, \{(ct_j, \{i_j\})\}_{j \in [\ell]})$  evaluates an arithmetic circuit  $f$  and sets  $\mathcal{K} = \{i_j : j \in [\ell]\}$ , subject to  $k = |\mathcal{K}| \leq K_{\max}$ .
- $m \leftarrow \text{Dec}(pp, \{sk_i\}_{i \in \mathcal{K}}, (ct, \mathcal{K}))$  returns a message in  $\mathcal{R}_p$  or the failure symbol  $\perp$ .

In this paper, a supported vector-accumulator circuit is an arithmetic circuit with an evaluation order in which every multiplication applies an input cGSW matrix to an existing vector ciphertext. Additions act on vector ciphertexts. The scheme satisfies the following properties.

- **Correctness.** For honestly generated keys and ciphertexts, every supported vector-accumulator circuit  $f$  of depth at most  $L$  whose lifted wire values have coefficient  $\ell_1$  norm at most  $M$  satisfies

$$\Pr [\text{Dec}(pp, \{sk_i\}_{i \in \mathcal{K}}, (ct_f, \mathcal{K})) = f(m_1, \dots, m_\ell)] \geq 1 - \text{negl}(\lambda),$$

where equality is in  $\mathcal{R}_p$  and  $(ct_f, \mathcal{K})$  is returned by **Eval** on the corresponding outputs of **Enc** or **CEnc**.

- **Compactness.** There exists a polynomial  $\text{poly}$  such that the size of  $ct_f$  and the running time of **Dec** are bounded by  $\text{poly}(\lambda, K_{\max}, L, \log M)$ , independently of the size of  $f$ .

### 2.3 Gadget Decomposition, cLEV, and cGSW

Following the gadget notation of MPS [25, Definition 5], let  $\mathbf{g} = (g_0, \dots, g_{d-1}) \in \mathcal{R}_{QP}^d$ .

**Definition 4 (Scaled gadget decomposition).** A map  $D : \mathcal{R}_Q \rightarrow \mathcal{R}_{QP}^d$  is a  $P$ -scaled decomposition with respect to  $\mathbf{g}$  if, for every  $x \in \mathcal{R}_Q$ ,

$$\|D(x)\|_\infty \leq B_g, \quad \langle \mathbf{g}, D(x) \rangle = Px \pmod{QP}.$$

The map is applied entrywise to vectors and matrices. For dimension  $m$ , set

$$G_m = \mathbf{g}^\top \otimes I_m = [g_0 I_m \cdots g_{d-1} I_m]. \quad (1)$$

For every compatible  $X$  over  $\mathcal{R}_Q$ ,

$$G_m D(X) = PX \pmod{QP}. \quad (2)$$

We instantiate this definition with an RNS basis  $Q = \prod_{\ell=0}^{d-1} q_\ell$  of pairwise coprime moduli. Let  $P$  be coprime to  $pQ$ , set  $Q_\ell = Q/q_\ell$ , and take  $g_\ell = PQ_\ell$ . The

$\ell$ -th component of  $D(x)$  is the base extension to  $\mathcal{R}_{QP}$  of  $(x \bmod q_\ell)(Q_\ell^{-1} \bmod q_\ell)$ . Thus  $d$  counts the  $Q$  limbs, whereas  $P$  is used only for arithmetic over  $\mathcal{R}_{QP}$  and modulus down. We write  $n = K_{\max} + 1$  and  $G = G_n$  when the dimension is clear.

For  $z \in \mathcal{R}_{QP}$ , choose a coefficientwise correction  $p\delta(z)$  satisfying  $p\delta(z) \equiv z \pmod{P}$ , and define

$$\text{ModDown}_P(z) = \frac{z - p\delta(z)}{P} \pmod{Q}. \quad (3)$$

The map is applied entrywise, and  $B_\delta$  bounds the coefficients of  $\delta(z)$ .

**Definition 5 (Vector external product).** For  $C \in \mathcal{R}_{QP}^{m \times dm}$  and  $c \in \mathcal{R}_Q^m$ , define

$$\text{ExtProd}(C, c) = \text{ModDown}_P(CD(c)) \in \mathcal{R}_Q^m. \quad (4)$$

This is the RNS analogue of the vector external product of MPS [25, Definition 9].

**Definition 6 (Shared- $A$ , cLEV, and cGSW relations).** A shared- $A$  ciphertext of  $\mu \in \mathcal{R}_p^r$  under  $s \in \mathcal{R}^r$  is  $c = (b_1, \dots, b_r, a)$  satisfying

$$A_s c = \mu + pe \pmod{Q}, \quad A_s = [I_r \mid s].$$

A cLEV ciphertext of  $\mu \in \mathcal{R}_p$  relative to the same map is a matrix  $L \in \mathcal{R}_{QP}^{(r+1) \times dr}$  satisfying

$$A_s L = \mu G_r + pE_L \pmod{QP}.$$

A cGSW ciphertext of  $\mu \in \mathcal{R}_p$  relative to the same map is a matrix  $C \in \mathcal{R}_{QP}^{(r+1) \times d(r+1)}$  satisfying

$$A_s C = \mu A_s G_{r+1} + pE \pmod{QP}$$

for a small error matrix  $E$ .

These are the cLEV and cGSW formats of MPS [25, Section 3.1], with errors scaled by the plaintext modulus. RGSW uses a scalar left decryption row, whereas cGSW uses the shared- $A$  map  $[I_r \mid s]$ .

### 3 Cofactor Compatibility

In plain-model multi-key GSW evaluation, ciphertexts generated under unrelated local decryption maps are not directly compatible. MPS [25] resolves this mismatch by extending each local cGSW ciphertext to an RGSW ciphertext under a joint key. We instead derive a common cofactor relation for the local maps and use it to evaluate their ciphertexts without conversion.

### 3.1 Multi-Key Somewhat Homomorphic Encryption (SHE) with a Common Decryption Direction

Fix a current party set  $\mathcal{K} \subseteq [K_{\max}]$  with  $k = |\mathcal{K}| \geq 1$ . Party  $i \in \mathcal{K}$  has a local decryption map  $A_i = [I_{K_{\max}} \mid \mathbf{s}_i]$ . Suppose its cGSW ciphertext  $C_i \in \mathcal{R}_{QP}^{(K_{\max}+1) \times d(K_{\max}+1)}$  encrypts  $\mu_i \in \mathcal{R}_p$  and satisfies

$$A_i C_i = \mu_i A_i G + p E_i \pmod{QP}. \quad (5)$$

**Definition 7 (Common decryption direction).** A nonzero vector  $\mathbf{h}_{\mathcal{K}} \in \mathcal{R}^{K_{\max}+1}$  is a common decryption direction for  $\mathcal{K}$  if, for every  $i \in \mathcal{K}$ , there exists  $\mathbf{w}_{\mathcal{K},i} \in \mathcal{R}^{K_{\max}}$  such that  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K},i}^\top A_i$ , or equivalently, if  $\mathbf{h}_{\mathcal{K}}^\top$  lies in the intersection of the row modules of the local maps  $A_i$ .

Write  $\mathcal{K} = (i_1, \dots, i_k)$  in canonical order. Here  $i_1, \dots, i_k$  identify parties, while the second index in  $\mathbf{s}_{i,j}$  identifies a coordinate of party  $i$ 's fixed-capacity secret vector. For each party, let  $\mathbf{s}_i^{(k)} = (s_{i,0}, \dots, s_{i,k-1})^\top$  be the first  $k$  coordinates of  $\mathbf{s}_i$ . Form the  $k \times (k+1)$  matrix

$$V_{\mathcal{K}} = \begin{pmatrix} (\mathbf{s}_{i_1}^{(k)})^\top & -1 \\ \vdots & \vdots \\ (\mathbf{s}_{i_k}^{(k)})^\top & -1 \end{pmatrix}, \quad \hat{h}_j = (-1)^j \det(V_{\mathcal{K}})_{\hat{j}}, \quad (6)$$

where  $(V_{\mathcal{K}})_{\hat{j}}$  deletes column  $j$  and  $0 \leq j \leq k$ . Embed the cofactor vector  $\hat{\mathbf{h}}_{\mathcal{K}} = (\hat{h}_0, \dots, \hat{h}_k)^\top$  and its head in the fixed ambient dimensions as

$$\begin{aligned} \mathbf{h}_{\mathcal{K}} &= (\hat{h}_0, \dots, \hat{h}_{k-1}, 0, \dots, 0, \hat{h}_k)^\top \in \mathcal{R}^{K_{\max}+1}, \\ \mathbf{w}_{\mathcal{K}} &= (\hat{h}_0, \dots, \hat{h}_{k-1}, 0, \dots, 0)^\top \in \mathcal{R}^{K_{\max}}. \end{aligned}$$

**Lemma 2 (Common cofactor relation).** Over the commutative ring  $\mathcal{R}$ ,  $V_{\mathcal{K}} \hat{\mathbf{h}}_{\mathcal{K}} = 0$  and  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K}}^\top A_i$  for every  $i \in \mathcal{K}$ . If  $\mathbf{h}_{\mathcal{K}} \neq 0$ , it is a common decryption direction for  $\mathcal{K}$ . In particular,  $h_{\mathcal{K},0} \bmod p \in \mathcal{R}_p^\times$  implies  $\mathbf{h}_{\mathcal{K}} \neq 0$  and makes  $h_{\mathcal{K},0}$  an invertible decoding coefficient. A permutation of the parties multiplies all coordinates of  $\mathbf{h}_{\mathcal{K}}$  by the same sign.

