On Secure Distributed Linearly Separable Computation

Kai Wan, *Member, IEEE*, Hua Sun, *Member, IEEE*, Mingyue Ji, *Member, IEEE*, and Giuseppe Caire, *Fellow, IEEE*

***Abstract*— Distributed linearly separable computation, where a user asks some distributed servers to compute a linearly separable function, was recently formulated by the same authors and aims to alleviate the bottlenecks of stragglers and communication cost in distributed computation. The data center assigns a subset of input datasets to each server in an uncoded manner, and each server computes some coded packets on the assigned datasets, which are then sent to the user. The user should recover the task function from the answers of a subset of servers, such that the effect of stragglers could be tolerated. In this paper, we formulate a novel secure framework for this distributed linearly separable computation, where we aim to let the user only retrieve the desired task function without obtaining any other information about the input datasets, even if it receives the answers of all servers. In order to preserve the security of the input datasets, some common randomness variable independent of the datasets should be introduced into the transmission. We show that any non-secure linear-coding based computing scheme for the original distributed linearly separable computation problem, can be made secure without increasing the communication cost (number of symbols the user should receive). Then we focus on the case where the computation cost of each server (number of datasets assigned to each server) is minimum and aim to minimize the size of the randomness variable (i.e., randomness size) introduced in the system while achieving the optimal communication cost. We first propose an information theoretic converse bound on the randomness size. We then propose secure computing schemes based on two well-known data assignments, namely *fractional repetition assignment* and *cyclic assignment*. These schemes are optimal subject to using these assignments. Motivated by the observation of the general limitation of these two schemes on the randomness size, we propose a computing scheme with novel assignment, which strictly outperforms the above two schemes. Some additional optimality results are also obtained.**

Manuscript received May 28, 2021; revised October 31, 2021; accepted December 21, 2021. Date of publication January 12, 2022; date of current version February 17, 2022. The work of Kai Wan and Giuseppe Caire was supported in part by the European Research Council under the ERC Advanced Grant 789190 (CARENET). The work of Hua Sun was supported in part by NSF under Grant CCF-2007108 and Grant CCF-2045656. The work of Mingyue Ji was supported in part by NSF under Award 1817154 and Award 1824558. An earlier version of this paper was presented at the 2021 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT45174.2021.9517896]. *(Corresponding author: Kai Wan.)*

Kai Wan and Giuseppe Caire are with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: kai.wan@tu-berlin.de; caire@tu-berlin.de).

Hua Sun is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: hua.sun@unt.edu).

Mingyue Ji is with the Electrical and Computer Engineering Department, The University of Utah, Salt Lake City, UT 84112 USA (e-mail: mingyue.ji@utah.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3142373.

Digital Object Identifier 10.1109/JSAC.2022.3142373

***Index Terms*— Distributed computing, security, straggler mitigation.**

I. Introduction

DISTRIBUTED linearly separable computation, which is a generalization of many existing distributed computing problems such as distributed gradient coding [1] and distributed linear transform [2], was originally proposed in [3] considering two important bottlenecks in the distributed computation systems: communication cost and stragglers. In this computation scenario, a user aims to compute a function of K datasets $(D_1, \ldots, D_{\mathsf{K}})$ on a finite field $\mathbb{F}_{\mathsf{q}}$ through N distributed servers. The task function can be seen as K_{c} linear combinations of K intermediate messages (the n^{th} intermediate message W_n is a function of dataset D_n and contains L symbols). As pointed out in [3], the task function can be seen as the product of the demand matrix and the message matrix. Since the matrix multiplication is one of the key building blocks underlying many data analytics, machine learning algorithms and engineering problems, the considered model also has potential applications in those areas, where each intermediate message represents the output after some pretreatment of the corresponding dataset. The problem contains three phases, *assignment, computing, decoding*. During the assignment phase, the data center with access to the K datasets assigns M datasets to each server in an uncoded manner, where M represents the computation cost of each server. During the computing phase, each server first computes the intermediate message of each dataset assigned to it, and then transmits a coded packet of the computed intermediate messages to the user. During the decoding phase, from the answers of any N_{r} servers, the user should recover the task function such that the system can tolerate $\mathsf{N}-\mathsf{N}_{\mathrm{r}}$ stragglers. The worst-case number of symbols (normalized by L) needed to be received is defined as the communication cost. The objective is to minimize the communication cost for each given computation cost. The optimality results for some cases have been founded in the literature and are summarized below:

- $\mathsf{K}_{\mathrm{c}} = 1$. The computation problem reduces to the distributed gradient coding problem in [1]. When the computation cost is minimum (i.e., $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1)$), the gradient coding scheme in [1] achieves the optimal communication cost (equal to N_{r}) as proved in [3]. Then some extended gradient coding schemes were proposed in [4], [5] which characterize the optimal communication

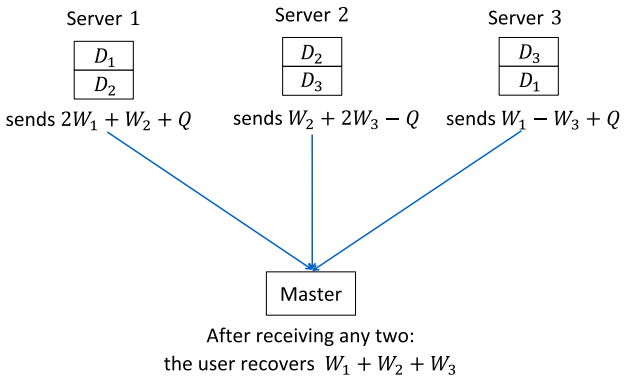


Fig. 1. Secure distributed linearly separable computation with $\mathsf{K} = \mathsf{N} = 3$, $\mathsf{N_r} = 2$, $\mathsf{K_c} = 1$, and $\mathsf{M} = 2$.

cost under the constraint of linear coding, for each possible computation cost.

- *Minimum computation cost* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$. The optimal communication cost with the cyclic assignment (an assignment widely used in the related distributed computing problems) was characterized in [3], when the computation cost is minimum.
- For the general case, [6] proposed a computing scheme under some parameter regimes, which is order optimal within a factor of 2 under the constraint of the cyclic assignment.

In this paper, we consider a novel secure framework for this distributed linearly separable computation problem, where we aim to let the user only retrieve the desired task function without obtaining any other information about the K datasets. We notice that this security model has been widely used in the literature in the context of secure multiparty computation [7], [8] and secure aggregation for federated learning [9], [10].

Let us focus on a small but instructive example in Fig. 1, where $\mathsf{K} = \mathsf{N} = 3$, $\mathsf{N_r} = 2$, $\mathsf{K_c} = 1$, $\mathsf{M} = 2$, and the task function is $W_1 + W_2 + W_3$. Assume the field is $\mathbb{F}_3$. We use the cyclic assignment in [1] to assign D_1 to servers $1, 3$; assign D_2 to servers $1, 2$; assign D_3 to servers $2, 3$. In addition, for the sake of secure computation, the data center generates a randomness variable Q uniformly over $[\mathbb{F}_\mathsf{q}]^\mathsf{L}$, which is independent of the datasets, and assigns Q to each server. In the computing phase of the novel proposed scheme, server 1 computes $2W_1 + W_2 + Q$; server 2 computes $W_2 + 2W_3 - Q$; server 3 computes $W_1 - W_3 + Q$. It can be seen that from the answers of any two servers, the user can recover the task function $W_1 + W_2 + W_3$. Moreover, even if the user receives the answers of all servers, it cannot get any other information about the messages (nor the datasets) because Q is unknown to it. Notice that the communication cost in this example is 2, which is the same as the gradient coding scheme in [1].

The above example shows that it is possible to preserve the security of the datasets (except the task function) from the user. The main questions we ask in this paper are (i) do we need additional communication cost to satisfy this security constraint? (ii) how much randomness is required to guarantee security?

