# Secure Multi-User Linearly-Separable Distributed Computing

Amir Masoud Jafarpisheh
am.jafarpisheh@ed.ac.uk

Ali Khalesi
ali.khalesi@ipsa.fr

Petros Elia
elia@eurecom.fr

***Abstract*—The multi-user linearly separable distributed computing framework enables parallel processing via a sparse matrix factorization $\mathbf{F} = \mathbf{DE}$, where $\mathbf{F}$ describes users' requests, a $\gamma$-sparse $\mathbf{E}$ defines task allocation across $N$ servers, and a $\delta$-sparse $\mathbf{D}$ specifies connectivity between $N$ servers and $K$ users as well as decoding. This approach achieves significant parallelization gains with low communication and computation costs by decomposing the users' request matrix into structured sparse components. However, its linear nature also raises data secrecy concerns.**

**We adopt an information-theoretic secrecy framework requiring that each user learns nothing more than its own requested function. Our main results provide (i) a necessary condition stating that for each user $k$ observing $\alpha_k$ server responses, the common randomness visible to that user must span a subspace of dimension greater than $\alpha_k - 1$, and (ii) a necessary and sufficient condition requiring that removing from $\mathbf{D}$ the columns corresponding to the servers observed by a user leaves a matrix of rank at least $K - 1$. Based on these conditions, we design a general, cost-preserving secrecy-enforcing transformation valid over both finite and real fields, obtained by appending to $\mathbf{E}$ a basis of $\mathrm{Null}(\mathbf{D})$ and carefully injecting shared randomness. This scheme preserves communication and computation costs, guarantees perfect information-theoretic secrecy over finite fields, and in the real case yields an explicit mutual-information bound that can be made arbitrarily small by increasing the variance of Gaussian common randomness.**

## I. Introduction

Distributed data analytics increasingly operate across administrative and trust boundaries, where multiple users are authorized to compute functions of shared datasets while the underlying data must remain confidential. Prominent examples arise in multi-tenant cloud services, federated learning, healthcare analytics, financial risk assessment, and smart-grid monitoring. In these systems, users request distinct linear functionals of the same data, yet any information beyond the authorized computation may lead to severe privacy breaches.

A subtle but fundamental challenge arises even in purely linear workflows. While linear functions are among the simplest classes of computations, the ability to recover *additional* unintended linear combinations can enable strong inference attacks through aggregation, side information, or repeated queries. This issue persists in both discrete computations over $\mathbf{GF}(q)$ and real-valued computations over $\mathbb{R}$, where privacy guarantees based on computational hardness may fail due to auxiliary information or numerical precision. These considerations motivate an information-theoretic treatment of secrecy that provides unconditional guarantees independent of computational assumptions.

***Linearly Separable Computation*:** We restrict attention to *linearly separable* distributed computing schemes, in which each user's requested function can be expressed as a linear combination of intermediate computations performed independently at the servers. Each user $k \in [K]$ requests a linear combination of the messages, $\langle \mathbf{f}_k, \mathbf{w} \rangle$, where $\mathbf{f}_k \in \mathbb{F}^L$ is the request vector of user $k$, and $\mathbf{w} = [W_1, \ldots, W_L]^\mathsf{T}$ denotes the message vector (subfunction outputs) computed across the various servers. For $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_K]^\mathsf{T} \in \mathbb{F}^{K \times L}$ denote the users' request matrix, linear separability requires that $\mathbf{F}$ admits a factorization $\mathbf{F} = \mathbf{DE}$, where $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_N]^\mathsf{T} \in \mathbb{F}^{N \times L}$ specifies the linear computations assigned to the servers and $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K]^\mathsf{T} \in \mathbb{F}^{K \times N}$ specifies the linear aggregation performed by the users. Server $n$ computes $A_n = \mathbf{e}_n^\mathsf{T}\mathbf{w}$, while user $k$ recovers its desired value as $\mathbf{d}_k^\mathsf{T}\mathbf{A} = \mathbf{f}_k^\mathsf{T}\mathbf{w}$, where $\mathbf{A} = [A_1, \ldots, A_N]^\mathsf{T}$. This algebraic structure encompasses a broad class of distributed inference and analytics tasks and enables a precise characterization of correctness, efficiency [1], and as we will see here, information-theoretic secrecy as well.

***Related Works*:** Distributed computing systems are constrained by limited computation and communication resources, leading to a fundamental communication–computation tradeoff studied in [2]–[5]. Coding techniques have been widely used to reduce communication load and mitigate stragglers in linear computation settings [6]–[10]. In particular, single- and multi-user linearly separable models have been extensively studied, yielding near-optimal communication and computation schemes under various task assignment structures [1], [11]–[16]. However, these works primarily focus on efficiency and do not address whether the decoding structure itself may lead to unintended information leakage.

Data secrecy has been studied in related areas such as private function retrieval and secure computation [17]–[20], secret sharing and its multi-user extensions [21]–[24], and private access control [25], [26]. More closely related is [27],

The work of Amir Masoud Jafarpisheh is supported by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding Guarantee under grant EP/Z536404/1, as part of the FOCAL project funded under Marie Skłodowska-Curie grant agreement No 101169042. This work was also supported by the Huawei France-funded Chair towards Future Wireless Networks, and by the French government under the France 2030 ANR program "PEPR Networks of the Future" (ref. ANR-22-PEFT-0010).
A. Jafarpisheh is with the Institute for Imaging, Data, and Communications (IDCOM), School of Engineering, University of Edinburgh, Edinburgh, UK. A. Khalesi is with the Institut Polytechnique des Sciences Avancées (IPSA) and LINCS Lab, Paris, France. P. Elia is with the Communication Systems Department, EURECOM, Sophia Antipolis, France.

which studies information-theoretic secrecy in linearly separable computation for single-user settings and specific data assignment structures. Despite these advances, an information-theoretic characterization of data secrecy based on the structure of the decoding process has not yet been fully established. This work addresses this gap in multi-user linearly separable computation by identifying structural conditions ensuring that each user learns only its intended output without additional information leakage, regardless of the underlying field.