*Proof.* The signed-maximal-minor identity from Section 2 gives  $V_{\mathcal{K}} \hat{\mathbf{h}}_{\mathcal{K}} = 0$ . Hence  $(\mathbf{s}_i^{(k)})^\top (\hat{h}_0, \dots, \hat{h}_{k-1})^\top = \hat{h}_k$ , which proves  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K}}^\top A_i$  after embedding. The row-permutation statement follows from the same identity.

**Lemma 3 (Cofactor compatibility).** For every  $i \in \mathcal{K}$ , an unmodified local ciphertext satisfying (5) also satisfies  $\mathbf{h}_{\mathcal{K}}^\top C_i = \mu_i \mathbf{h}_{\mathcal{K}}^\top G + p \mathbf{w}_{\mathcal{K}}^\top E_i \pmod{QP}$ . When  $h_{\mathcal{K},0} \bmod p$  is a unit, this is a common projective relation with a valid decoder obtained by inverting  $h_{\mathcal{K},0}$  in  $\mathcal{R}_p$ .

*Proof.* By Lemma 2,  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K}}^\top A_i$  for every  $i \in \mathcal{K}$ . Therefore, left-multiplying (5) by  $\mathbf{w}_{\mathcal{K}}^\top$  gives  $\mathbf{h}_{\mathcal{K}}^\top C_i = \mu_i \mathbf{h}_{\mathcal{K}}^\top G + p \mathbf{w}_{\mathcal{K}}^\top E_i \pmod{QP}$ .

*Projective decryption.* The framework requires a decodable common direction rather than the fixed normalization  $h_{\mathcal{K},0} = 1$  in  $\mathcal{R}_Q$ . Scaling  $\mathbf{h}_{\mathcal{K}}$  and the corresponding phase by the same factor preserves the decoded message when that factor is a unit modulo  $p$  and the scaled phase still satisfies the no-wrap condition. Compatibility therefore depends on the common direction rather than a chosen normalization. Forcing  $h_{\mathcal{K},0} = 1$  during evaluation would require division by  $h_{\mathcal{K},0}$  in  $\mathcal{R}_Q$ . Such an inverse may not exist. Even when it exists, its centered representative may be large enough to destroy the small-error relation. We therefore retain the unnormalized cofactor vector during evaluation.

Suppose the final vector ciphertext satisfies  $\mathbf{h}_{\mathcal{K}}^\top \mathbf{c} = \mu h_{\mathcal{K},0} + pe \pmod{Q}$ . The decoder takes the centered lift, reduces it modulo  $p$ , and normalizes in the plaintext ring as

$$\mu = (h_{\mathcal{K},0} \bmod p)^{-1} ([\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}]_Q \bmod p).$$

This succeeds whenever  $h_{\mathcal{K},0} \bmod p$  is a unit in  $\mathcal{R}_p$  and the phase has not wrapped modulo  $Q$ . Theorem 1 bounds the probability that the unit condition fails for honestly generated keys, and Theorem 2 establishes the required no-wrap condition.

### 3.2 Extension-Free Multi-Key GSW Evaluation

For a party set  $\mathcal{K}$ , every local cGSW ciphertext satisfies the same cofactor relation. On the event  $h_{\mathcal{K},0} \bmod p \in \mathcal{R}_p^\times$ , this relation gives a common projective decryption direction.

**Lemma 4 (RNS external-product relation).** *Let  $C_i$  satisfy Lemma 3, and let a vector ciphertext satisfy  $\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}_x = xh_{\mathcal{K},0} + pe_x \pmod{Q}$ . Set  $\mathbf{z} = C_i D(\mathbf{c}_x)$  and let  $p\delta \equiv \mathbf{z} \pmod{P}$  be the correction used by  $\text{ModDown}_P$ . Then*

$$\mathbf{h}_{\mathcal{K}}^\top \text{ExtProd}(C_i, \mathbf{c}_x) = x\mu_i h_{\mathcal{K},0} + p \left( \mu_i e_x + \frac{\mathbf{w}_{\mathcal{K}}^\top E_i D(\mathbf{c}_x) - \mathbf{h}_{\mathcal{K}}^\top \delta}{P} \right) \pmod{Q}.$$

*Proof.* The compatibility relation and (2) give

$$\begin{aligned} \mathbf{h}_{\mathcal{K}}^\top \mathbf{z} &= \mu_i P \mathbf{h}_{\mathcal{K}}^\top \mathbf{c}_x + p \mathbf{w}_{\mathcal{K}}^\top E_i D(\mathbf{c}_x) \\ &= x\mu_i P h_{\mathcal{K},0} + p(P\mu_i e_x + \mathbf{w}_{\mathcal{K}}^\top E_i D(\mathbf{c}_x)) \pmod{QP}. \end{aligned}$$

Applying (3) gives the stated relation modulo  $Q$ . Indeed,  $\mathbf{z} - p\delta$  is divisible by  $P$  componentwise. After subtracting the message term and  $pP\mu_i e_x$ , the displayed relation shows that  $p(\mathbf{w}_{\mathcal{K}}^\top E_i D(\mathbf{c}_x) - \mathbf{h}_{\mathcal{K}}^\top \delta)$  is divisible by  $P$ . Since  $\gcd(p, P) = 1$ , the error numerator is also divisible by  $P$ .

### 3.3 Conversion-Free Party-Set Growth

The cofactor construction defines a common left relation for every party set of size at most  $K_{\max}$  in the fixed  $(K_{\max} + 1)$ -dimensional ciphertext space. On the unit event, this relation gives a decodable projective direction. Enlarging a party set changes the relation used to analyze and decrypt a ciphertext but does not change the ciphertext itself.

**Proposition 1 (Closure under party-set growth).** *Let  $\mathbf{c}$  be an intermediate vector ciphertext obtained from local cGSW inputs owned by parties in  $\mathcal{K}$  using extraction, vector addition, and RNS external products. For every  $\mathcal{T} \supseteq \mathcal{K}$  with  $|\mathcal{T}| \leq K_{\max}$ , the same evaluation trace establishes the corresponding relation for  $\mathbf{c}$  under  $\mathbf{h}_{\mathcal{T}}$ . Hence enlarging the party set requires neither ciphertext conversion nor evaluation keys.*

*Proof.* Fix  $\mathcal{T} \supseteq \mathcal{K}$ . Every leaf matrix  $C_i$  is owned by an  $i \in \mathcal{T}$ , so Lemma 3 gives its cGSW relation under  $\mathbf{h}_{\mathcal{T}}$ . Extraction is the external product with  $\mathbf{e}_0$ , vector addition preserves the relation, and Equation (4) preserves it for each subsequent external product. Induction over the evaluation trace proves the claim.

For an honestly generated fixed set  $\mathcal{T}$ , Theorem 1 bounds the probability that the required unit condition fails. The fixed ambient dimension and the noise budget must accommodate the final party set.

## 4 Compact-Input Technique

In Section 3, each fresh cGSW input is communicated to the evaluator as a full local cGSW matrix containing  $O(dn^2)$  ring polynomials. This communication cost grows quadratically with the system capacity. Our compact-input technique moves the cGSW structure into a reusable local hint and represents each fresh message by a compact GLEV ciphertext. After the hint has been published once, each fresh ciphertext contains only  $O(d)$  ring polynomials, with no explicit dimensional dependence on  $K_{\max}$ .

### 4.1 Compact Fresh-Input Communication

Let party  $i$  have local decryption map  $A_i = [I_{K_{\max}} \mid \mathbf{s}_i]$ . In addition to  $\mathbf{s}_i$ , it samples an independent scalar bridge key  $t_i$ . The party publishes a reusable local hint  $H_i \in \mathcal{R}_{QP}^{n \times dn}$  satisfying  $A_i H_i = t_i A_i G + p E_i^H \pmod{QP}$ . For  $a \in \mathcal{R}_Q$ , the hint satisfies

$$A_i H_i D(a I_n) = t_i P a A_i + p E_i^H D(a I_n).$$

For compact encryption, write  $\bar{\mathbf{g}} = (\bar{g}_0, \dots, \bar{g}_{d-1})$  with  $\bar{g}_\ell = Q_\ell \in \mathcal{R}_Q$ , so that  $g_\ell = P \bar{g}_\ell$  under the natural embedding into  $\mathcal{R}_{QP}$ .