Compared to the existing works on coded secure distributed computation on matrix multiplication (such as [11], [12] etc.), the main differences of the consider secure problem are as follows: (i) in the above existing works, the data center assigns a coded version of all input datasets to each distributed server, while in the considered problem the assignment phase is uncoded; (ii) the above existing works aim to preserve the security of the input datasets from the servers where each distributed server can only access the coded datasets assigned to it, while in the considered problem we aim to preserve the security of the input datasets (except the task function) from the user who may receive all answers of the servers.

A. Contributions

In this paper, we formulate the secure distributed linearly separable computation problem, which prevents the user from obtaining any other information about the datasets except the task. We first show that any non-secure linear-coding based computing scheme for the original distributed linearly separable computation problem, can be made secure without increasing the communication cost. Then we focus on the secure distributed linearly separable computation problem where $\mathsf{K_c} = 1$ and $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$ (i.e., the computation cost is minimum), and aim to minimize the randomness size[1] while achieving the optimal communication cost $\mathsf{N_r}$. To the best of our knowledge, this security issue of the distributed gradient coding has not been considered before. Our contributions on this objective are as follows:

- For each possible assignment, we propose an information theoretic converse bound on the randomness size, which is also a converse bound on the randomness size while achieving the optimal communication cost.
- When $\mathsf{N} - \mathsf{N_r} + 1$ divides N, we propose a secure computing scheme with the fractional repetition assignment in [1], which coincides with the proposed converse bound on the randomness size.
- Under the constraint of the widely used cyclic assignment [1], [3]–[6], [13], [14], we propose an optimal secure computing scheme in the sense that minimum randomness size is achieved.
- Motivated by the observation that the computing scheme with the fractional repetition assignment can only work for the case where $\mathsf{N} - \mathsf{N_r} + 1$ divides N and that the computing scheme with the cyclic assignment is highly suboptimal in terms of the randomness size, we propose a new computing scheme with novel assignment strategies. The novel computing scheme can cover the optimality results of the computing scheme with the fractional repetition assignment; in general it needs a lower randomness size while achieving the optimal communication cost than that of the computing scheme with the cyclic assignment. We also prove that it is optimal when $\frac{\mathsf{N}-\mathsf{N_r}+1}{\mathrm{GCD}(\mathsf{N},\mathsf{N}-\mathsf{N_r}+1)} \leq 4$.

[1] This randomness should be broadcasted from the data center to the servers and stored at the servers; thus reducing the randomness size can reduce the communication cost from the data center and the storage cost at the servers.

B. Paper Organization

The rest of this paper is organized as follows. Section II introduces the secure distributed linearly separable computation problem. Section III provides the main results in this paper and some numerical evaluations. Section IV presents the proposed secure distributed computing schemes. Section V concludes the paper. Some of the proofs are given in the Appendices or in the extended version of this paper [15].

C. Notation Convention

Calligraphic symbols denote sets, bold lower-case letters denote vector, bold upper-case letters denote matrices, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$; $[n] := [1:n]$; $\mathbb{F}_{\mathsf{q}}$ represents a finite field with order q; $\mathbf{M}^{\mathrm{T}}$ and $\mathbf{M}^{-1}$ represent the transpose and the inverse of matrix $\mathbf{M}$, respectively; $\mathbf{I}_n$ represents the identity matrix with dimension $n \times n$; $\mathbf{0}_{m\times n}$ represents the zero matrix with dimension $m \times n$; the matrix $[a;b]$ is written in a Matlab form, representing $[a,b]^{\mathrm{T}}$; $(\mathbf{M})_{m\times n}$ represents that the dimension of matrix $\mathbf{M}$ is $m \times n$; $\mathbf{M}^{(\mathcal{S})_{\mathrm{r}}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the rows of $\mathbf{M}$ with indices in $\mathcal{S}$ (here r represents 'rows'); $\mathbf{M}^{(\mathcal{S})_{\mathrm{c}}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the columns of $\mathbf{M}$ with indices in $\mathcal{S}$ (here c represents 'columns'); $\mathrm{Mod}(b,a)$ represents the modulo operation on b with integer divisor a and in this paper we let $\mathrm{Mod}(b,a) \in \{1,\ldots,a\}$ (i.e., we let $\mathrm{Mod}(b,a) = a$ if a divides b); $\mathrm{GCD}(b,a)$ represents the Greatest Common Divisor of integers b and a; we let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$. In this paper, for each set of integers $\mathcal{S}$, we sort the elements in $\mathcal{S}$ in an increasing order and denote the i^{th} smallest element by $\mathcal{S}(i)$, i.e., $\mathcal{S}(1) < \ldots < \mathcal{S}(|\mathcal{S}|)$.

II. System Model

We formulate a $(\mathsf{K}, \mathsf{N}, \mathsf{N}_{\mathrm{r}}, \mathsf{K}_{\mathrm{c}}, \mathsf{M})$ secure linearly separable computation problem over the canonical user-server distributed system, as illustrated in Fig. 1. Compared to the distributed computing framework in [3], an additional security constraint will be added.

The user wants to compute a function $f(D_1, \ldots, D_{\mathsf{K}})$ on K independent datasets $D_1, \ldots, D_{\mathsf{K}}$. As the data sizes are large, the computing task function is distributed over a group of N servers. For distributed computation to be possible, we assume that the function is *linearly separable* with respect to the datasets, i.e., there exist functions $f_1, \ldots, f_{\mathsf{K}}$ such that $f(\cdot)$ can be written as

$$\begin{aligned} f(D_1, \ldots, D_{\mathsf{K}}) &= g\big(f_1(D_1), \ldots, f_{\mathsf{K}}(D_{\mathsf{K}})\big) \\ &= \mathbf{G}\ [W_1; \ldots; W_{\mathsf{K}}], \end{aligned} \tag{1}$$

where $\mathbf{G}$ is a matrix known by the user and the servers with dimension $\mathsf{K}_{\mathrm{c}} \times \mathsf{K}$, whose elements are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$ for some large enough prime-power q. We model $f_k(D_k)$, $k \in [\mathsf{K}]$, as the k-th message W_k and $f_k(\cdot)$ is an arbitrary function. We assume that the K messages are independent and each message is composed of L uniformly i.i.d. symbols over a finite field $\mathbb{F}_{\mathsf{q}}$, where L is large enough such that any sub-message division is possible. As in [3] we also assume that $\frac{\mathsf{K}}{\mathsf{N}}$ is an integer.

A computation scheme contains three phases, *data assignment*, *computing*, and *decoding*.

A. Data Assignment Phase

The data center/global server assigns each dataset D_k where $k \in [\mathsf{K}]$ to a subset of the N servers in an uncoded manner. The set of datasets assigned to server $n \in [\mathsf{N}]$ is denoted by $\mathcal{Z}_n$, where $\mathcal{Z}_n \subseteq [\mathsf{K}]$. The assignment constraint is that $|\mathcal{Z}_n| \leq \mathsf{M}$, where it will be clarified soon that M represents the computation cost. We define $\mathbf{Z} = (\mathcal{Z}_1, \ldots, \mathcal{Z}_{\mathsf{N}})$.

As an additional problem constraint, we impose that the user learns no further information about $(D_1, \ldots, D_{\mathsf{K}})$ other than the task function $f(D_1, \ldots, D_{\mathsf{K}})$. To this purpose, the data center also generates a randomness variable $Q \in \mathcal{Q}$, and assigns Q to each server $k \in [\mathsf{K}]$. Notice that

$$I(Q; D_1, \ldots, D_{\mathsf{K}}) = I(Q; W_1, \ldots, W_{\mathsf{K}}) = 0. \tag{2}$$

The randomness size η measures the amount of randomness, i.e., $\eta = \frac{H(Q)}{\mathsf{L}}$.