***Our Contributions*:** We propose (i) **Per-user necessary secrecy condition:** If user $k$ observes $w_{\mathsf{H}}(\mathbf{d}_k^{\mathsf{T}})$[1] server responses, and the common randomness *visible to that user* spans a subspace of dimension less than $w_{\mathsf{H}}(\mathbf{d}_k^{\mathsf{T}}) - 1$, data secrecy cannot be guaranteed. This condition implies the access constraint $w_{\mathsf{H}}(\mathbf{d}_k^{\mathsf{T}}) \leq N - K + 1$ and yields the universal converse $\delta \leq 1 - \frac{K-1}{N}$ on the communication cost. (ii) **Decoding-matrix-only secrecy criterion:** Data secrecy holds if and only if, for each user $k$, removing from $\mathbf{D}$ the columns corresponding to the servers observed by that user reduces the rank by at most one. (iii) **Cost-preserving secure transformation:** Given any admissible (possibly non-secure) factorization $\mathbf{F} = \mathbf{DE}$, we construct a secure scheme by appending to $\mathbf{E}$ any basis of $\mathrm{Null}(\mathbf{D})$ and injecting shared common randomness. Over $\mathbf{GF}(q)$, perfect information-theoretic secrecy (zero leakage) is guaranteed. Over $\mathbb{R}$, data secrecy is achieved by turning every unintended decoding direction into a channel with low signal to noise ratio. Correctness is preserved, and both communication and computation costs remain unchanged under the standard cost model.

Unlike existing secure coded-computing constructions tailored to specific encodings or secret-sharing schemes [6]–[10], [20], [22], our work provides necessary and sufficient structural conditions on the decoding matrix and a universal, cost-preserving transformation that enforces secrecy for any admissible linearly separable multi-user scheme.

## II. System Model

As illustrated in Fig. 1, we consider an $(N, K, L)$ secure multi-user distributed computing system consisting of $N$ servers, $K$ users, and $L$ independent messages $W_1, W_2, \ldots, W_L$, coordinated by a master node. Suppose $N \geq K, L \geq K$, and for each $l \in [L]$, the message $W_l = f_l(D_l)$ is the output of a function $f_l(\cdot)$ that is generally a non-linear and computationally hard function of input data $D_l$. Furthermore, the master node serves as an orchestrator, coordinating the interactions between the users and the servers.

In our setting, each user $k \in [K]$ requests an independent linear combination $\langle \mathbf{f}_k, \mathbf{w} \rangle$ of the message vector $\mathbf{w} = [W_1, \ldots, W_L]^{\mathsf{T}}$, where $\mathbf{f}_k \in \mathbb{F}^L$ is the *request vector* defining the request of user $k$. These users' request vectors are assumed to be independent of the messages, i.e.,

$$I(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K; \mathbf{w}) = 0. \tag{1}$$

[1]For a vector $\mathbf{x}$, the Hamming weight $w_{\mathsf{H}}(\mathbf{x})$ is defined as the number of nonzero components of $\mathbf{x}$.

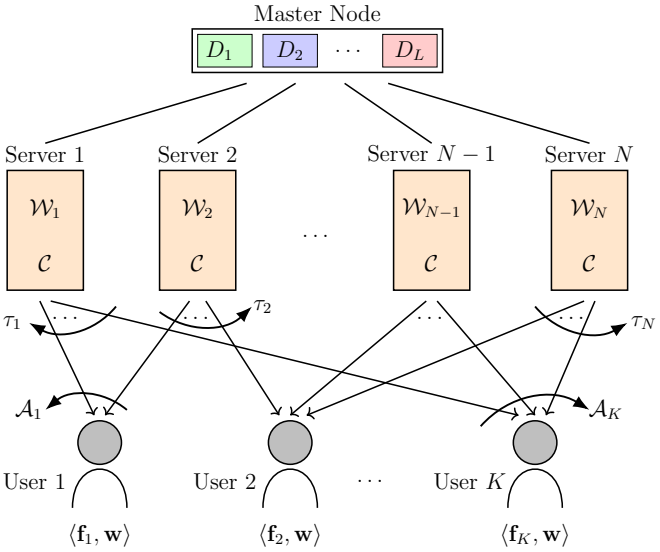


Fig. 1: System model of secure multi-user linearly-separable distributed computing

The messages are independent and identically distributed — either uniformly over $\mathbf{GF}(q)$ or according to $\mathcal{N}(0, \sigma_w^2)$ in the real-valued case — yielding

$$H(\mathbf{w}) = \sum_{i=1}^{L} H(W_i) = LH(W_1). \tag{2}$$

Following the notation in [1], upon receiving the users' request vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K$, the master node constructs full-rank *request matrix* $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K]^{\mathsf{T}} \in \mathbb{F}^{K \times L}$, and performs a matrix factorization $\mathbf{F} = \mathbf{DE}$. This factorization yields the *encoding matrix* $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]^{\mathsf{T}} \in \mathbb{F}^{N \times L}$ specifies the computation tasks assigned to the servers, where $\mathbf{e}_n^{\mathsf{T}}$ corresponds to server $n \in [N]$. This factorization also specifies full-rank *decoding matrix* $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]^{\mathsf{T}} \in \mathbb{F}^{K \times N}$ which determines how each user aggregates the servers' responses, with $\mathbf{d}_k^{\mathsf{T}}$ denoting the aggregation vector of user $k$.