For every evaluation-gadget level  $0 \leq \ell < d$ , a fresh encryption of  $\mu$  under  $t_i$  consists of  $(a_\ell, b_\ell) \in \mathcal{R}_Q^2$  satisfying

$$b_\ell + t_i a_\ell = \mu \bar{g}_\ell + p e_\ell \pmod{Q}.$$

The resulting compact GLEV ciphertext contains  $2d$  ring elements.

To lift it into the local cGSW domain, define

$$C_\mu^{(\ell)} = P b_\ell I_n + H_i D(a_\ell I_n) \pmod{QP}. \quad (7)$$

Here  $P b_\ell$  is well defined in  $\mathcal{R}_{QP}$  because changing an integer lift of  $b_\ell$  by a multiple of  $Q$  changes  $P b_\ell$  by a multiple of  $QP$ . The lifting algorithm outputs

$$\text{Lift}(H_i, \text{GLEV}_{t_i}^{\bar{\mathbf{g}}}(\mu)) = C_\mu = [C_\mu^{(0)} \mid \dots \mid C_\mu^{(d-1)}].$$

**Lemma 5 (Correctness of the lift).** *The matrix  $C_\mu$  is a local cGSW ciphertext of  $\mu$  relative to  $A_i$ . More precisely,*

$$\begin{aligned} A_i C_\mu^{(\ell)} &= \mu g_\ell A_i + p \widehat{E}_{i,\ell} \pmod{QP}, \\ \widehat{E}_{i,\ell} &= P e_\ell A_i + E_i^H D(a_\ell I_n), \end{aligned}$$

Consequently,

$$A_i C_\mu = \mu A_i G + p \widehat{E}_i \pmod{QP}. \quad (8)$$

*Proof.* Using the reusable-hint and compact-ciphertext relations,

$$\begin{aligned} A_i C_\mu^{(\ell)} &= P b_\ell A_i + t_i P a_\ell A_i + p E_i^H D(a_\ell I_n) \\ &= P(b_\ell + t_i a_\ell) A_i + p E_i^H D(a_\ell I_n) \\ &= \mu g_\ell A_i + p \widehat{E}_{i,\ell}. \end{aligned}$$

Concatenating the  $d$  block relations gives (8).

## 4.2 Fused External Product

Let  $\mathbf{c}_x$  be an existing vector ciphertext and write its decomposition as  $D(\mathbf{c}_x) = (\mathbf{v}_0, \dots, \mathbf{v}_{d-1})$ , where  $P\mathbf{c}_x = \sum_\ell g_\ell \mathbf{v}_\ell \pmod{QP}$ . Materializing the conceptual cGSW matrix and applying it to  $\mathbf{c}_x$  would give  $\sum_\ell C_\mu^{(\ell)} \mathbf{v}_\ell$ . Substituting the lifted blocks and exchanging the two finite sums gives

$$\sum_{\ell=0}^{d-1} C_\mu^{(\ell)} \mathbf{v}_\ell = P \sum_{\ell=0}^{d-1} b_\ell \mathbf{v}_\ell + H_i \left( \sum_{\ell=0}^{d-1} D(a_\ell I_n) \mathbf{v}_\ell \right).$$

Set  $\mathbf{z} = P \sum_\ell b_\ell \mathbf{v}_\ell$  and  $\mathbf{W} = \sum_\ell D(a_\ell I_n) \mathbf{v}_\ell$ , and define  $\text{FExt}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \mathbf{c}_x) = \text{ModDown}_P(\mathbf{z} + H_i \mathbf{W})$ .

**Lemma 6 (Correctness of FExt).** *For the conceptual lift  $C_\mu$  in (7),*

$$\text{FExt}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \mathbf{c}_x) = \text{ExtProd}(C_\mu, \mathbf{c}_x) \pmod{Q}. \quad (9)$$

*Proof.* Substituting (7) gives

$$\begin{aligned} C_\mu D(\mathbf{c}_x) &= \sum_\ell C_\mu^{(\ell)} \mathbf{v}_\ell \\ &= P \sum_\ell b_\ell \mathbf{v}_\ell + H_i \left( \sum_\ell D(a_\ell I_n) \mathbf{v}_\ell \right) \\ &= \mathbf{z} + H_i \mathbf{W}. \end{aligned}$$

Applying  $\text{ModDown}_P$  to both sides proves the claim.

**Proposition 2 (Evaluation completeness).** *For every supported vector accumulator circuit, lifting each compact input and applying the evaluation algorithm yields a valid evaluation. Any occurrence of  $\text{ExtProd}(\text{Lift}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu)), \mathbf{c})$  may be replaced by  $\text{FExt}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \mathbf{c})$ .*

*Proof.* The claim follows from Lemmas 5 and 6.

## 5 MKFHE Constructions

In this section, we first instantiate the cofactor compatibility technique of Section 3 with full local cGSW ciphertexts. We then combine it with the compact-input technique of Section 4 to reduce the communication cost of fresh encrypted inputs.

### 5.1 Levis-fast

We instantiate Levis-fast following the leveled MKFHE syntax of Definition 3 then analyze its correctness and noise growth.

**Setup.** On input  $(1^\lambda, 1^{K_{\max}}, 1^L, 1^M)$ , the setup algorithm selects ring and RNS gadget parameters  $(N, Q, P, p, (q_\ell)_{\ell < d})$  and distributions  $(\chi, \chi_r, \psi_{\text{pk}}, \psi_f)$  satisfying the security and correctness conditions developed below. It outputs these values together with  $(K_{\max}, L, M)$  as  $pp$ . The long-term secret distribution  $\chi$  is coefficientwise uniform over  $\{-1, 0, 1\}$ . The distribution  $\chi_r$  denotes the ephemeral distribution, and  $\psi_{\text{pk}}, \psi_f$  are centered error distributions.

**KeyGen.** Each party independently samples  $\mathbf{s}_i \leftarrow \chi^{K_{\max}}$ ,  $a_i \leftarrow U(\mathcal{R}_{QP})$ , and  $\mathbf{e}_i \leftarrow \psi_{\text{pk}}^{K_{\max}}$ . Its public key is  $\mathbf{p}_i = (-\mathbf{s}_i a_i + p \mathbf{e}_i, a_i) \in \mathcal{R}_{QP}^{K_{\max}+1}$ . It outputs  $(sk_i, ek_i, pk_i) = (\mathbf{s}_i, \emptyset, \mathbf{p}_i)$ .

**Enc.** On input  $(pp, pk_i, \mu)$  with  $\mu \in \mathcal{R}_p$ , independently sample  $r_\ell \leftarrow \chi_r$  and  $\boldsymbol{\eta}_\ell \leftarrow \psi_f^n$  for each  $\ell \in [dn]$ , set  $C[:, \ell] = \mathbf{p}_i r_\ell + p \boldsymbol{\eta}_\ell + \mu G[:, \ell] \pmod{QP}$ , and output  $(C, \{i\})$  with  $C \in \mathcal{R}_{QP}^{n \times dn}$ .

**Eval.** On input  $(pp, f, \{(pk_i, ek_i)\}_{i \in K}, \{(ct_j, \{i_j\})\}_{j \in [\ell]})$ , evaluate  $f$  gate by gate using vector ciphertexts over  $R_Q$ . The accumulator can be initialized as a vector ciphertext by  $\text{ExtProd}(C, e_0)$ . Vector addition is  $c_{x+y} = c_x + c_y$ . If  $C_y$  is a cGSW ciphertext encrypting  $y$ , multiplication is  $c_{xy} = \text{ExtProd}(C_y, c_x)$ . Each ciphertext is maintained together with a party label. A gate on  $(ct_u, K_u)$  and  $(ct_v, K_v)$  outputs a ciphertext labeled by  $K_u \cup K_v$ , provided that  $|K_u \cup K_v| \leq K_{\max}$ .

**Dec.** On input  $(pp, \{sk_i\}_{i \in \mathcal{K}}, (ct, \mathcal{K}))$ , take  $\mathbf{c} = \text{Extract}(C)$  when  $ct = C$  is a matrix, and set  $\mathbf{c} = ct$  otherwise. Compute  $\mathbf{h}_{\mathcal{K}}$  from the parties' secret keys and return  $\perp$  if  $h_{\mathcal{K},0}$  is not a unit in  $\mathcal{R}_p$ . Otherwise, output  $\text{Dec}(\mathbf{h}_{\mathcal{K}}, \mathbf{c}) = (h_{\mathcal{K},0} \bmod p)^{-1} ([\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}]_Q \bmod p)$ .

**Correctness Analysis** The correctness analysis focuses on the decodability of the cofactor direction and the associated error growth. We first establish the decoding condition, then bound the probability that  $h_{\mathcal{K},0} \bmod p$  is not a unit, and finally propagate the cofactor-weighted error through evaluation. When the party set is fixed in the following analysis, write  $\mathcal{K} = (i_1, \dots, i_k)$  and abbreviate  $\mathbf{h} = \mathbf{h}_{\mathcal{K}}$  and  $\mathbf{w} = \mathbf{w}_{\mathcal{K}}$ .