B. Computing Phase

Each server $n \in [\mathsf{N}]$ first computes the message $W_k = f_k(D_k)$ for each $k \in \mathcal{Z}_n$. Then it generates $X_n = \psi_n(\{W_k : k \in \mathcal{Z}_n\}, Q)$, where the encoding function ψ_n is such that $\psi_n : [\mathbb{F}_{\mathsf{q}}]^{|\mathcal{Z}_n|\mathsf{L}} \times |\mathcal{Q}| \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{T}_n}$, and T_n represents the length of X_n. Finally, server n sends X_n to the user. As in [3], we assume that the complexity of computing the messages $\{W_k : k \in \mathcal{Z}_n\}$ is much higher than computing each function ψ_n. Hence, we denote the computation cost by M.

C. Decoding Phase

The computation scheme should tolerate $\mathsf{N} - \mathsf{N}_{\mathrm{r}}$ stragglers. As the user does not know a priori which servers are stragglers, the computation scheme should be designed so that from the answers of any N_{r} servers, the user can recover $\mathbf{G}[W_1; \ldots; W_{\mathsf{K}}]$. Hence, for any subset of servers $\mathcal{A} \subseteq [\mathsf{N}]$ where $|\mathcal{A}| = \mathsf{N}_{\mathrm{r}}$, with the definition

$$X_{\mathcal{S}} := \{X_n : n \in \mathcal{S}\}, \tag{3}$$

there exists a decoding function $\phi_{\mathcal{A}}$ such that

$$\hat{g}_{\mathcal{A}} = \phi_{\mathcal{A}}\big(X_{\mathcal{A}}, \mathbf{F}\big), \tag{4}$$

where the decoding function $\phi_{\mathcal{A}}$ is such that

$$\phi_{\mathcal{A}} : [\mathbb{F}_{\mathsf{q}}]^{\sum_{n\in\mathcal{A}} \mathsf{T}_n} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\mathrm{c}}\mathsf{K}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\mathrm{c}}\mathsf{L}}. \tag{5}$$

The worst-case probability of error is defined as

$$\varepsilon := \max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N}_{\mathrm{r}}} \Pr\{\hat{g}_{\mathcal{A}} \neq g(W_1, \ldots, W_{\mathsf{K}})\}. \tag{6}$$

Q is unknown to the user, and thus Q cannot be used in the decoding procedure in (5). In order to protect the security, even if receiving the answers of all servers in $[\mathsf{N}]$, the user

cannot learn any information about the messages except the desired task function; it should satisfy that[2]

$$I\big(W_1,\ldots,W_{\mathsf{K}};X_{[\mathsf{N}]}|\mathbf{G}[W_1;\ldots;W_{\mathsf{K}}]\big)=0. \tag{7}$$

We denote the communication cost by $\mathsf{R}:=\max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N}_\mathrm{r}}\frac{\sum_{n\in\mathcal{A}}\mathsf{T}_n}{\mathsf{L}}$, representing the maximum normalized number of symbols received by the user from any N_r responding servers. The communication cost R is achievable if there exists a computing scheme satisfying the above constraints and that $\lim_{\mathsf{q}\to\infty}\varepsilon=0$.

When the computation cost is minimum, it was proved in [3, Lemma 1] that **each dataset is assigned to** $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$ **servers and each server obtains** M **datasets**, where

$$\mathsf{M}=|\mathcal{Z}_1|=\cdots=|\mathcal{Z}_{\mathsf{N}}|=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1).$$

In this paper, we mainly focus on the case where $\mathsf{K}_\mathrm{c}=1$ and the computation cost is minimum, and search for the minimum communication cost $\mathsf{R}^\star$. In addition, with the optimal communication cost, we aim to search for the minimum randomness size $\eta^\star$. Note that when $\mathsf{K}_\mathrm{c}=1$, without loss of generality, $f(D_1,\ldots,D_{\mathsf{K}})=\mathbf{G}\,[W_1;\ldots;W_{\mathsf{K}}]=W_1+\cdots+W_{\mathsf{K}}$.[3]

III. Main Results

In this section, we present our main results.

Theorem 1: Any linear-coding based computing scheme $\mathscr{S}$ for the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{M},\mathsf{K}_\mathrm{c})$ non-secure distributed linearly separable computation problem where $\mathsf{K}_\mathrm{c}\in[\mathsf{K}]$, $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+\mathsf{m})$ and $\mathsf{m}\in[\mathsf{N}_\mathrm{r}]$, can be made secure without increasing the communication cost and with the randomness size $\eta_{\mathscr{S}}=\frac{H_{\mathscr{S}}}{\mathsf{L}}-\mathsf{K}_\mathrm{c}$, where $H_{\mathscr{S}}$ represents the entropy of the transmissions by all servers in the scheme $\mathscr{S}$.

The proof of Theorem 1 could be found in Appendix A. From Theorem 1, it can be seen that the additional security constraint does not increase the communication cost for the linear-coding based computing scheme. This is done by proposing a novel transformation approach which adds the security guarantee into any linear-coding based non-secure computing scheme.

Note that the distributed computing scheme in [1] was shown in [3] to be exactly optimal for the non-secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$. Applying this scheme into Theorem 1, we can directly obtain the following theorem, which shows that in this case the optimal communication cost does not change when the security constraint is added.

Theorem 2: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$, the optimal communication cost is $\mathsf{R}^\star=\mathsf{N}_\mathrm{r}$.

From Theorem 1, we can also add the security into the distributed computing schemes in [4], [5] for the case that $\mathsf{K}_\mathrm{c}=1$, into the distributed computing scheme in [3] for the case that $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$, and into the distributed computing scheme in [6] for the case that $\mathsf{K}_\mathrm{c}\in[\mathsf{K}]$ and $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+\mathsf{m})$ where $\mathsf{m}\in[\mathsf{N}_\mathrm{r}]$, without increasing the communication cost.

In the rest of this paper, we focus on the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$, and aim to minimize the randomness size η while achieving the optimal communication cost $\mathsf{R}^\star=\mathsf{N}_\mathrm{r}$.

We first introduce a novel converse bound on η for a fixed assignment, whose proof can be found in Appendix B.

Theorem 3: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$, for a fixed assignment $\mathbf{Z}=(\mathcal{Z}_1,\ldots,\mathcal{Z}_{\mathsf{N}})$, if there exists an ordered set of servers in $[\mathsf{N}]$ denoted by $\mathbf{s}=(s_1,\ldots,s_{|\mathbf{s}|})$, such that

$$\mathcal{Z}_{s_i}\setminus\big(\mathcal{Z}_{s_1}\cup\cdots\mathcal{Z}_{s_{i-1}}\big)\neq\emptyset,\quad\forall i\in[|\mathbf{s}|], \tag{8}$$

it must hold that

$$\eta\geq|\mathbf{s}|-1. \tag{9}$$

Notice that while deriving the converse bound in Theorem 3, we do not use the constraint that communication cost is minimum. Hence, it is a converse on the randomness size, which is also a converse bound on the randomness size while achieving the optimal communication cost.

A general converse bound over all possible assignments is directly obtained from Theorem 3.

Corollary 1: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$, it must hold that

$$\eta^\star\geq\min_{\mathbf{Z}}\max_{\mathbf{s}:\mathcal{Z}_{s_i}\setminus\left(\mathcal{Z}_{s_1}\cup\cdots\mathcal{Z}_{s_{i-1}}\right)\neq\emptyset,\forall i\in[|\mathbf{s}|]}|\mathbf{s}|-1. \tag{10}$$

To solve the min-max optimization problem in (10) is highly combinatorial and becomes a part of on-going works. For some specific cases, this optimization problem has been solved in this paper (see Theorems 4 and 7). In the following, we provide a generally loosen version of the converse bound in (10).