Each server $n \in [N]$ receives $\mathbf{e}_n^{\mathsf{T}}$ and a broadcast schedule $\tau_n$ from the master node, and computes a deterministic function of a subset of messages $\mathcal{W}_n \subseteq \mathcal{W} = \{W_1, \ldots, W_L\}$, possibly using shared independent common randomness $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. This randomness is shared among the servers, is independent of the messages, and is used to guarantee data secrecy. Naturally, for each server $n$, we have

$$H(A_n \mid \mathcal{W}_n, \mathcal{C}, \mathbf{e}_n) = 0. \tag{3}$$

Server $n$ broadcasts its response $A_n = \mathbf{e}_n^{\mathsf{T}}\mathbf{w}$ to the users specified by $\tau_n$.[2] Each user $k \in [K]$ receives the set of responses $\mathcal{A}_k = \{A_n : n \in \mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}})\}$, from a subset of servers indexes by the support $\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}}) \subseteq [N]$ of their own decoding vector $\mathbf{d}_k^{\mathsf{T}}$. Each user $k$ must be able to recover its

[2]Our framework imposes a one-shot constraint: each server is allowed to send only one linear combination, called a response to its assigned users.

requested linear combination using $\mathbf{d}_k^{\mathsf{T}}$. This requirement is captured by the following *correctness condition*:

$$H(\langle \mathbf{f}_k, \mathbf{w} \rangle \mid \mathcal{A}_k, \mathbf{d}_k, \mathbf{f}_k) = 0. \tag{4}$$

Furthermore, user $k$ must learn nothing about the messages more than its requested linear combination. This is formalized by the *data secrecy constraint*. In case of $\mathbf{GF}(q)$, we require

$$I(\mathcal{W}; \mathcal{A}_k, \mathbf{f}_k \mid \langle \mathbf{f}_k, \mathbf{w} \rangle) = 0, \tag{5}$$

while for real-valued computations, data secrecy is satisfied if for any $\varepsilon > 0$, there exists a variance $\sigma_c^2$ for the i.i.d. Gaussian common random variables with $C_i \sim \mathcal{N}(0, \sigma_c^2)$, such that,

$$I(\mathcal{W}; \mathcal{A}_k, \mathbf{f}_k \mid \langle \mathbf{f}_k, \mathbf{w} \rangle) \leq \varepsilon. \tag{6}$$

We briefly summarize that an $(N, K, L)$ secure distributed computing scheme entails the users' request vectors $\{\mathbf{f}_k\}_{k=1}^K$, the servers' encoding vectors $\{\mathbf{e}_n\}_{n=1}^N$, the users' decoding vectors $\{\mathbf{d}_k\}_{k=1}^K$, and the shared common randomness $\mathcal{C}$.

Finally, the *communication cost* is defined as

$$\delta = \frac{\sum_{n=1}^N |\tau_n|}{KN} = \frac{\sum_{k=1}^K w_{\mathsf{H}}(\mathbf{d}_k^{\mathsf{T}})}{KN}, \tag{7}$$

while the *computation cost* is defined as

$$\gamma = \frac{\max_{l \in [L]} w_{\mathsf{H}}(\mathbf{E}(:, l))}{N}, \tag{8}$$

where, for each $l \in [L]$, $w_{\mathsf{H}}(\mathbf{E}(:, l))$ denotes the number of servers that compute the message $W_l$.

**Definition 1.** A pair $(\gamma, \delta)$ is said to be *feasible* if there exists a secure distributed computing scheme with computation cost $\gamma$ and communication cost $\delta$ that satisfies the correctness and data secrecy constraints in (4) and either (5) or (6), depending on the computation field.[3]

**Example 1.** Consider an $(N = 6, K = 4, L = 5)$ multi-user linearly separable distributed computing system with $N = 6$ servers, $K = 4$ users, and $L = 5$ independent messages $W_1, W_2, \ldots, W_5$. Each user $k \in [4]$ requests an independent linear combination $\langle \mathbf{f}_k, \mathbf{w} \rangle$ of the messages, where here $\mathbf{f}_1 = [3, 0, -3, 4, -1]^{\mathsf{T}}$, $\mathbf{f}_2 = [0, 0, 2, 6, 1]^{\mathsf{T}}$, $\mathbf{f}_3 = [0, 3, 1, 3, 1]^{\mathsf{T}}$, and $\mathbf{f}_4 = [3, 3, 6, 1, 1]^{\mathsf{T}}$.

After forming the request matrix

$$\mathbf{F} = \begin{bmatrix} 3 & 0 & -3 & 4 & -1 \\ 0 & 0 & 2 & 6 & 1 \\ 0 & 3 & 1 & 3 & 1 \\ 3 & 3 & 6 & 1 & 1 \end{bmatrix},$$

the master node performs matrix factorization $\mathbf{F} = \mathbf{DE}$, where

$$\mathbf{D} = \begin{bmatrix} 2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \ \mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & -2 & 3 & 0 \\ 0 & 0 & 1 & 3 & 1 \\ -1 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 4 & 2 & 3 & 1 & 1 \end{bmatrix}.$$

[3]In this paper, we exclude the trivial case in which each user can directly access specific components of the common randomness $\mathcal{C}$ via a link to the master node, which would trivially guarantee data secrecy in the multi-user distributed computing problem.