**Lemma 7 (No-wrap decoding).** *Suppose  $\mathbf{h}^\top \mathbf{c} = \mu h_0 + pe \pmod{Q}$ ,  $h_0 \pmod{p}$  is a unit, and  $\|\mu h_0 + pe\|_\infty < Q/2$ . Then Dec returns  $\mu \pmod{p}$ .*

*Proof.* The centered lift equals the actual polynomial  $\mu h_0 + pe$  under the stated bound. Reduction modulo  $p$  removes  $pe$ . Multiplying by the plaintext-ring inverse of  $h_0$  recovers the message.

*Invertibility with ternary secrets.* Let  $p$  be an odd prime,  $f_p = \text{ord}_{2N}(p)$ , and  $g_p = N/f_p$ . The polynomial  $X^N + 1$  has  $g_p$  distinct irreducible factors over  $\mathbb{F}_p$ , each of degree  $f_p$ .

**Theorem 1 (Invertible decoding coordinate).** *Let  $p$  be an odd prime. For a fixed honestly sampled set of  $k$  parties whose secret coefficients are independent and uniform over  $\{-1, 0, 1\}$ , the cofactor construction satisfies*

$$\Pr[h_0 \notin \mathcal{R}_p^\times] \leq \varepsilon_{p,N,k} := g_p \sum_{j=1}^{k-1} 3^{-f_p j} \leq \frac{g_p}{3^{f_p} - 1}. \quad (10)$$

*For every fixed odd prime  $p$ , this probability is  $2^{-\Omega_p(N)}$  as the power-of-two dimension  $N$  grows.*

*Proof.* Since  $p \nmid 2N$ , the polynomial  $X^N + 1$  is square-free over  $\mathbb{F}_p$  and factors into  $g_p$  irreducible polynomials  $F_1, \dots, F_{g_p}$  of degree  $f_p$ . Let  $\alpha_\ell$  be a root of  $F_\ell$ . The Chinese remainder theorem gives  $\mathcal{R}_p \cong \prod_{\ell=1}^{g_p} \mathbb{F}_{p^{f_p}}$ ,  $h_0 \in \mathcal{R}_p^\times \iff h_0(\alpha_\ell) \neq 0, \forall \ell$ . It therefore suffices to bound  $\Pr[h_0(\alpha_\ell) = 0]$  for one factor.

Fix a root  $\alpha = \alpha_\ell$  and write  $s(X) = \sum_{r=0}^{N-1} c_r X^r$ . After conditioning on  $c_{f_p}, \dots, c_{N-1}$ , their contribution is a fixed element  $\gamma \in \mathbb{F}_{p^{f_p}}$ , and  $s(\alpha) = \gamma + \sum_{r=0}^{f_p-1} c_r \alpha^r$ . The elements  $1, \alpha, \dots, \alpha^{f_p-1}$  are linearly independent over  $\mathbb{F}_p$ . Since  $p$  is odd, the map from  $(c_0, \dots, c_{f_p-1}) \in \{-1, 0, 1\}^{f_p}$  to the displayed sum is injective. Consequently,  $\max_{x \in \mathbb{F}_{p^{f_p}}} \Pr[s(\alpha) = x] \leq \pi$ ,  $\pi := 3^{-f_p}$ .

The claim is immediate for  $k = 1$ , when  $h_0 = -1$ . Assume  $k \geq 2$ . Subtracting the first row of the minor defining  $h_0$  from every remaining row and expanding along the last column gives, up to sign,  $h_0(\alpha) = \det(D_\alpha)$ ,  $D_\alpha = (s_{i_r, j}(\alpha) - s_{i_1, j}(\alpha))_{2 \leq r \leq k, 1 \leq j \leq k-1}$ . Condition on the first party's secret. The entries of  $D_\alpha$  are independent, and each has maximum point probability at most  $\pi$ .

Set  $m = k - 1$  and reveal the rows of  $D_\alpha$  in order. A random vector with independent coordinates of maximum point probability at most  $\pi$  lies in a fixed codimension- $c$  subspace with probability at most  $\pi^c$ . Applying this bound at the first dependent row gives

$$\Pr[D_\alpha \text{ is singular}] \leq \pi^m + \pi^{m-1} + \dots + \pi = \sum_{j=1}^{k-1} \pi^j.$$

A union bound over the  $g_p$  factor fields proves (10).

For the asymptotic claim, write  $N = 2^a$  and let  $t_p = \max\{v_2(p-1), v_2(p+1)\}$ . For fixed  $p$  and  $a \geq t_p$ , the standard order-lifting identity gives  $f_p = 2^{a+1-t_p}$  and  $g_p = 2^{t_p-1}$ . Hence  $f_p = \Theta_p(N)$  while  $g_p$  is constant, and the failure probability is  $2^{-\Omega_p(N)}$ .

*Cofactor magnitude and error propagation.* For chosen integer lifts of the inputs, let  $m_w \in \mathcal{R}$  denote the value at wire  $w$  obtained without intermediate reduction modulo  $p$ , and define  $M_f = \max\{1, \max_{w \in f} \|m_w\|_1\}$ . Suppose each relevant secret component satisfies  $\|s_{i,j}\|_1 \leq H$ . For a uniform bound over all party sets of size at most  $K_{\max}$ , subtracting a baseline row in each minor and using submultiplicativity gives

$$L_w := \sum_{j < K_{\max}} \|h_j\|_1 \leq K_{\max}!(2H)^{K_{\max}-1}, \quad (11)$$

$$\|h_0\|_{\infty} \leq (K_{\max} - 1)!(2H)^{K_{\max}-1}, \quad (12)$$

$$L_h := \sum_{j \leq K_{\max}} \|h_j\|_1 \leq (1 + H)L_w. \quad (13)$$

For (13), use  $h_{K_{\max}} = \sum_{j < K_{\max}} s_{i,j} h_j$ . The bounds hold for all keys satisfying the norm restrictions. For uniform ternary secrets,  $H \leq N$ . The cofactors increase coefficient norms. They remain elements of the same degree- $(N - 1)$  ring. For public encryption, if  $\|r\|_1 \leq H_r$ ,  $\|e_i\|_{\infty} \leq B_{\text{pk}}$ , and  $\|\eta\|_{\infty} \leq B_f$ , then

$$\beta_0 \leq L_w B_{\text{pk}} H_r + L_h B_f. \quad (14)$$

These are bounds on the base error, before multiplying by  $p$ . Let  $\Gamma_{\text{RNS}} = dnNB_g$ . For an RNS external product whose matrix operand has common-direction error at most  $\beta_C$ , define

$$\Xi(\beta_C) = \left\lceil \frac{\Gamma_{\text{RNS}} \beta_C + L_h B_{\delta}}{P} \right\rceil. \quad (15)$$

Indeed, if the matrix encrypts  $y$  and the vector ciphertext has error  $e_x$ , Lemma 4 gives  $\beta_{xy} \leq \|y\|_1 \beta_x + \Xi(\beta_C)$ . Vector addition satisfies  $\beta_{x+y} \leq \beta_x + \beta_y$ . For the *Levis-fast* set  $\beta_{\text{ext}} = \Xi(\beta_0)$  and  $\Lambda_{\text{RNS}} = \max\{2, M + 1\}$ . Induction over a supported depth- $L$  vector-accumulator circuit gives  $\beta_L \leq \Lambda_{\text{RNS}}^L \beta_{\text{ext}}$ . No independence between accumulated errors, RNS digits, and modulus-down corrections is assumed.

**Theorem 2 (Fixed-capacity leveled correctness).** *Fix a set of at most  $K_{\max}$  honestly generated parties and a supported vector-accumulator circuit  $f$  over  $\mathcal{R}_p$  of depth at most  $L$  whose lifted wire values satisfy  $M_f \leq M$ . Assume the conditions of Theorem 1 and that all sampled error and ephemeral-secret bounds above hold except with total probability  $\delta_E$ . If*

$$Q > 2(M \|h_0\|_{\infty} + p \Lambda_{\text{RNS}}^L \beta_{\text{ext}}), \quad (16)$$

*using the deterministic upper bounds (12) and (14), then the decoded result is correct except with probability at most  $\delta_E + \varepsilon_{p,N,K_{\max}}$ .*

*Proof.* Lemma 2 supplies the shared input relation. Induction using the RNS external-product recurrence and vector addition gives the output error bound. The definition of  $M_f$  and the assumption  $M_f \leq M$  give  $\|m_{\text{out}} h_0\|_{\infty} \leq M \|h_0\|_{\infty}$ , so (16) implies Lemma 7's no-wrap condition. Union the error-bound failure event with Theorem 1's bad-key event.

For subgaussian coefficient errors, a union bound supplies the values  $B_{\text{pk}}$  and  $B_{\text{f}}$  required by the theorem:  $T$  coefficients with parameter  $\tau$  have magnitude at most  $\tau\sqrt{2\ln(2T/\delta_E)}$  except with probability  $\delta_E$ . A reused public-key error is counted once. Parameters must satisfy (16), the RNS decomposition bounds, and the RLWE hardness requirements simultaneously. At fixed  $(K_{\text{max}}, L)$ , ciphertext dimensions are independent of circuit size, while the leveled parameters may depend on  $L$ .