Corollary 2: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ and $\mathsf{K}_\mathrm{c}=1$, it must hold that

$$\eta^\star\geq\left\lceil\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}\right\rceil-1. \tag{11}$$

Proof: By definition, there are K datasets in the library and we assign $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$ datasets to each server. Hence, for any possible assignment, we can find $\lceil\frac{\mathsf{K}}{\mathsf{M}}\rceil=\left\lceil\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}\right\rceil$ servers, where each server has some dataset which is not assigned to other $\left\lceil\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}\right\rceil-1$ servers. By Theorem 3, we have $\eta^\star\geq\left\lceil\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}\right\rceil-1$. ∎

For the achievability part, we use the proposed transformation approach in Theorem 1 to add the security guarantee

[2]Notice $X_{[\mathsf{N}]}$ is a function of $(W_1,\ldots,W_{\mathsf{N}})$ and Q. By the data processing inequality, the security constraint in (7) is equivalent to $I\big(D_1,\ldots,D_{\mathsf{K}};X_{[\mathsf{N}]}|\mathbf{G}[W_1;\ldots;W_{\mathsf{K}}]\big)=0$.

[3] Assume that the task function is $d_1W_1+d_2W_2+\cdots+d_{\mathsf{K}}W_{\mathsf{K}}$, where $d_j\in\mathbb{F}_\mathsf{q}\setminus\{0\}$ for each $j\in[\mathsf{K}]$. Note that each W_j is assumed to be uniformly i.i.d. over $[\mathbb{F}_\mathsf{q}]^\mathsf{L}$. Hence, d_jW_j is also uniformly i.i.d. over $[\mathbb{F}_\mathsf{q}]^\mathsf{L}$. In addition, the elements in $\mathbf{G}$ are uniformly i.i.d. over large enough finite field. Thus with high probability they are not 0, and we can treat the task as $W_1+\cdots+W_{\mathsf{K}}$ for most possible values of $\mathbf{G}$.

for the non-secure computing scheme, with the randomness size equal to the number of transmitted linear combinations of messages by all servers in the original non-secure scheme reduced by 1. Hence, under our construction, minimizing the randomness size is equivalent to minimize the total number of transmitted linear combinations in the non-secure scheme.

We then characterize the optimal randomness size for the case where $\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1$ divides N.

Theorem 4: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},\mathsf{K}_{\mathrm{c}},\mathsf{M})$ *secure distributed linearly separable computation problem where* $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1)$, $\mathsf{K}_{\mathrm{c}}=1$, *and* $\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1$ *divides* N, *to achieve the optimal communication cost, the minimum randomness size is*

$$\eta^{\star}=\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1}-1. \tag{12}$$

Proof: The converse part of (12) directly comes from Corollary 2. For the achievable scheme, we apply Theorem 1 with the non-secure computing scheme with the fractional repetition assignment [1], in which the number of transmitted linear combinations of messages by all servers is $\frac{\mathsf{N}}{\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1}$. Hence, from Theorem 1, to make it secure, the needed randomness size is as in (12). ∎

In the following theorem, we focus on the cyclic assignment.

Theorem 5: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},\mathsf{K}_{\mathrm{c}},\mathsf{M})$ *secure distributed linearly separable computation problem with* $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1)$ *and* $\mathsf{K}_{\mathrm{c}}=1$, *to achieve the optimal communication cost, the minimum randomness size under the constraint of the cyclic assignment is*

$$\eta^{\star}_{cyc}=\mathsf{N}_{\mathrm{r}}-1. \tag{13}$$

Proof: Achievability. we note that in the non-secure computing scheme with the cyclic assignment in [3], in which the number of transmitted linear combinations of messages by all servers is N_{r}. By Theorem 1, the needed randomness size is $\mathsf{N}_{\mathrm{r}}-1$.

Converse. If the cyclic assignment is used, let us focus on an ordered set of N_{r} neighbouring servers

$$\mathbf{s}=\big(\mathsf{N}_{\mathrm{r}},\mathsf{N}_{\mathrm{r}}-1,\ldots,1\big). \tag{14}$$

For each $n\in[\mathsf{N}_{\mathrm{r}}]$, dataset D_n is assigned to servers in $\{n,\mathrm{Mod}(n-1,\mathsf{N}),\ldots,\mathrm{Mod}(n-\mathsf{N}+\mathsf{N}_{\mathrm{r}},\mathsf{N})\}$; thus servers in $\{\mathsf{N}_{\mathrm{r}},\mathsf{N}_{\mathrm{r}}-1,\ldots,n+1\}$ do not know D_n. Hence, the ordered set $\mathbf{s}$ in (14) satisfies the constraint in (8), and we have $\eta^{\star}_{\mathrm{cyc}}\geq|\mathbf{s}|-1=\mathsf{N}_{\mathrm{r}}-1$, which proves (13). ∎

Comparing Theorems 4 and 5, it can be seen that the computing scheme with the cyclic assignment is highly sub-optimal where the multiplicative gap to the optimality could be unbounded.[4] However, when $\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1$ does not divide N, the fractional repetition assignment in [1] cannot be used. On the observation that most of existing works on this distributed linearly separable computing problem (without security) are either based on the cyclic assignment (such as [1], [3]–[6], [14]) or the fractional repetition assignment (such as [1], [16]), we need to design new assignments for the considered secure computation problem.

For the ease of notation, we define that

$$\mathsf{M}'\coloneqq\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1. \tag{15}$$

In Section IV, we will propose five novel achievable schemes for different ranges of system parameters. The performance of the combined scheme given in the following theorem is based on a recursive algorithm illustrated in Fig. 2, which will be explained in Remark 1.

Theorem 6: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},\mathsf{K}_{\mathrm{c}},\mathsf{M})$ *secure distributed linearly separable computation problem with* $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{M}'$ *and* $\mathsf{K}_{\mathrm{c}}=1$, *to achieve the optimal communication cost, the randomness size* $\eta=h(\mathsf{N},\mathsf{M}')-1$ *is achievable, where the output of the function* $h(\mathsf{N},\mathsf{M}')$ *is given by the recursive algorithm illustrated in Fig. 2 with the following properties:*

- *By directly using the scheme with the fractional repetition assignment for Theorem 4,*

$$h(\mathsf{N},1)=\mathsf{N}. \tag{16}$$

- *By Scheme 1 described in Section IV-A,*

$$h(\mathsf{N},\mathsf{M}')=h\left(\frac{\mathsf{N}}{GCD(\mathsf{N},\mathsf{M}')},\frac{\mathsf{M}'}{GCD(\mathsf{N},\mathsf{M}')}\right). \tag{17}$$

- *For the case where* $\mathsf{N}>2\mathsf{M}'$, *by Scheme 2 described in Section IV-B,*

$$h(\mathsf{N},\mathsf{M}')=h\big(\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}'-1\rfloor\,\mathsf{M}',\mathsf{M}'\big)+\lfloor\mathsf{N}/\mathsf{M}'-1\rfloor. \tag{18}$$

- *For the case where* $1.5\mathsf{M}'\leq\mathsf{N}<2\mathsf{M}'$ *and* M' *is even, by Scheme 3 described in Section IV-C,*

$$h(\mathsf{N},\mathsf{M}')=h\left(\mathsf{N}-\mathsf{M}',\frac{\mathsf{M}'}{2}\right)+1. \tag{19}$$

- *For the case where* $1.5\mathsf{M}'\leq\mathsf{N}<2\mathsf{M}'$ *and* M' *is odd, by Scheme 4 described in Section IV-D,*

$$h(\mathsf{N},\mathsf{M}')=\mathsf{N}-\frac{3\mathsf{M}'-5}{2}. \tag{20}$$

- *For the case where* $\mathsf{M}'<\mathsf{N}<1.5\mathsf{M}'$, *by Scheme 5 described in Section IV-E,*

$$h(\mathsf{N},\mathsf{M}')=h(\mathsf{M}',2\mathsf{M}'-\mathsf{N}). \tag{21}$$

It will be clarified later that there must exist an output for each input case in the flow diagram illustrated in Fig. 2. Notice that, the needed randomness size of the combined scheme for Theorem 6 is

$$h(\mathsf{N},\mathsf{M}')-1\leq\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{M}'+1)-1=\mathsf{N}_{\mathrm{r}}-1,$$

where $\mathsf{N}_{\mathrm{r}}-1$ is the needed randomness size of the computing scheme with the cyclic assignment for Theorem 5. In addition, the multiplicative gap between the needed randomness sizes of the computing scheme with the cyclic assignment for Theorem 5 and the combined scheme for Theorem 6, could be unbounded (see the numerical evaluations at the end of this section).