TABLE I: Server-side computations in Example 1.

| Server | Computed linear combination $A_n = \mathbf{e}_n^{\mathsf{T}}\mathbf{w}$ | Broadcast set $\tau_n$ |
|---|---|---|
| 1 | $W_1 + 2W_4$ | $\{1\}$ |
| 2 | $W_1 - 2W_3 + 3W_4$ | $\{1, 2\}$ |
| 3 | $W_3 + 3W_4 + W_5$ | $\{1, 2, 3\}$ |
| 4 | $-W_1 + 3W_3$ | $\{2, 4\}$ |
| 5 | $W_2$ | $\{3, 4\}$ |
| 6 | $4W_1 + 2W_2 + 3W_3 + W_4 + W_5$ | $\{4\}$ |

TABLE II: User-side aggregation in Example 1.

| User | Accessed servers $\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}})$ | Computation $\mathbf{d}_k^{\mathsf{T}}\mathbf{A}$ |
|---|---|---|
| 1 | $\{1, 2, 3\}$ | $2A_1 + A_2 - A_3$ |
| 2 | $\{2, 3, 4\}$ | $A_2 + A_3 + A_4$ |
| 3 | $\{3, 5\}$ | $A_3 + 3A_5$ |
| 4 | $\{4, 5, 6\}$ | $A_4 + A_5 + A_6$ |

Based on $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_6]^{\mathsf{T}}$ and $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_4]^{\mathsf{T}}$, the master node assigns computation tasks and broadcast sets $\{\tau_n\}_{n=1}^6$ to the servers. In particular, each server $n$ computes $A_n = \mathbf{e}_n^{\mathsf{T}}\mathbf{w}$ and broadcasts it to users in $\tau_n$, as summarized in Table I, and each user $k \in [4]$ accesses servers in $\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}})$ and computes $\mathbf{d}_k^{\mathsf{T}}\mathbf{A} = \mathbf{d}_k^{\mathsf{T}}\mathbf{E}\mathbf{w}$, with $\mathbf{A} = [A_1, A_2, \ldots, A_6]^{\mathsf{T}}$, as shown in Table II.

Each user $k \in [4]$ can correctly recover its desired linear combination since $\mathbf{d}_k^{\mathsf{T}}\mathbf{A} = \mathbf{d}_k^{\mathsf{T}}\mathbf{E}\mathbf{w} = \mathbf{f}_k^{\mathsf{T}}\mathbf{w}$. Using (7) and (8), resulting communication and computation costs are $\delta = \frac{\sum_{n=1}^N |\tau_n|}{KN} = \frac{11}{24}, \gamma = \frac{\max_{l \in [L]} w_{\mathsf{H}}(\mathbf{E}(:,l))}{N} = \frac{4}{6}$.

In Example 1, each user $k \in [4]$ has access to a set of responses $\mathcal{A}_k$. Specifically, $\mathcal{A}_1 = \{W_1 + 2W_4,\ W_1 - 2W_3 + 3W_4,\ W_3 + 3W_4 + W_5\}, \mathcal{A}_2 = \{W_1 - 2W_3 + 3W_4,\ W_3 + 3W_4 + W_5,\ -W_1 + 3W_3\}, \mathcal{A}_3 = \{W_3 + 3W_4 + W_5,\ W_2\}, \mathcal{A}_4 = \{-W_1 + 3W_3,\ W_2,\ 4W_1 + 2W_2 + 3W_3 + W_4 + W_5\}$. The collection $\mathcal{A}_k$ generally spans a subspace of dimension greater than one. Consequently, user $k$ may form additional linear combinations beyond its request from $\mathrm{Span}(\mathcal{A}_k)$, which raises the question of whether one can provably guarantee that each user can recover exactly one authorized linear functional of the data, and no other additional information. This question is addressed in the following.

## III. Main Results

Lemma 1 establishes a necessary condition for data secrecy, leading to a converse bound on the communication cost $\delta$ in Theorem 1. Theorem 2 provides a necessary and sufficient condition for data secrecy over both $\mathbb{R}$ and $\mathbf{GF}(q)$.

As one would expect, the solution uses common randomness with *the common randomness coefficient matrix* $\mathbf{C}$ that collects the coefficients used by the servers. For each user $k \in [K]$ connected to the $\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}})$ servers, let $\mathbf{C}(\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}}), :)$ denote the submatrix formed by the rows of $\mathbf{C}$ corresponding to the servers. Moreover, assume that the submatrix $\mathbf{E}(\mathrm{Sup}(\mathbf{d}_k^{\mathsf{T}}), :)$ has full rank.[4] Lemma 1 provides a necessary condition on

[4]This is a mild non-degeneracy condition on $\mathbf{E}$. Without it, some server responses carry redundant message mixtures.

$\mathbf{C}(\mathrm{Sup}(\mathbf{d}_k^\mathsf{T}),:)$ to guarantee data secrecy.

**Lemma 1.** In any multi-user linearly separable distributed computing scenario, where each user $k \in [K]$ is connected to $w_\mathsf{H}(\mathbf{d}_k^\mathsf{T})$ servers, if

$$\mathrm{Rank}\big(\mathbf{C}(\mathrm{Sup}(\mathbf{d}_k^\mathsf{T}),:)\big) < w_\mathsf{H}(\mathbf{d}_k^\mathsf{T}) - 1, \tag{9}$$

then data secrecy cannot be guaranteed for user $k$.