**Theorem 3 (Asymptotic parameter feasibility).** *Let  $K_{\text{max}}(\lambda)$  and  $L(\lambda)$  be polynomially bounded. Let  $p$  and  $N$  be admissible polynomially bounded parameter families with  $p \nmid 2N$  and  $\varepsilon_{p,N,K_{\text{max}}}$  negligible in  $\lambda$ . Assume that the execution samples polynomially many coefficients whose high-probability bounds are polynomial in  $(\lambda, K_{\text{max}}, L, N)$  and that  $M$  is polynomially bounded. Assume admissible RNS families for which  $d$ ,  $\log B_g$ ,  $\log B_\delta$ , and  $\log P$  are polynomially bounded. Then there exist moduli  $Q$  coprime to  $pP$  for which Theorem 2 has negligible failure probability and every public parameter has polynomial bit length. If the resulting RLWE instances are pseudorandom, the same parameter family also satisfies Theorem 4.*

*Proof.* The assumed bound on  $\varepsilon_{p,N,K_{\text{max}}}$  makes the nonunit event negligible. Since  $H = O(N)$ , Equations (11)–(14) give

$$\log_2 \beta_0 = O(K_{\text{max}} \log K_{\text{max}} + K_{\text{max}} \log N + \log(B_{\text{pk}} H_r + N B_{\text{f}})). \quad (17)$$

The assumed RNS bounds make  $\log \Gamma_{\text{RNS}}$ ,  $\log \beta_{\text{ext}}$ , and  $\log A_{\text{RNS}}$  polynomial in  $(\lambda, K_{\text{max}}, L, N)$ . Consequently, the bit length required by (16) is polynomial. Choose an RNS modulus  $Q$  of sufficient bit length and coprime to  $pP$ . Thus the complete parameter description is polynomial in  $(\lambda, K_{\text{max}}, L)$ . The final security statement is conditional on the stated computational assumptions and does not constitute a concrete hardness estimate.

**Security of Levis-fast** The cofactor vector is computed from secret keys only for decryption and never appears in the public material of **Levis-fast**. We formalize the static-coalition game and give the corresponding hybrid argument.

*Static-coalition security game.* We use indistinguishability under chosen-plaintext attack (IND-CPA) with a static coalition. Let  $U$  users be registered. Before their keys are sampled, the adversary fixes a corrupt set. Honest keys use independent randomness. Corrupt users' honestly generated secrets are given to the adversary. A hidden bit  $b$  chooses the left or right message in each challenge query  $(i, \mu_0, \mu_1)$  for an honest user. The challenger returns the corresponding matrix ciphertext produced by **Enc**. Queries may be adaptive and there may be polynomially many. Their messages are ordinary adversary-selected elements of  $\mathcal{R}_p$ , not challenger-supplied functions of unknown secret keys. Public computation of all challenge ciphertexts, including extraction, is unrestricted. The advantage is the absolute difference between the probabilities that the adversary outputs

one in the two worlds. The game protects inputs against the evaluator and a static colluding coalition. Collaborative-decryption transcripts and outputs are outside this experiment.

**Theorem 4 (Public-key multi-user IND-CPA).** *Suppose the two RLWE distributions in Section 2 are pseudorandom. Let  $U_h$  be the number of honest users, and let  $J$  be the total number of challenge ciphertext columns. With a conservative convention for distinguishing advantage,*

$$\text{Adv}_{\text{CPA}} \leq 2U_h K_{\max} \varepsilon_{\text{pk}} + 2U_h J \varepsilon_{\text{enc}}, \quad (18)$$

where  $\varepsilon_{\text{pk}}$  bounds a one-sample RLWE distinguisher modulo  $QP$  for  $(\chi, \psi_{\text{pk}})$  and  $\varepsilon_{\text{enc}}$  bounds an  $n$ -sample distinguisher modulo  $QP$  for  $(\chi_r, \psi_f)$ . Reductions have polynomial simulation overhead. The same bound applies after any polynomial-time public evaluation.

*Proof.* For either message world, replace each honest public key by a uniform vector in  $\mathcal{R}_{QP}^n$ . Lemma 1 performs this replacement response by response at cost at most  $U_h K_{\max} \varepsilon_{\text{pk}}$ . Next replace the pre-message term  $\mathbf{p}r + \mathbf{p}\boldsymbol{\eta}$  in every challenge column by a uniform vector. For uniform  $\mathbf{p}$ , its coordinates form RLWE samples sharing the ephemeral secret  $r$ . Sequential replacement of the columns, including a guess of an adaptively selected party and column, costs at most  $U_h J \varepsilon_{\text{enc}}$ . The shifts  $\mu G[:, \ell]$  preserve uniformity. Applying the hybrids to both message worlds and using the triangle inequality proves (18). Public evaluation is efficient postprocessing.

The invertibility of  $h_0$  is a correctness condition and does not affect key generation. Concrete security parameters must account for both the underlying RLWE hardness and the reduction losses in (18) [3].

## 5.2 Levis-small

Levis-small extends Levis-fast with the party-side CEnc syntax of Definition 3 and the technique of Section 4.

*Modified Setup.* Retain the RNS gadget  $G = G_n$ , its decomposition  $D$ , and its  $d$  limbs, where  $n = K_{\max} + 1$ . Extend  $pp$  with  $(\chi_t, \psi_{\text{in}})$ .

*Modified KeyGen.* In addition to the base key pair, party  $i$  samples  $t_i \leftarrow \chi_t$  independently. For every column  $j \in [dn]$ , it samples  $r_j^H \leftarrow \chi_r$  and  $\boldsymbol{\eta}_j^H \leftarrow \psi_f^n$  and sets

$$H_i[:, j] = \mathbf{p}_i r_j^H + \mathbf{p} \boldsymbol{\eta}_j^H + t_i G[:, j] \pmod{QP}.$$

Set  $H_i \in \mathcal{R}_{QP}^{n \times dn}$  and output  $(sk_i, ek_i, pk_i) = ((\mathbf{s}_i, t_i), H_i, \mathbf{p}_i)$ .

*CEnc.* On input  $(pp, sk_i, \mu)$  with  $\mu \in \mathcal{R}_p$ , independently sample  $a_\ell \leftarrow U(\mathcal{R}_Q)$  and  $e_\ell \leftarrow \psi_{\text{in}}$  for  $0 \leq \ell < d$ , set  $b_\ell = -t_i a_\ell + \mu \bar{g}_\ell + p e_\ell \pmod{Q}$ , and output  $(\text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \{i\})$ , where  $\text{GLEV}_{t_i}^{\mathbf{g}}(\mu) = ((a_\ell, b_\ell))_{\ell=0}^{d-1} \in (\mathcal{R}_Q^2)^d$ .

*Modified Eval.* On input  $(\text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \{i\})$  and  $ek_i = H_i$ , the evaluator may output

$$(\text{Lift}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu)), \{i\})$$

and apply *Levis-fast* algorithm. Given a vector ciphertext  $(\mathbf{c}_x, \mathcal{K}_x)$ , it may instead output

$$(\text{FExt}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \mathbf{c}_x), \mathcal{K}_x \cup \{i\}),$$

as defined in Section 4, provided that  $|\mathcal{K}_x \cup \{i\}| \leq K_{\max}$ . Subsequent gates use the *Levis-fast* evaluation algorithm and its party-label union rule.

*Dec.* Run the *Levis-fast* decryption algorithm using the  $\mathbf{s}_i$  component of each  $sk_i = (\mathbf{s}_i, t_i)$ .