[4] For example, when $\mathsf{N}=2(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1)$ and N is very large, the optimal randomness size is 1 as shown in (12), while the needed randomness size of the computing scheme with the cyclic assignment is $\mathsf{N}_{\mathrm{r}}-1=\frac{\mathsf{N}}{2}$.

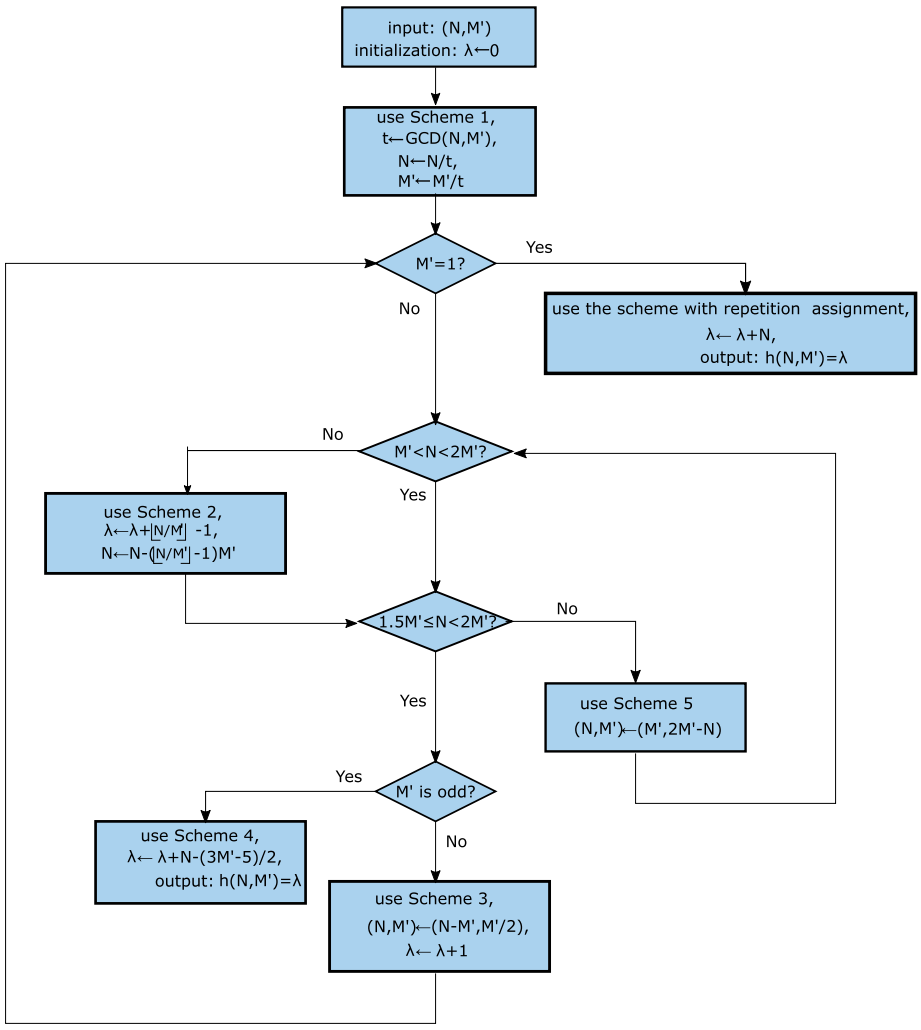


Fig. 2. Flow diagram of the combined scheme in Theorem 6. Notice that the condition to use Scheme 5 is that $\mathsf{M}' < \mathsf{N} < 1.5\mathsf{M}'$; in this case, $2\mathsf{M}' - \mathsf{N} > 1$.

Remark 1 (High-Level Ideas for Theorem 6): We divide the K datasets into N non-overlapping and equal-length groups, where the i^{th} group denoted by $\mathcal{G}_i = \{k \in [\mathsf{K}] : \text{Mod}(k, \mathsf{N}) = i\}$ contains $\frac{\mathsf{K}}{\mathsf{N}}$ datasets, for each $i \in [\mathsf{N}]$. Group $\mathcal{G}_i$ is assigned to $\mathsf{M}' = \mathsf{N} - \mathsf{N}_{\text{r}} + 1$ servers, each of which can compute the merged message W_i'. Hence, we treat the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_{\text{r}}, 1, \mathsf{M})$ secure distributed linearly separable computation problem as the $(\mathsf{N}, \mathsf{N}, \mathsf{N}_{\text{r}}, 1, \mathsf{M}')$ secure distributed linearly separable computation problem.

As in Appendix A, the design on the computing phase contains two stages.

- In the first stage, we do not consider the security constraint in (7). We let each server send one linear combination of merged messages which it can compute, such that from any set of N_{r} responding servers, the user can recover $W_1' + \cdots + W_{\mathsf{N}}'$. Assume that from the answers of all servers, the user can recover $\mathbf{F}[W_1'; \ldots; W_{\mathsf{N}}']$ where the dimension of $\mathbf{F}$ is $\lambda \times \mathsf{N}$ and λ represents the number of totally transmitted linearly independent combinations of merged messages. Thus the transmission of server $n \in [\mathsf{N}]$ can be expressed as $\mathbf{s}_n \mathbf{F}[W_1'; \ldots; W_{\mathsf{N}}']$, where $\mathbf{s}_n$ represents the transmission vector of server n.
- In the second stage, we take the security constraint in (7) into consideration. We introduce $\lambda - 1$ independent randomness variables $Q_1, \ldots, Q_{\lambda-1}$, where $Q_i, i \in [\lambda - 1]$, is uniformly i.i.d. over $[\mathbb{F}_{\mathsf{q}}]^{\mathsf{L}}$. We then generate the matrix $\mathbf{F}' = [(\mathbf{F})_{\lambda \times \mathsf{N}}, (\mathbf{S})_{\lambda \times (\lambda-1)}]$, where $\mathbf{S} = [\mathbf{0}_{1 \times (\lambda-1)}; \mathbf{S}']$ and $\mathbf{S}'$ is full-rank with dimension $(\lambda - 1) \times (\lambda - 1)$. We let each server $n \in [\mathsf{N}]$ transmit $\mathbf{s}_n \mathbf{F}'[W_{1,1}; \ldots; W_{\mathsf{K}}; Q_1; \ldots; Q_{\lambda-1}]$. It is proved in Appendix A that the resulting scheme is decodable and secure. The needed randomness size η is equal to $\lambda - 1$.