*Proof:* User $k$ observes $w_\mathsf{H}(\mathbf{d}_k^\mathsf{T})$ linear combinations of the messages and the common randomness. Data secrecy requires that no additional information about the messages be revealed. This is possible only if at most one degree of freedom remains after canceling the common randomness, which implies $\mathrm{Rank}\big(\mathbf{C}(\mathrm{Sup}(\mathbf{d}_k^\mathsf{T}),:)\big) < w_\mathsf{H}(\mathbf{d}_k^\mathsf{T})-1$. A detailed proof is given in Appendix A of [28]. ∎

**Theorem 1.** Consider any $(N, K, L)$ multi-user linearly separable distributed computing scheme with $N \geq K$. If the communication cost satisfies

$$\delta > 1 - \frac{K-1}{N},$$

then there exists at least one user that can obtain information beyond its requested linear combination.

*Proof:* Using Lemma 1, we can derive the converse bound $w_\mathsf{H}(\mathbf{d}_k^\mathsf{T}) > N - K + 1$, which limits the number of servers accessible to user $k$ over which data secrecy is impossible. The result follows by applying Lemma 1 to all $K$ users and using (7). A detailed proof is given in Appendix B of [28]. ∎

**Remark 1.** Consider an $(N, K, L)$ scheme with $N \geq K$ and $L = K$ in the fully decentralized case $\mathbf{F} = \mathbf{D}\mathbf{I}_N$ and $\mathbf{E} = \mathbf{I}_N$. Each server $n$ computes $W_n = f_n(D_n)$, giving $\gamma = \frac{1}{N}$ and $\mathbf{D} = \mathbf{F}$, generally with $\delta > 1 - \frac{K-1}{N}$. Thus, at least one user can infer more than its requested function, showing that limited servers' computational capabilities prevents data secrecy.

The following theorem, valid over $\mathbb{R}$ and $\mathbf{GF}(q)$, gives a necessary and sufficient condition on $\mathbf{D}$ for data secrecy. Let $\mathbf{D}(:, \mathrm{Sup}(\mathbf{d}_k^\mathsf{T}))$ denote the submatrix formed by the columns indexed by $\mathrm{Sup}(\mathbf{d}_k^\mathsf{T})$, and define $\mathbf{D}_{\mathrm{Red},k} \triangleq \mathbf{D} \backslash \mathbf{D}(:, \mathrm{Sup}(\mathbf{d}_k^\mathsf{T}))$ to denote the submatrix obtained by removing these columns. Let $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum eigenvalues.

**Theorem 2.** A multi-user linearly separable distributed computing scheme can provide data secrecy if and only if

$$\mathrm{Rank}\big(\mathbf{D} \setminus \mathbf{D}(:, \mathrm{Sup}(\mathbf{d}_k^\mathsf{T}))\big) \geq K - 1, \tag{10}$$

for every user $k$. For the case of $\mathbf{GF}(q)$, this entails zero information leakage, while in the case of the reals, the information leakage to the user $k \in [K]$ is upper bounded as

$$I(\mathcal{W}; \mathcal{A}_k, \mathbf{f}_k \mid \langle \mathbf{f}_k, \mathbf{w} \rangle) \leq \frac{w_\mathsf{H}(\mathbf{d}_k^\mathsf{T}) - 1}{2} \log\left(1 + M_k \frac{\sigma_w^2}{\sigma_c^2}\right), \tag{11}$$

where $M_k = \frac{\lambda_{\max}(\mathbf{X}_k \mathbf{X}_k^\mathsf{T})}{\lambda_{\min}(\mathbf{Y}_k \mathbf{Y}_k^\mathsf{T})}$, $\mathbf{X}_k := \mathbf{E}(\mathcal{S}_k, :)$, and $\mathbf{Y}_k := \mathbf{C}(\mathcal{S}_k, :)$, with $\mathcal{S}_k$ denoting the index set of a maximal linearly independent subset of server responses in $\mathbf{C}(\mathrm{Sup}(\mathbf{d}_k^\mathsf{T}), :)$.

*Proof:* The proof contains two parts: the converse proof of (10) is provided in Appendix C of [28]. The achievable scheme that induces data secrecy when (10) holds is provided in Section IV. The proof of the real-valued leakage bound in (11) is given in Appendix D of [28]. ∎

**Remark 2.** The bound in (11) admits a Gaussian subchannel interpretation. Each observed response is a mixture of message (signal) and randomness (noise), with $\sigma_w^2 \mathbf{X}_k \mathbf{X}_k^\mathsf{T}$ and $\sigma_c^2 \mathbf{Y}_k \mathbf{Y}_k^\mathsf{T}$ capturing their respective contributions. The ratio $M_k = \frac{\lambda_{\max}(\mathbf{X}_k \mathbf{X}_k^\mathsf{T})}{\lambda_{\min}(\mathbf{Y}_k \mathbf{Y}_k^\mathsf{T})}$ acts as an effective signal-to-noise ratio, while $w_\mathsf{H}(\mathbf{d}_k^\mathsf{T}) - 1$ represents the number of independent observations available for leakage. Thus, the total leakage scales like the sum-rate of parallel Gaussian subchannels, and increasing $\sigma_c^2$ suppresses leakage in the real-field setting.

## IV. Achievable Algorithm

This section presents the core idea for achieving data secrecy in multi-user linearly separable distributed computing, along with a general achievable scheme, and finally characterizes decoding matrices $\mathbf{D}$ that satisfy Theorem 2.