**Correctness** Writing  $\boldsymbol{\eta}_j^H = (\boldsymbol{\eta}_{\text{head},j}^H, \boldsymbol{\eta}_{\text{last},j}^H)$ , define  $E_i^H[:, j] = \mathbf{e}_i r_j^H + \boldsymbol{\eta}_{\text{head},j}^H + \mathbf{s}_i \boldsymbol{\eta}_{\text{last},j}^H$ . Then  $A_i H_i = t_i A_i G + p E_i^H \pmod{QP}$ . For any current party set  $\mathcal{K}$  and  $i \in \mathcal{K}$ , Lemma 2 gives  $\mathbf{h}_{\mathcal{K}}^\top = \mathbf{w}_{\mathcal{K}}^\top A_i$ . Applying this identity to the reusable hint gives  $\mathbf{h}_{\mathcal{K}}^\top H_i = t_i \mathbf{h}_{\mathcal{K}}^\top G + p \mathbf{w}_{\mathcal{K}}^\top E_i^H$ . Thus  $H_i$  and every compact ciphertext of party  $i$  remain valid for every future admissible superset of that party. Neither object is modified when the party set grows. Let  $(\mathbf{c}_x, \mathcal{K}_x)$  be an existing vector ciphertext and set  $\mathcal{K} = \mathcal{K}_x \cup \{i\}$ . Let  $\boldsymbol{\epsilon}_\mu = \mathbf{w}_{\mathcal{K}}^\top \widehat{E}_i$ . If  $\mathbf{h}_{\mathcal{K}}^\top \mathbf{c}_x = x h_{\mathcal{K},0} + p e_x$ , let  $\mathbf{z}$  be the fused expression before  $\text{ModDown}_P$  and let  $p\boldsymbol{\delta} \equiv \mathbf{z} \pmod{P}$  be its correction. Lemmas 5 and 6 give

$$\mathbf{h}_{\mathcal{K}}^\top \text{FExt}(H_i, \text{GLEV}_{t_i}^{\mathbf{g}}(\mu), \mathbf{c}_x) = x \mu h_{\mathcal{K},0} + p \left( \mu e_x + \frac{\boldsymbol{\epsilon}_\mu \text{D}(\mathbf{c}_x) - \mathbf{h}_{\mathcal{K}}^\top \boldsymbol{\delta}}{P} \right) \pmod{Q}. \quad (19)$$

The fused computation is identical to the explicit lift followed by an external product. Suppose  $\|r_j^H\|_1 \leq H_r$ ,  $\|\mathbf{e}_i\|_\infty \leq B_{\text{pk}}$ ,  $\|\boldsymbol{\eta}_j^H\|_\infty \leq B_{\text{f}}$ ,  $\|e_\ell\|_\infty \leq B_{\text{in}}$ , and the RNS digits are bounded by  $B_g$ . Then

$$\begin{aligned} \|\mathbf{w}_{\mathcal{K}}^\top E_i^H\|_\infty &\leq \beta_H := L_w B_{\text{pk}} H_r + L_h B_{\text{f}}, \\ \|\mathbf{w}_{\mathcal{K}}^\top \widehat{E}_{i,\ell}\|_\infty &\leq \beta_{\text{lift}} := P B_{\text{in}} H_h + d N B_g \beta_H, \end{aligned} \quad (20)$$

where  $H_h = \max_{0 \leq j \leq K_{\max}} \|h_j\|_1 \leq \max\{1, H\} L_w$ .

**Theorem 5 (Correctness of Levis-small).** *Fix an honestly generated party set of size at most  $K_{\max}$  and a supported vector-accumulator circuit of depth at most  $L$  whose lifted wire values have  $\ell_1$  norm at most  $M$ . The evaluator may lift a compact input before an RNS external product or apply it directly to a vector ciphertext with  $\text{FExt}$ . Let  $\beta_{\text{lift}}$  be defined by (20), set  $\beta_\star = \max\{\beta_0, \beta_{\text{lift}}\}$ ,  $\beta_{\text{ext}}^\star = \Xi(\beta_\star)$ , and  $\Lambda_{\text{RNS}} = \max\{2, M + 1\}$ . If all sampling bounds hold except with probability  $\delta_{\text{cmp}}$  and*

$$Q > 2(M \|h_0\|_\infty + p \Lambda_{\text{RNS}}^L \beta_{\text{ext}}^\star), \quad (21)$$

*then Levis-small decrypts correctly except with probability at most  $\delta_{\text{cmp}} + \varepsilon_{p,N,K_{\max}}$ .*

*Proof.* Lemma 5 makes every lifted input a valid local cGSW ciphertext. Equations (19), (20), and (15) give  $\beta_{x\mu} \leq \|\mu\|_1 \beta_x + \Xi(\beta_{\text{lift}})$  for a compact gate. Since every lifted wire value has norm at most  $M$ , induction over the circuit gives  $\beta_{\text{out}} \leq A_{\text{RNS}}^L \beta_{\text{ext}}^*$ . Equation (21) implies the no-wrap condition of Lemma 7. Combining the sampling failure with the nonunit event proves the claim.

**Security** The public material now contains objects from two independent secret domains:  $H_i \in \mathcal{R}_{QP}^{n \times dn}$  is a target-domain encryption of  $t_i$  under  $A_i$ , whereas the compact pairs lie in  $\mathcal{R}_Q^2$  and are RLWE samples under  $t_i$ . We assume circular security for the following joint distribution. For independent  $\mathbf{s}_i \leftarrow \chi^{K_{\max}}$  and  $t_i \leftarrow \chi_t$ , the joint distribution of  $pk_i$ ,  $H_i$ , and polynomially many compact inputs under  $t_i$  is computationally indistinguishable when  $H_i$  is replaced by a uniform matrix of the same dimensions.

**Theorem 6 (Static privacy of compact inputs).** *Assume the RLWE conditions of Theorem 4, circular security for the joint distribution described above, and that multi-sample RLWE( $N, Q, \psi_{\text{in}}, \chi_t$ ) is hard. Then the compact-input construction is IND-CPA secure in the static-coalition game for honestly generated keys and hints and a polynomial number of compact-input queries.*

*Proof.* For each honest party, apply circular security to replace  $H_i$  by a uniform matrix while retaining the compact inputs under  $t_i$ . Next replace all pairs  $(b_\ell, a_\ell)$  by uniform auxiliary-key RLWE pairs. The shifts  $\mu \bar{g}_\ell$  preserve uniformity. Repeating over honest parties and applying the triangle inequality proves the claim. Reuse of  $H_i$  is covered by replacing its columns once, and reuse of  $t_i$  is covered by multi-sample auxiliary RLWE.

**Communication and Arithmetic Cost** For one party supplying  $m_{\text{in}}$  fresh inputs, Levis-fast sends  $m_{\text{in}} dn^2$  ring elements. Levis-small sends one  $dn^2$ -element hint over  $\mathcal{R}_{QP}$  and  $2d$  elements over  $\mathcal{R}_Q$  for each fresh ciphertext; compact ciphertexts contain no  $P$  limb. After the party publishes its reusable hint, the communication required for each subsequent fresh ciphertext drops from  $dn^2 = \Theta(dK_{\max}^2)$  to  $2d = \Theta(d)$ . The exact amortization threshold depends on the byte sizes of  $Q$  and  $QP$ .

For decryption, the evaluator sends the extracted vector ciphertext, which contains  $K_{\max} + 1$  elements of  $\mathcal{R}_Q$ . A fresh input that will serve as a matrix operand is retained as a full cGSW matrix of  $d(K_{\max} + 1)^2$  elements of  $\mathcal{R}_{QP}$ .

Ignoring additions, multiplication by  $P$ , digit extraction, and modulus down, an RNS external product between a full  $n \times dn$  cGSW matrix and a decomposed vector uses  $dn^2$  ring multiplications. Computing  $\mathbf{z}$ , the aggregated decomposition vector  $\mathbf{W}$ , and  $H_i \mathbf{W}$  uses respectively  $dn$ ,  $d^2 n$ , and  $dn^2$  ring multiplications. The fused count is therefore  $d(d+1)n + dn^2$ ; its benefit is avoiding materialization of the lifted matrix, not reducing this multiplication count below a full external product.

### 5.3 Discussion

We first discuss an additional consequence of cofactor compatibility: evaluation is independent of the multiplication order. We then briefly describe how bootstrapping and distributed decryption can be incorporated into our construction. The latter two components are not the focus of this work and are included only for completeness.

*Order-independent evaluation.* In MPS [25], the joint key depends on the multiplication order. Consequently, ciphertexts produced under different multiplication orders may be associated with different joint keys and require key switching before they can be combined in subsequent computations. In our construction, the cofactor relation is independent of the ordering of the participating parties. Permuting the rows used in the cofactor construction changes the signed-minor vector only by an overall sign, which represents the same projective decryption direction. Therefore, ciphertexts produced under different evaluation orders remain compatible, and no key switching is required to reconcile different multiplication orders.

*Bootstrapping.* As in MPS [25], *Levis-fast* can be extended to general bounded-depth circuit evaluation by retaining intermediate ciphertexts in the cGSW matrix form and applying the standard GSW-style internal product. Cofactor compatibility is preserved under this operation, so the product of two compatible matrix ciphertexts remains a compatible matrix ciphertext, without first converting either operand to a joint-key RGSW representation. This provides the multiplicative closure needed for multi-hop evaluation. Full homomorphism can then be obtained by bootstrapping under an appropriate circular-security assumption. In particular, each party publishes local cGSW encryptions of its secret-key components, and the evaluator homomorphically evaluates the decryption circuit, including the computation of the cofactor vector, the decryption phase, and the final normalization. *Levis-small* can use the same procedure after lifting its compact ciphertexts to the local cGSW form.

*Distributed decryption.* As in MPS [25], the participating parties can perform distributed decryption using a generic secure multiparty computation (MPC) protocol. Each party provides its local secret key as a private input, and the parties jointly evaluate the cofactor construction and the decryption circuit, revealing only the decoded plaintext. As in MPS, the resulting plain-model MKFHE can also be combined with the generic transformation of Ananth et al. [4] to obtain one-round decryption.