The second stage can be immediately obtained once the first stage is fixed. **Hence, now we only need to focus on the first stage where we aim to minimize the number of totally transmitted linearly independent combinations (i.e., λ) for the $(\mathsf{N}, \mathsf{N}, \mathsf{N}_{\text{r}}, 1, \mathsf{M}')$ non-secure distributed linearly separable computation problem (for the sake of simplicity, we will call it $(\mathsf{N}, \mathsf{M}')$ non-secure problem since $\mathsf{N}_{\text{r}} = \mathsf{N} - \mathsf{M}' + 1$).**

The flow diagram of the combined scheme for Theorem 6 whose output value λ is assigned to $h(\mathsf{N},\mathsf{M}')$, is given in Fig. 2. The procedure in the flow diagram is finished when either $\mathsf{M}'=1$ or $1.5\mathsf{M}'\leq\mathsf{N}<2\mathsf{M}'$ and M' is odd. There must exist an output for each input case because when none of the above two constraints are satisfied, M' will be further reduced.

For the secure computing scheme in Theorem 6, the decoding complexity (i.e., the number of multiplications) is $O\left(h(\mathsf{N},\mathsf{M}')\mathsf{L}\right)$, while the computation complexity at the server side is mainly due to the computation on the messages from the assigned datasets (as stated in Section II). □

Comparing the achievable scheme in Theorem 6 with the proposed converse bounds in Corollaries 1 and 2, we can characterize the following optimality result, whose proof could be found in the extended version of this paper [15, Appendix C].

Theorem 7: For the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},\mathsf{K}_{\mathrm{c}},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{M}'$, $\mathsf{K}_{\mathrm{c}}=1$, and $\frac{\mathsf{M}'}{\mathrm{GCD}(\mathsf{N},\mathsf{M}')}\leq 4$, to achieve the optimal communication cost, the minimum randomness size is $h(\mathsf{N},\mathsf{M}')-1$, where $h(\cdot,\cdot)$ is defined in Theorem 6.

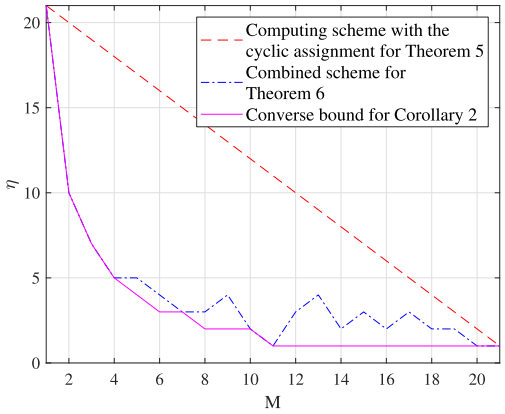
Notice that when $\frac{\mathsf{M}'}{\mathrm{GCD}(\mathsf{N},\mathsf{M}')}=1$, M' divides N; in this case Theorem 7 reduces to Theorem 4.

At the end of this section, we provide some numerical evaluations to compare the needed randomness sizes of the computing scheme with the cyclic assignment for Theorem 5 (equal to $\mathsf{N}_{\mathrm{r}}-1$) and the combined scheme for Theorem 6 (equal to $h(\mathsf{N},\mathsf{M}')-1$), while achieving the optimal communication cost. We consider the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},\mathsf{K}_{\mathrm{c}},\mathsf{M})$ secure distributed linearly separable computation problem with $\mathsf{K}=\mathsf{N}$, $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{M}'$, and $\mathsf{K}_{\mathrm{c}}=1$. In Fig.3a, we fix $\mathsf{N}=22$ and plot the tradeoffs between M and η. In Fig. 3b, we fix $\mathsf{M}=8$ and plot the tradeoffs between N and η. From both figures, it can be seen that the combined scheme for Theorem 6 needs a much lower randomness size than that in Theorem 5.

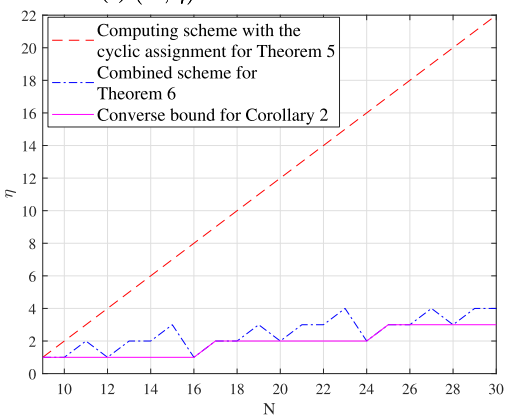
IV. Novel Achievable Schemes for Theorem 6

As explained in Remark 1, by a grouping strategy, we treat the $(\mathsf{K},\mathsf{N},\mathsf{N}_{\mathrm{r}},1,\mathsf{M})$ secure distributed linearly separable computation problem as the $(\mathsf{N},\mathsf{N},\mathsf{N}_{\mathrm{r}},1,\mathsf{M}')$ secure distributed linearly separable computation problem. For the ease of notation, in this section we directly consider the case where $\mathsf{K}=\mathsf{N}$; thus we also have $\mathsf{M}=\mathsf{M}'$. In the rest of this section, we consider a $(\mathsf{N},\mathsf{N},\mathsf{N}_{\mathrm{r}},1,\mathsf{M})$ non-secure distributed linearly separable computation problem (a.k.a., (N,M) non-secure problem), and aim to minimize the number of totally transmitted linear combinations of messages λ while achieving the optimal communication cost N_{r}.

The proofs of the decodability of Schemes 4 and 5 follow the decodability proof in [3, Appendix C], based on the Schwartz-Zippel lemma [17]–[19]. Due to the limitation of pages, we put those proofs in the extended version of this paper [15].



(a) (M,η) tradeoff with $\mathsf{N}=22$.



(b) (N,η) tradeoff with $\mathsf{M}=8$.

Fig. 3. Numerical evaluations for the considered secure distributed linearly separable computation problem.

A. Scheme 1 for (17)

We consider the (N,M) non-secure problem where $\mathrm{GCD}(\mathsf{N},\mathsf{M})>1$, and aim to construct a scheme (Scheme 1) to prove (17). Intuitively, we want to consider a set of $\mathrm{GCD}(\mathsf{N},\mathsf{M})$ messages as a single message and a set of $\mathrm{GCD}(\mathsf{N},\mathsf{M})$ servers as a single server. Thus Scheme 1 is a recursive scheme which is based on the proposed scheme for the $\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})},\frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$ non-secure problem. We assume that the latter scheme has been designed before, whose number of totally transmitted linearly independent combinations of messages is $h\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})},\frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$.

We first partition the N datasets into $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$ groups, where the i^{th} group is

$$\mathcal{K}_i=[(i-1)\mathrm{GCD}(\mathsf{N},\mathsf{M})+1:i\ \mathrm{GCD}(\mathsf{N},\mathsf{M})]$$

for each $i\in\left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$. In addition, we let $M_i=\sum_{k\in\mathcal{K}_i}W_k$; thus the task function could be expressed as $W_1+\cdots+W_{\mathsf{N}}=M_1+\cdots+M_{\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}}$.

We also partition the N servers into $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$ groups, where the i^{th} group of servers is

$$\mathcal{U}_i=[(i-1)\mathrm{GCD}(\mathsf{N},\mathsf{M})+1:i\ \mathrm{GCD}(\mathsf{N},\mathsf{M})]$$

for each $i\in\left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$.

We now prove that the proposed scheme for the $\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}, \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$ non-secure problem can be directly applied to the (N,M) non-secure problem.

In the proposed scheme for the $\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}, \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$ non-secure problem, we assume that the set of assigned datasets to each server $n' \in \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$ is $\mathcal{Z}'_{n'} \subseteq \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$; obviously, $|\mathcal{Z}'_{n'}| = \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$. In the computing phase, server n' computes a linear combination of the $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$ messages, where the coefficients of the messages with indices in $\left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right] \setminus \mathcal{Z}'_{n'}$ are 0. We assume that the vector of the coefficients in this linear combination is $\mathbf{v}_{n'}$, containing $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$ elements. From the answers of any $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} - \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} + 1$ servers, the user can recover the sum of the $\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}$ messages.

We then apply the above scheme to the (N,M) non-secure problem.