### A. Example scheme

**Example 2.** Consider the $(N = 6, K = 4, L = 5)$ multi-user distributed computing scenario as described in Example 1.

**Step 1: Verifying the sufficient condition.** For user $k \in [4]$, the reduced matrices $\mathbf{D}_{\mathrm{Red},k}$ are given by

$$\mathbf{D}_{\mathrm{Red},1} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 3 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{D}_{\mathrm{Red},2} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 1 & 1 \end{bmatrix},$$

$$\mathbf{D}_{\mathrm{Red},3} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{D}_{\mathrm{Red},4} = \begin{bmatrix} 2 & 1 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

For all $k \in [4]$, we have $\mathrm{Rank}(\mathbf{D}_{\mathrm{Red},k}) = 3 = K - 1$, and hence the condition of Theorem 2 is satisfied. Therefore, data secrecy can be induced in this scheme.

**Step 2: Computing a basis for** $\mathrm{Null}(\mathbf{D})$**.** The null space of $\mathbf{D}$ is given by $\mathrm{Null}(\mathbf{D}) = \mathrm{Span}\left\{(-7, 8, -6, -2, 2, 0)^\mathsf{T}, (-1, 2, 0, -2, 0, 2)^\mathsf{T}\right\}$.

**Step 3: Constructing the augmented encoding matrix.** Using the above basis vectors, construct $\mathbf{C}$ (such that the columns of $\mathbf{C}$ span $\mathrm{Null}(\mathbf{D})$), then form $\tilde{\mathbf{E}} = [\mathbf{E}, \mathbf{C}]$, yielding

$$\tilde{\mathbf{E}} = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 & -7 & -1 \\ 1 & 0 & -2 & 3 & 0 & 8 & 2 \\ 0 & 0 & 1 & 3 & 1 & -6 & 0 \\ -1 & 0 & 3 & 0 & 0 & -2 & -2 \\ 0 & 1 & 0 & 0 & 0 & 2 & 0 \\ 4 & 2 & 3 & 1 & 1 & 0 & 2 \end{bmatrix}. \tag{12}$$

**Step 4: Generating augmented message vector.** Create random datasets $C_1$ and $C_2$, which are formed to be independent shared random variables uniformly distributed with the

TABLE III: Server-side computations in Example 2.

| Server | Computed linear combination $A_n = \tilde{\mathbf{e}}_n^\mathsf{T}\tilde{\mathbf{w}}$ | Broadcast set $\tau_n$ |
|---|---|---|
| 1 | $W_1 + 2W_4 - 7C_1 - C_2$ | $\{1\}$ |
| 2 | $W_1 - 2W_3 + 3W_4 + 8C_1 + 2C_2$ | $\{1,2\}$ |
| 3 | $W_3 + 3W_4 + W_5 - 6C_1$ | $\{1,2,3\}$ |
| 4 | $-W_1 + 3W_3 - 2C_1 - 2C_2$ | $\{2,4\}$ |
| 5 | $W_2 + 2C_1$ | $\{3,4\}$ |
| 6 | $4W_1 + 2W_2 + 3W_3 + W_4 + W_5 + 2C_2$ | $\{4\}$ |

same entropy as the messages. This results to an augmented message vector $\tilde{\mathbf{w}} = [W_1, W_2, \ldots, W_5, C_1, C_2]^\mathsf{T}$.

**Step 5: Generating secured responses.** Each server $n$ computes $A_n = \tilde{\mathbf{e}}_n^\mathsf{T}\tilde{\mathbf{w}}$, and transmits $A_n$ to all users in $\tau_n$.

These server-side computations are summarized in Table III.

**Step 6: Computing requested linear combinations.** Each user $k \in [4]$, computes $\mathbf{d}_k^\mathsf{T}\mathbf{A}$, with $\mathbf{A} = [A_1, \ldots, A_6]^\mathsf{T}$, to recover its requested linear combination. These computations are similar to Example 1, and are summarized in Table II.

**Analysis.** We verify that both the correctness condition (4) and the data secrecy condition (5) are satisfied. Detailed proofs are provided in [28], while the general-case arguments are presented in Section IV-B.

Finally, the communication and computation costs remain the same as in Example 1. In particular, $\gamma = \frac{2}{3}$ and $\delta = \frac{11}{24} \leq 1 - \frac{K-1}{N} = \frac{1}{2}$, as required by Theorem 1. Note that $\gamma$ is defined based on the computation tasks assigned to the servers and depends on $\mathbf{E}$, not on the augmented matrix $\tilde{\mathbf{E}}$. Therefore, the pair $(\gamma, \delta)$ is feasible for the proposed scheme.[5]

### *B. General scheme*

Let us assume a known request matrix $\mathbf{F} \in \mathbb{F}^{K\times L}$, an existing (potentially non-secure) scheme defined by the encoding matrix $\mathbf{E} \in \mathbb{F}^{N\times L}$, and the decoding matrix $\mathbf{D} \in \mathbb{F}^{K\times N}$. We now focus on describing the procedure that transforms any admissible scheme into a scheme with data secrecy.

**Step 1:** Given the decomposition $\mathbf{F} = \mathbf{DE}$, the master node checks whether $\mathbf{D}$ satisfies the condition in (10). If the condition is not satisfied, a different factorization is chosen.