## 6 Evaluation

### 6.1 Experimental Setup

We provide a C++ proof-of-concept implementation of *Levis-fast* and *Levis-small*. We use the Intel Homomorphic Encryption Acceleration Library (HEXL) [7] to

**Table 2.** Parameter sets targeting at least 128-bit classical security. Each entry gives  $(N, \log_2 Q, \sigma)$ .

| $K_{\max}$ | MPS [25]         | Levis-fast      | Levis-small     |
|------------|------------------|-----------------|-----------------|
| 2          | (4096, 100, 3.2) | (8192, 96, 4)   | (16384, 144, 4) |
| 4          | (8192, 214, 3.2) | (16384, 135, 4) | (16384, 208, 4) |
| 6          | –                | (16384, 168, 4) | (16384, 235, 4) |
| 8          | (8192, 214, 3.2) | (16384, 205, 4) | (16384, 270, 4) |

accelerate number-theoretic transform (NTT) operations. All experiments are conducted on a machine equipped with an Intel Xeon Platinum 8358 central processing unit (CPU) at 2.60 GHz and 2 TB of random-access memory (RAM), running Ubuntu 22.04. For timing measurements, we execute each protocol 50 times and report the average runtime. All experiments are performed using *a single thread*. All parameter sets provide at least 128 bits of computational security.

## 6.2 Parameter Selection and Analysis

For MPS [25], we use the reported parameter sets in their paper. For Levis-fast and Levis-small, we select the parameters using the LWE Estimator [3] under the same security target. Table 2 lists the parameter sets used for the comparison. For Levis-fast and Levis-small, the 55-bit special modulus  $P$  is used only for arithmetic over  $QP$  and modulus down and is not included in  $\log_2 Q$ . As in MPS [25], we choose parameters for supported circuits of depth  $L = 2$ .

*Plaintext-Ring Invertibility* As in MPS, we select  $p = 257$ . For  $N = 2^a \geq 128$ , the identity  $v_2(257^{2^t} - 1) = 8 + t$  gives  $\text{ord}_{2N}(257) = f = N/128$ . Thus  $X^N + 1$  has 128 irreducible factors of degree  $f$  over  $\mathbb{F}_{257}$ . For a root  $\alpha$  of one factor, let  $\zeta = \alpha^f$ . Then  $\zeta$  generates  $\mathbb{F}_{257}^\times$ . A ternary secret satisfies  $s(\alpha) = \sum_{j=0}^{f-1} \alpha^j S_j$ , where the independent variables  $S_j = \sum_{b=0}^{127} c_{j+bf} \zeta^b$  use disjoint secret coefficients. Let  $\omega = e^{2\pi i/257}$  and  $F(u) = 1 + \omega^u + \omega^{-u}$ . For  $t \neq 0$ , the Fourier coefficient of  $S_j$  is

$$\widehat{P}(t) = 3^{-128} \prod_{b=0}^{127} F(t\zeta^b) = -3^{-128}.$$

Indeed,  $\zeta^{128} = -1$ , so the product selects one element from each pair  $\{u, -u\}$ . Moreover,  $F(u) = \omega^{-u}(1 - \omega^{3u})/(1 - \omega^u)$ , and multiplication by 3 permutes  $\mathbb{F}_{257}^\times$ . The squared product is one, and its sign is negative because it contains one representative from each of the 43 pairs on which  $F$  is negative. Fourier inversion gives

$$\Pr[S_j = x] = \begin{cases} (1 - 256 \cdot 3^{-128})/257, & x = 0, \\ (1 + 3^{-128})/257, & x \neq 0. \end{cases}$$

**Table 3.** Comparison of communication and runtime for MPS, **Levis-fast**, and **Levis-small** for the 128-bit security level. **Levis-fast** denotes the construction with full local cGSW inputs. **Levis-small** denotes its compact-input variant with reusable hints and compact online inputs. All rows use full occupancy, so  $k = K_{\max}$ . Communication is reported in mebibytes (MiB), and runtime is reported in milliseconds (ms).

| Scheme      | $k$ | Communication (MiB) |          |         | Runtime (ms) |          |          |           |
|-------------|-----|---------------------|----------|---------|--------------|----------|----------|-----------|
|             |     | Setup               | Join     | Post    | Setup        | Enc.     | Join     | Post-EP   |
| MPS         | 2   | 7.025               | 4.151    | 2.076   | 237.646      | 65.773   | 12.652   | 212.046   |
| MPS         | 4   | 178.467             | 56.689   | 14.172  | 3667.193     | 236.183  | 76.354   | 1175.027  |
| MPS         | 8   | 2202.188            | 359.438  | 44.930  | 52402.044    | 993.193  | 463.975  | 12885.840 |
| Levis-fast  | 2   | 0.000               | 5.309    | 2.654   | 10.289       | 28.688   | 6.314    | 3.130     |
| Levis-fast  | 4   | 0.000               | 111.328  | 27.832  | 70.608       | 236.694  | 82.838   | 22.259    |
| Levis-fast  | 6   | 0.000               | 512.203  | 85.367  | 161.730      | 645.139  | 305.027  | 57.936    |
| Levis-fast  | 8   | 0.000               | 1316.250 | 164.531 | 289.178      | 1078.499 | 645.305  | 92.424    |
| Levis-small | 2   | 20.988              | 1.688    | 0.844   | 192.967      | 10.703   | 38.174   | 18.033    |
| Levis-small | 4   | 205.469             | 6.500    | 1.625   | 1527.596     | 17.080   | 251.612  | 59.711    |
| Levis-small | 6   | 832.617             | 13.770   | 2.295   | 5350.210     | 23.482   | 689.853  | 114.089   |
| Levis-small | 8   | 2467.969            | 25.313   | 3.164   | 15581.823    | 31.065   | 2011.792 | 236.810   |

Consequently, with  $\pi_N = ((1 + 3^{-128})/257)^{N/128}$ , Theorem 1’s rank argument gives

$$\Pr[h_{\mathcal{K},0} \notin \mathcal{R}_{257}^\times] \leq 128 \sum_{j=1}^{k-1} \pi_N^j \leq \frac{128\pi_N}{1 - \pi_N}. \quad (22)$$

For  $N = 4096, 8192, 16384$ , the negative base-two logarithms of these bounds are at least 249.18, 505.36, and 1017.72, respectively. With eight registered parties, a union bound over all 247 nontrivial party sets remains below  $2^{-241}$  at  $N = 4096$ .

### 6.3 Concrete Evaluation

Table 3 presents our concrete evaluation. Communication counts serialized public material and ciphertexts transmitted to the evaluator. Setup includes one-time public and evaluation keys, including the reusable hints. Join includes the ciphertexts used to form the current party-set computation. Post includes the fresh ciphertext used by the post-join external product. Final-result transmission and distributed-decryption traffic are outside this benchmark. Setup and Join are offline, one-time costs, so we compare the recurring costs using the Post and Post-EP columns. Post-EP denotes the post-join external-product stage. For MPS, it includes RGSW extension and the subsequent external product. The post-join cost depends on the position of the party supplying the fresh ciphertext in the multiplication order. An earlier party has a smaller local ciphertext but requires a longer RGSW-extension path to the current joint key, whereas a later party has a larger local ciphertext but requires less extension work. In particular, the

first party incurs the smallest communication cost and the largest post-join computation cost, while the last party exhibits the opposite behavior. We therefore report the average MPS post-join cost over all  $k$  party positions. For each position, we repeat the measurement 50 times and first average these runs; the values reported in Table 3 are then averaged over the  $k$  party positions. For *Levis-fast*, it contains only the external product. For *Levis-small*, it contains the fused lifting and external-product computation. For communication measurements, we use seeded serialization for uniformly random mask components. In *Levis-small*, the  $d$  mask polynomials  $a_\ell$  are regenerated from a short PRNG seed, so only the  $d$  body polynomials  $b_\ell$  are transmitted, reducing the concrete ciphertext size by approximately a factor of two. The same optimization gives only limited savings for MPS, since each shared random mask  $a$  is accompanied by multiple body components. Over the measurements  $k \in \{2, 4, 8\}$ , *Levis-small* reduces Post communication by 2.5–14.2 $\times$  and Post-EP time by 11.8–54.4 $\times$  relative to MPS. *Levis-fast* reduces Post-EP time by 52.8–139.4 $\times$ . Its Post communication is 1.3–3.7 $\times$  larger. These stage-level measurements use the same security target but different scheme-specific secret distributions, moduli, and gadget parameters. They are therefore not an identical-parameter or end-to-end comparison.

## Acknowledgments

The authors used ChatGPT(GPT-5.6 Sol) to assist with language polishing formatting, coding, and drafting parts of the proofs. All outputs were independently checked, revised where necessary, and fully verified by the authors.