1) Assignment Phase: For each $i \in \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$, we assign all datasets in group $\mathcal{K}_i$ to each server in group $\mathcal{U}_j$ where $j \in \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$ and $i \in \mathcal{Z}'_j$. As each group of servers contains $\mathrm{GCD}(\mathsf{N},\mathsf{M})$ servers, each dataset is assigned to $\mathrm{GCD}(\mathsf{N},\mathsf{M})\frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} = \mathsf{M}$ servers; as each group of datasets contains $\mathrm{GCD}(\mathsf{N},\mathsf{M})$ datasets, the number of datasets assigned to each server is $\mathrm{GCD}(\mathsf{N},\mathsf{M})\frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} = \mathsf{M}$. Thus the assignment constraints are satisfied.

2) Computing Phase: For each $i \in \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$, we let each server in group $\mathcal{U}_i$ compute

$$\mathbf{v}_n \left[M_1; \ldots; M_{\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}}\right],$$

where $\mathbf{v}_n$ represents the vector of the coefficients in the linear combination sent by server n in the $\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}, \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$ non-secure problem.

3) Decoding Phase: Following the original scheme for the $\left(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}, \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right)$ non-secure problem, for any set $\mathcal{A} \subseteq \left[\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right]$ where $|\mathcal{A}| = \frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} - \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} + 1$, if the user receives the answers of the servers in $\mathcal{A}$, it can recover the task function.

Let us go back to the (N,M) non-secure problem. The user can receive the answers of $\mathsf{N}-\mathsf{M}+1$ servers. As each group of servers contains $\mathrm{GCD}(\mathsf{N},\mathsf{M})$ servers, it can be seen that these $\mathsf{N}-\mathsf{M}+1$ servers are from at least $\left\lceil\frac{\mathsf{N}-\mathsf{M}+1}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\right\rceil = \frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} - \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})} + 1$ groups. Hence, the user recovers the task function.

In conclusion, we proved $h(\mathsf{N},\mathsf{M}) = h\big(\frac{\mathsf{N}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}, \frac{\mathsf{M}}{\mathrm{GCD}(\mathsf{N},\mathsf{M})}\big)$, coinciding with (17).

B. Scheme 2 for (18)

We will start with an example to illustrate the main idea.

Example 1: We consider the $(\mathsf{N},\mathsf{M}) = (5,2)$ *non-secure problem. It can be seen that in this example* $\mathsf{N}_\mathrm{r} = \mathsf{N} - \mathsf{M} + 1 = 4$. *For the sake of simplicity, while illustrating the proposed schemes through examples, we assume that the field is a large enough prime field. In general this assumption on prime field size is not necessary in our proposed schemes, since we only need the field size to be large enough such that the Schwartz-Zippel lemma could be used to show the decodability.*

Assignment phase. We assign the datasets as follows.

Server 1	*Server 2*	*Server 3*	*Server 4*	*Server 5*
D_1	D_1	D_3	D_4	D_5
D_2	D_2	D_4	D_5	D_3

Computing phase. We let servers 1 *and* 2 *compute* W_1+W_2. *We then focus on servers* $3,4,5$. *It can be seen that datasets* D_4, D_5, D_6 *are assigned to servers* $3,4,5$ *in a cyclic way. Hence, as the computing scheme illustrated in the Introduction, we let server* 3 *compute* $2W_3+W_4$*; let server* 4 *compute* W_4+2W_5*; let server* 5 *compute* W_3-W_5.

Decoding phase. Among the answers of any $\mathsf{N}_\mathrm{r}=4$ *servers, there must exist* W_1+W_2 *and two answers of servers* $3,4,5$. *From any two answers of servers* $3,4,5$*, the user can recover* $W_3+W_4+W_5$. *Together with* W_1+W_2*, the user can recover* $W_1+\cdots+W_5$.

It can be seen that the number of linearly independent combinations transmitted by servers $3,4,5$ *is two. Hence, the number of totally transmitted linearly independent combinations is* $h(5,2)=3$*, which is equal to* $h(3,2)+1$ *coinciding with* (18). □

We now consider the (N,M) non-secure problem where $\mathsf{N} > 2\mathsf{M}$, and aim to construct a scheme (Scheme 2) to prove (18). Scheme 2 is a recursive scheme which is based on the proposed scheme for the $(\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M},\mathsf{M})$ non-secure problem. We assume that the latter scheme has been designed before, whose number of totally transmitted linearly independent combinations of messages is $h\left(\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M},\mathsf{M}\right)$.

1) Assignment Phase: We divide the whole system into $\lfloor\mathsf{N}/\mathsf{M}\rfloor$ blocks. For each $i \in [\lfloor\mathsf{N}/\mathsf{M}\rfloor]$, the i^{th} block contains datasets $\{D_k : k \in \mathcal{B}_i\}$ and servers in $\mathcal{B}_i$, where

$$\mathcal{B}_i = \begin{cases} [(i-1)\mathsf{M}+1 : i\mathsf{M}], & \text{if } i \in [\lfloor\mathsf{N}/\mathsf{M}-1\rfloor]; \\ [\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M}+1 : \mathsf{N}], & \text{if } i = \lfloor\mathsf{N}/\mathsf{M}\rfloor. \end{cases} \tag{22}$$

The datasets in one block are only assigned to the servers in the same block. More precisely, for each $i \in [\lfloor\mathsf{N}/\mathsf{M}\rfloor]$,

- if $i \in [\lfloor\mathsf{N}/\mathsf{M}-1\rfloor]$, we assign all datasets in $\{D_k : k \in \mathcal{B}_i\}$ to each server in $\mathcal{B}_i$;
- if $i = \lfloor\mathsf{N}/\mathsf{M}\rfloor$, the block contains $\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M}$ servers and $\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M}$ datasets, where each dataset should be assigned to M servers and each server should obtain M datasets. Hence, we can apply the assignment phase of the proposed scheme for the $(\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M},\mathsf{M})$ non-secure problem, to assign the datasets $\{D_k : k \in \mathcal{B}_i\}$ to the servers in $\mathcal{B}_i$.

2) Computing Phase: For each $i \in [\lfloor\mathsf{N}/\mathsf{M}-1\rfloor]$, we let the servers in the i^{th} block compute

$$\sum_{k\in\mathcal{B}_i} W_k.$$

We then focus on the i^{th} block where $i = \lfloor\mathsf{N}/\mathsf{M}\rfloor$ (i.e., the last block), to which we apply the computing phase of the proposed scheme for the $(\mathsf{N}-\lfloor\mathsf{N}/\mathsf{M}-1\rfloor\mathsf{M},\mathsf{M})$

non-secure problem. In the proposed scheme for the $(\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M},\mathsf{M})$ non-secure problem, server $n' \in [\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}]$ computes a linear combination of the $\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}$ messages, where the coefficients of the messages it cannot compute are 0. We assume that the vector of the coefficients in this linear combination is $\mathbf{v}_{n'}$, containing $\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}$ elements.

Go back to the i^{th} block, where $i=\lfloor \mathsf{N}/\mathsf{M}\rfloor$, of the (N,M) non-secure problem. For each $j\in[|\mathcal{B}_i|]$, we let server $\mathcal{B}_i(j)$ compute[5]

$$\mathbf{v}_j \ \left[W_{\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}+1};W_{\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}+2};\ldots;W_{\mathsf{N}}\right],$$

where $\mathbf{v}_j$ represents the vector of the coefficients in the linear combination sent by server j in the $(\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M},\mathsf{M})$ non-secure problem.