**Step 2:** The master node computes a basis for $\mathrm{Null}(\mathbf{D})$.

**Step 3:** The master node arranges the basis vectors of $\mathrm{Null}(\mathbf{D})$ as columns of a matrix $\mathbf{C}$ and constructs the augmented encoding matrix $\tilde{\mathbf{E}} = [\mathbf{E}, \mathbf{C}] = [\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2, \ldots, \tilde{\mathbf{e}}_N]^\mathsf{T}$.

**Step 4:** Using shared common randomness $\mathcal{C}$, the servers generate the independent random datasets $C_1, C_2, \ldots, C_{N-K}$, which are common across the servers. In the case of operating over $\mathbb{R}$, these are distributed as $C_i \sim \mathcal{N}(0, \sigma_c^2)$ for some sufficiently large $\sigma_c^2$, and in the case of $\mathbf{GF}(q)$, these are chosen with uniform distribution to match the entropy of the original output messages. The augmented message vector $\tilde{\mathbf{w}} = [W_1, \ldots, W_L, C_1, \ldots, C_{N-K}]^\mathsf{T}$ is formed.

**Step 5:** Each server $n \in [N]$ computes $A_n = \tilde{\mathbf{e}}_n^\mathsf{T}\tilde{\mathbf{w}}$, and transmits the result to all users in $\tau_n$.

[5]This example can also be considered over $\mathbb{R}$; in that case, $C_i \sim \mathcal{N}(0, \sigma_c^2)$, and $\sigma_c^2$ can be chosen sufficiently large.

**Step 6:** Each user $k \in [K]$ computes $\mathbf{d}_k^\mathsf{T}\mathbf{A}$, with $\mathbf{A} = [A_1, \ldots, A_N]^\mathsf{T}$, to recover its requested linear combination.

**Correctness:** From Steps 1–3, $\mathbf{D}\tilde{\mathbf{E}} = \mathbf{D}[\mathbf{E}, \mathbf{C}] = [\mathbf{F}, \mathbf{0}]$, since $\mathbf{DE} = \mathbf{F}$ and each column of $\mathbf{C}$ lies in $\mathrm{Null}(\mathbf{D})$. Hence, the correctness condition in (4) holds.

**Data Secrecy:** We proceed by contradiction. Suppose that there exists a user $k \in [K]$ who can recover two linearly independent message-bearing linear combinations $\langle \mathbf{f}_k, \mathbf{w}\rangle$ and $\langle \tilde{\mathbf{f}}_k, \mathbf{w}\rangle$. Since the submatrix $\mathbf{E}(\mathrm{Sup}(\mathbf{d}_\mathrm{k}^\mathsf{T}), :)$ has full rank, any decoding vector that cancels the common randomness necessarily produces distinct nonzero linear combination of the messages. So, there exist two linearly independent decoding vectors $\mathbf{a}_k^\mathsf{T}$ and $\tilde{\mathbf{a}}_k^\mathsf{T}$ that cancel the randomness and yield nonzero linear combinations of the messages, i.e., $\langle \mathbf{f}_k, \mathbf{w}\rangle = \mathbf{a}_k^\mathsf{T}\mathbf{E}\mathbf{w} \neq \tilde{\mathbf{a}}_k^\mathsf{T}\mathbf{E}\mathbf{w} = \langle \tilde{\mathbf{f}}_k, \mathbf{w}\rangle$, and $\mathbf{a}_k^\mathsf{T}\mathbf{C}\mathbf{c} = \tilde{\mathbf{a}}_k^\mathsf{T}\mathbf{C}\mathbf{c} = 0$.

Since $\mathbf{C}$ has full-rank, both decoding vectors belong to the row span of the decoding matrix, i.e., $\mathbf{a}_k^\mathsf{T}, \tilde{\mathbf{a}}_k^\mathsf{T} \in \mathrm{Span}\{\mathbf{d}_1^\mathsf{T}, \mathbf{d}_2^\mathsf{T}, \ldots, \mathbf{d}_K^\mathsf{T}\}$. Hence, there exist linearly independent vectors $\mathbf{b}_k, \tilde{\mathbf{b}}_k \in \mathbb{F}^K$ such that $\mathbf{a}_k^\mathsf{T} = \mathbf{b}_k^\mathsf{T}\mathbf{D}, \tilde{\mathbf{a}}_k^\mathsf{T} = \tilde{\mathbf{b}}_k^\mathsf{T}\mathbf{D}$. Moreover, since both decoding vectors are supported only on the servers accessible to user $k$, we have $\mathrm{Sup}(\mathbf{a}_k^\mathsf{T}), \mathrm{Sup}(\tilde{\mathbf{a}}_k^\mathsf{T}) \subseteq \mathrm{Sup}(\mathbf{d}_k^\mathsf{T})$. Thus, $\mathbf{b}_k^\mathsf{T}\mathbf{D}_{\mathrm{Red},k} = \mathbf{0}^\mathsf{T}, \tilde{\mathbf{b}}_k^\mathsf{T}\mathbf{D}_{\mathrm{Red},k} = \mathbf{0}^\mathsf{T}$. Since $\mathbf{b}_k^\mathsf{T}$ and $\tilde{\mathbf{b}}_k^\mathsf{T}$ are linearly independent, the null space of $\mathbf{D}_{\mathrm{Red},k}$ has dimension at least two, implying $\mathrm{Rank}\big(\mathbf{D}_{\mathrm{Red},k}\big) \leq K - 2$. This contradicts the rank condition in (10), which is verified in Step 1 of the construction. Therefore, no user can recover more than a single linear combination of the messages, and the data secrecy requirement in (5) is satisfied. When computations are performed over $\mathbb{R}$, the information-theoretic leakage bound in (11) applies, as proved in Appendix D of [28].