## References

1. Afshar, A., Goyal, R.: Fully-succinct multi-key FHE and rate-1 simulatable threshold decryption from LWE. In: Advances in Cryptology – CRYPTO 2026, Part II. Lecture Notes in Computer Science, vol. 16801, pp. 591–623 (2026). [https://doi.org/10.1007/978-3-032-35374-0\\_19](https://doi.org/10.1007/978-3-032-35374-0_19), <https://eprint.iacr.org/2026/1683>
2. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched ntru assumptions: Cryptanalysis of some fhe and graded encoding schemes. In: Annual international cryptology conference. pp. 153–178. Springer (2016)
3. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015). <https://doi.org/10.1515/jmc-2015-0016>
4. Ananth, P., Jain, A., Jin, Z., Malavolta, G.: Multi-key fully-homomorphic encryption in the plain model. In: Theory of Cryptography – TCC 2020. Lecture Notes in Computer Science, vol. 12550, pp. 28–57 (2020). [https://doi.org/10.1007/978-3-030-64375-1\\_2](https://doi.org/10.1007/978-3-030-64375-1_2)
5. Bae, Y., Cheon, J.H., Hanrot, G., Park, J.H., Stehlé, D.: Plaintext-ciphertext matrix multiplication and fhe bootstrapping: fast and fused. In: Annual International Cryptology Conference. pp. 387–421. Springer (2024)
6. Bergerat, L., Bonte, C., Curtis, B.R., Orfila, J.B., Paillier, P., Tap, S.: Sharing the mask: Ttfe bootstrapping on packed messages. IACR Transactions on Cryptographic Hardware and Embedded Systems **2025**(4), 925–971 (2025)

7. Boemer, F., Kim, S., Seifu, G., de Souza, F.D.M., Gopal, V.: Intel HEXL: Accelerating homomorphic encryption with Intel AVX512-IFMA52. Cryptology ePrint Archive, Report 2021/420 (2021), <https://eprint.iacr.org/2021/420>
8. Brakerski, Z., Perlman, R.: Lattice-based fully dynamic multi-key FHE with short ciphertexts. In: Advances in Cryptology – CRYPTO 2016, Part I. Lecture Notes in Computer Science, vol. 9814, pp. 190–213 (2016). [https://doi.org/10.1007/978-3-662-53018-4\\_8](https://doi.org/10.1007/978-3-662-53018-4_8), <https://eprint.iacr.org/2016/339>
9. Chen, H., Chillotti, I., Song, Y.: Multi-key homomorphic encryption from TFHE. In: Advances in Cryptology – ASIACRYPT 2019, Part II. Lecture Notes in Computer Science, vol. 11922, pp. 446–472 (2019). [https://doi.org/10.1007/978-3-030-34621-8\\_16](https://doi.org/10.1007/978-3-030-34621-8_16), <https://eprint.iacr.org/2019/116>
10. Chen, H., Dai, W., Kim, M., Song, Y.: Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 395–412 (2019). <https://doi.org/10.1145/3319535.3363207>
11. Chen, L., Zhang, Z., Wang, X.: Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension. In: Theory of Cryptography – TCC 2017, Part II. Lecture Notes in Computer Science, vol. 10678, pp. 597–627 (2017). [https://doi.org/10.1007/978-3-319-70503-3\\_20](https://doi.org/10.1007/978-3-319-70503-3_20)
12. Cheon, J.H., Jeong, J., Lee, C.: An algorithm for ntru problems and cryptanalysis of the ggh multilinear map without a low-level encoding of zero. LMS Journal of Computation and Mathematics **19**(A), 255–266 (2016)
13. Clear, M., McGoldrick, C.: Multi-identity and multi-key leveled FHE from learning with errors. In: Advances in Cryptology – CRYPTO 2015, Part II. Lecture Notes in Computer Science, vol. 9216, pp. 630–656 (2015). [https://doi.org/10.1007/978-3-662-48000-7\\_31](https://doi.org/10.1007/978-3-662-48000-7_31)
14. Dai, X., Wu, W., Feng, Y.: Key lifting: Multi-key fully homomorphic encryption in plain model without noise flooding. Cryptology ePrint Archive, Paper 2022/055 (2022), <https://eprint.iacr.org/2022/055>
15. Daza, V., Morillo, P., Rovira, S.: Leveled multikey FHE with constant-size ciphertexts from RLWE. Cryptology ePrint Archive, Paper 2022/447 (2022), <https://eprint.iacr.org/2022/447>
16. Ducas, L., van Woerden, W.: Ntru fatigue: how stretched is overstretched? In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 3–32. Springer (2021)
17. Gentry, C., Szydlo, M.: Cryptanalysis of the revised ntru signature scheme. In: International conference on the theory and applications of cryptographic techniques. pp. 299–320. Springer (2002)
18. Kim, T., Kwak, H., Lee, D., Seo, J., Song, Y.: Asymptotically faster multi-key homomorphic encryption from homomorphic gadget decomposition. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 726–740 (2023), <https://doi.org/10.1145/3576915.3623176>
19. Kirchner, P., Fouque, P.A.: Revisiting lattice attacks on overstretched ntru parameters. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–26. Springer (2017)
20. Kwak, H., Min, S., Song, Y.: Towards practical multi-key TFHE: Parallelizable, key-compatible, quasi-linear complexity. In: Public-Key Cryptography – PKC 2024, Part IV. Lecture Notes in Computer Science, vol. 14604, pp. 354–385 (2024). [https://doi.org/10.1007/978-3-031-57728-4\\_12](https://doi.org/10.1007/978-3-031-57728-4_12)

21. Liao, J., Wang, H., Wu, J.: A multikey fully homomorphic encryption privacy protection protocol based on blockchain for edge computing system. *Concurrency and Computation: Practice and Experience* **35**(4), e7539 (2023). <https://doi.org/10.1002/cpe.7539>
22. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In: *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*. pp. 1219–1234 (2012). <https://doi.org/10.1145/2213977.2214086>
23. Luo, F., Wang, H., Al-Kuwari, S., Han, W.: Multi-key fully homomorphic encryption without CRS from RLWE. *Computer Standards & Interfaces* **86**, 103742 (2023). <https://doi.org/10.1016/j.csi.2023.103742>
24. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: *Advances in Cryptology – EUROCRYPT 2010. Lecture Notes in Computer Science*, vol. 6110, pp. 1–23 (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_1](https://doi.org/10.1007/978-3-642-13190-5_1)
25. Min, S., Park, J., Song, Y.: Multi-key fully homomorphic encryption with non-interactive setup in the plain model. In: *Advances in Cryptology – CRYPTO 2026, Part II. Lecture Notes in Computer Science*, vol. 16801, pp. 331–362 (2026). [https://doi.org/10.1007/978-3-032-35374-0\\_11](https://doi.org/10.1007/978-3-032-35374-0_11), <https://eprint.iacr.org/2026/322>
26. Mukherjee, P., Wichs, D.: Two round multiparty computation via multi-key FHE. In: *Advances in Cryptology – EUROCRYPT 2016, Part II. Lecture Notes in Computer Science*, vol. 9666, pp. 735–763 (2016). [https://doi.org/10.1007/978-3-662-49896-5\\_26](https://doi.org/10.1007/978-3-662-49896-5_26), <https://eprint.iacr.org/2015/345>
27. N. Bourbaki: *Algebra I: Chapters 1–3*. Springer Berlin, Heidelberg (1998), <https://link.springer.com/book/9783540642435>
28. Park, J., van Leeuwen, B., Zajonc, O.: FINALLY: A multi-key FHE scheme based on NTRU and LWE. *IACR Communications in Cryptology* **1**(3), 15 (2024). <https://doi.org/10.62056/aebn-4c2h>, <https://eprint.iacr.org/2024/1505>
29. Peikert, C., Shiehian, S.: Multi-key FHE from LWE, revisited. In: *Theory of Cryptography – TCC 2016-B, Part II. Lecture Notes in Computer Science*, vol. 9986, pp. 217–238 (2016). [https://doi.org/10.1007/978-3-662-53644-5\\_9](https://doi.org/10.1007/978-3-662-53644-5_9), <https://eprint.iacr.org/2016/196>
30. Wang, J., Huang, R., Tang, X.: Dynamic multi-key FHE without CRS from LWE. *Cybersecurity* **9**, 31 (2026). <https://doi.org/10.1186/s42400-025-00431-z>
31. Xiang, B., Min, S., Hwang, I., Wang, Z., He, H., Wei, Y., Yang, K., Zhang, J., Deng, Y., Yu, Y.: HERDS: Multi-key fully homomorphic encryption with sublinear bootstrapping. In: *Advances in Cryptology – EUROCRYPT 2026, Part IV. Lecture Notes in Computer Science*, vol. 16544, pp. 214–242 (2026). [https://doi.org/10.1007/978-3-032-25327-9\\_8](https://doi.org/10.1007/978-3-032-25327-9_8), <https://eprint.iacr.org/2025/1804>
32. Xiang, B., Zhang, J., Wang, K., Deng, Y., Feng, D.: NTRU-based bootstrapping for MK-FHEs without using overstretched parameters. In: *Advances in Cryptology – ASIACRYPT 2024, Part I. Lecture Notes in Computer Science*, vol. 15484, pp. 241–270 (2024). [https://doi.org/10.1007/978-981-96-0875-1\\_8](https://doi.org/10.1007/978-981-96-0875-1_8)