This completes the achievable proof of Theorem 2.

### *C. Some forms of the decoding matrix* $\mathbf{D}$

To induce data secrecy, the decoding matrix $\mathbf{D}$ must satisfy the condition stated in Theorem 2. Several non-trivial classes of matrices naturally satisfy this condition. We highlight a few representative examples. (i) **Systematic form $\mathbf{D} = [\mathbf{I}_K \mid \mathbf{P}]$:** Here, $\mathbf{D}$ is a $K \times N$ matrix, where $\mathbf{I}_K$ denotes the $K \times K$ identity matrix. In $\mathbf{GF}(q)$, $\mathbf{D}$ can be interpreted as the parity-check matrix of a systematic linear code. As before, each column $\mathbf{E}(:, i)$ may then be viewed as a lowest-weight coset leader corresponding to the syndrome $\mathbf{F}(:, i)$ [1], corresponding to the same code. Interestingly, the aforementioned basis vectors of $\mathrm{Null}(\mathbf{D})$ naturally form the generator matrix of the linear code whose parity-check matrix is $\mathbf{D}$. Consequently, the common randomness matrix can be chosen as $\mathbf{C} = [-\mathbf{P}^\mathsf{T} \mid \mathbf{I}_{N-K}]^\mathsf{T}$. This tells us that any decoding matrix $\mathbf{D}$ can be transformed into its systematic form (with a possible change in $\gamma, \delta$) in order to guarantee the structural secrecy condition of Theorem 2. (ii) **Cyclic code structure:** The matrix $\mathbf{D}$ is the generator matrix of a cyclic code in circulant form. In this case, the corresponding parity-check matrix is also circulant. (iii) **Identity matrix $\mathbf{D} = \mathbf{I}$:** Here, $\mathbf{D}$ is a $K \times K$ identity matrix. Each user directly receives their requested linear combination from a single assigned server.

## REFERENCES


[1] A. Khalesi and P. Elia, "Multi-user linearly-separable distributed computing," *IEEE Transactions on Information Theory*, vol. 69, no. 10, pp. 6314–6339, 2023.

[2] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2017.

[3] Q. Yan, S. Yang, and M. Wigger, "Storage-computation-communication tradeoff in distributed computing: Fundamental limits and complexity," *IEEE Transactions on Information Theory*, vol. 68, no. 8, pp. 5496–5512, 2022.

[4] A. Khalesi and P. Elia, "Tessellated distributed computing," *IEEE Transactions on Information Theory*, vol. 71, no. 6, pp. 4754–4784, 2025.

[5] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-server multi-function distributed computation," *Entropy*, vol. 26, no. 6, p. 448, 2024.

[6] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3368–3376.

[7] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5610–5619.

[8] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.

[9] A. Gholami, T. Jahani-Nezhad, K. Wan, and G. Caire, "Optimal communication-computation trade-off in hierarchical gradient coding," in *2025 IEEE International Symposium on Information Theory (ISIT)*, 2025, pp. 1–6.

[10] F. Brunero and P. Elia, "Multi-access distributed computing," *IEEE Transactions on Information Theory*, vol. 70, no. 5, pp. 3385–3398, 2024.

[11] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Transactions on Information Theory*, vol. 68, no. 2, pp. 1259–1278, 2021.

[12] ——, "On the tradeoff between computation and communication costs for distributed linearly separable computation," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7390–7405, 2021.

[13] K. Namboodiri, E. Peter, D. Malak, and P. Elia, "Fundamental limits of distributed computing for linearly separable functions," *arXiv preprint arXiv:2509.23447*, 2025.

[14] ——, "Fundamental limits of multi-user distributed computing of linearly separable functions," *arXiv preprint arXiv:2601.10603*, 2026.

[15] A. Khalesi and P. Elia, "Perfect multi-user distributed computing," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 1349–1354.

[16] A. Khalesi, S. Daei, M. Kountouris, and P. Elia, "Multi-user distributed computing via compressed sensing," *arXiv preprint arXiv:2301.03448*, 2023.

[17] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.

[18] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *2018 Iran Workshop on Communication and Information Theory (IWCIT)*. IEEE, 2018, pp. 1–6.

[19] A. Gholami, K. Wan, T. Jahani-Nezhad, H. Sun, M. Ji, and G. Caire, "Fundamental limits of multi-message private computation," *IEEE Transactions on Communications*, vol. 73, no. 9, pp. 7462–7477, 2025.

[20] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1215–1225.

[21] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[22] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4609–4619, 2020.

[23] K. Tjell and R. Wisniewski, "Privacy in distributed computations based on real number secret sharing," *arXiv preprint arXiv:2107.00911*, 2021.

[24] A. Khalesi, M. Mirmohseni, and M. A. Maddah-Ali, "The capacity region of distributed multi-user secret sharing," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1057–1071, 2021.

[25] A. M. Jafarpisheh, M. Mirmohseni, and M. A. Maddah-Ali, "Distributed attribute-based private access control," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 2856–2861.

[26] S. Meel and S. Ulukus, "Hetdapac: Distributed attribute-based private access control with heterogeneous attributes," in *2024 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2024, pp. 3267–3272.

[27] K. Wan, H. Sun, M. Ji, and G. Caire, "On secure distributed linearly separable computation," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 912–926, 2022.

[28] A. M. Jafarpisheh, A. Khalesi, and P. Elia, "Secure multi-user linearly-separable distributed computing," *arXiv preprint arXiv:2602.02489*, 2026.