3) Decoding Phase: The user receives the answers of $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1$ servers. In other words, the user does not receive the answers of $\mathsf{M}-1$ servers. Recall that in each of the first $\lfloor \mathsf{N}/\mathsf{M}-1\rfloor$ blocks there are M servers; in the last block there are $\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}$ servers. Hence, among these $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1$ responding servers, there must be at least one server in each of the first $\lfloor \mathsf{N}/\mathsf{M}-1\rfloor$ blocks, and at least $\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}-\mathsf{M}+1$ servers in the last block. By construction, from the answers of any $\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}-\mathsf{M}+1$ servers in the last block, the user can recover $\sum_{k\in[\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}+1:\mathsf{N}]}W_k$. Together with the transmissions of the first $\lfloor \mathsf{N}/\mathsf{M}-1\rfloor$ blocks, the user can recover $W_1+\cdots+W_{\mathsf{N}}$.

In conclusion, we proved $h(\mathsf{N},\mathsf{M})=\lfloor \mathsf{N}/\mathsf{M}-1\rfloor+h\left(\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M},\mathsf{M}\right)$, coinciding with (18). In addition, it can be seen that $\mathsf{M}<\mathsf{N}-\lfloor \mathsf{N}/\mathsf{M}-1\rfloor \mathsf{M}<2\mathsf{M}$ if $\mathsf{N}>2\mathsf{M}$ and M does not divide N.

C. Scheme 3 for (19)

We first provide an example to illustrate the main idea.

Example 2: We consider the $(\mathsf{N},\mathsf{M})=(7,4)$ non-secure problem. Notice that $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1=4$.

Assignment phase. We assign the datasets as follows.

Server 1	*Server 2*	*Server 3*	*Server 4*
D_1	D_1	D_1	D_1
D_2	D_2	D_5	D_5
D_3	D_3	D_6	D_6
D_4	D_4	D_7	D_7
Server 5	*Server 6*	*Server 7*	
D_2	D_3	D_4	
D_5	D_6	D_7	
D_3	D_4	D_2	
D_6	D_7	D_5	

Computing phase. We let servers $1,2$ compute a same linear combination of messages, assumed to be A_1. Similarly, we let servers $3,4$ compute a same linear combination of messages, assumed to be A_2. Recall that $\mathsf{N}_{\mathrm{r}}=4$. Thus from A_1 and A_2, the user should recover the task function. We construct A_1 and A_2 such that from A_1 and A_2, we can recover the following two linear combinations, $F_1=W_1+\cdots+W_7$ and $F_2=W_2+W_3+W_4+2(W_5+W_6+W_7)$. This can be done by letting $A_1=2F_1-F_2=W_1+W_2+W_3+W_4$, which can be computed by servers $1,2$, and letting $A_2=F_2-F_1=-W_1+W_5+W_6+W_7$, which can be computed by servers $3,4$.

We then focus on servers $5,6,7$. The assignment for servers $5,6,7$ can be expressed as follows. We divide the datasets in $[2:7]$ into three pairs, $\mathcal{P}_1=\{2,5\}$, $\mathcal{P}_2=\{3,6\}$, $\mathcal{P}_3=\{4,7\}$. The three pairs of datasets are assigned to servers $5,6,7$ in a cyclic way. We also let $P_1=W_2+2W_5$, $P_2=W_3+2W_6$, $P_3=W_4+2W_7$. Hence, we can treat servers $5,6,7$ and P_1,P_2,P_3 as a $(3,2)$ non-secure problem, where from the answers of any two servers we can recover $F_2=P_1+P_2+P_3$. We construct the answers of servers $5,6,7$ (denoted by A_3,A_4,A_5, respectively) as $A_3=2P_1+P_2$, $A_4=P_2+2P_3$, and $A_5=P_1-P_3$.

Decoding phase. As shown before, if the set of $\mathsf{N}_{\mathrm{r}}=4$ responding servers contains one server in $[2]$ and one server in $\{3,4\}$, the user can recover the task function from A_1 and A_2.

We then consider the case where from the answers of the responding servers, the user can only receive one of A_1 and A_2. In this case, the set of responding servers must contain at least two servers in $[5:7]$. By construction, from the answers of any two servers in $[5:7]$, the user can recover F_2. Together with $A_1=2F_1-F_2$ or with $A_2=F_2-F_1$, the user can recover F_1, which is the task function.

The number of totally transmitted linearly independent combinations is $h(7,4)=3$, which is equal to $h(3,2)+1$ coinciding with (19). □

We now consider the (N,M) non-secure problem where $1.5\mathsf{M}\le\mathsf{N}<2\mathsf{M}$ and M is even, and aim to construct a scheme (Scheme 3) to prove (19). Scheme 3 is a recursive scheme which is based on the proposed scheme for the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem. We assume that the latter scheme has been designed before, whose number of totally transmitted linearly independent combinations of messages is $h\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$.

We define that $\mathsf{N}=2\mathsf{M}-\mathsf{y}$. In this case, we have $\mathsf{y}\le\mathsf{M}/2$ and $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1=\mathsf{M}-\mathsf{y}+1\le\mathsf{M}$.

Assignment phase. We first focus on the assignment for the servers in $[\mathsf{M}]$, which is given as follows,

Server 1	$\cdots$	Server $\frac{\mathsf{M}}{2}$	Server $\frac{\mathsf{M}}{2}+1$	$\cdots$	Server M
D_1	$\cdots$	D_1	D_1	$\cdots$	D_1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
D_{y}	$\cdots$	D_{y}	D_{y}	$\cdots$	D_{y}
$D_{\mathsf{y}+1}$	$\cdots$	$D_{\mathsf{y}+1}$	$D_{\mathsf{M}+1}$	$\cdots$	$D_{\mathsf{M}+1}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
D_{M}	$\cdots$	D_{M}	D_{N}	$\cdots$	D_{N}

It can be seen that we assign $D_1,\ldots,D_{\mathsf{y}}$ to all servers in $[\mathsf{M}]$, and assign each dataset D_k where $k\in[\mathsf{y}+1:\mathsf{N}]$ to $\frac{\mathsf{M}}{2}$ servers in $[\mathsf{M}]$.

We then focus on the assignment for the servers in $[\mathsf{M}+1:\mathsf{N}]$. We need to assign $\mathsf{N}-\mathsf{y}=2(\mathsf{N}-\mathsf{M})$ datasets (which are in $[\mathsf{y}+1:\mathsf{N}]$) to totally $\mathsf{N}-\mathsf{M}$ servers, where each dataset is assigned to $\frac{\mathsf{M}}{2}$ servers and each server obtains M datasets. We divide datasets in $[\mathsf{y}+1:\mathsf{N}]$ into $\frac{\mathsf{N}-\mathsf{y}}{2}=\mathsf{N}-\mathsf{M}$ pairs, where the i^{th} pair is $\mathcal{P}_i=\{\mathsf{y}+i,\mathsf{M}+i\}$ for each $i\in[\mathsf{N}-\mathsf{M}]$. Hence,

[5]Recall that $\mathcal{B}_i(j)$ represents the j^{th} element in $\mathcal{B}_i$.

we can apply the assignment phase of the proposed scheme for the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem, to assign $\frac{\mathsf{N}-\mathsf{y}}{2}=\mathsf{N}-\mathsf{M}$ pairs to $\mathsf{N}-\mathsf{M}$ servers where each pair is assigned $\frac{\mathsf{M}}{2}$ servers and each server obtains $\frac{\mathsf{M}}{2}$ pairs.

Computing phase. We first focus on the servers in $[\mathsf{M}]$. We let the servers in $\left[\frac{\mathsf{M}}{2}\right]$ with the same datasets compute a same linear combination of messages, which is denoted by A_1. Similarly, we let the servers in $\left[\frac{\mathsf{M}}{2}+1:\mathsf{M}\right]$ with the same datasets compute a same linear combination of messages, which is denoted by A_2. We construct A_1 and A_2 such that from A_1 and A_2, we can recover the following two linear combinations

$$F_1=W_1+\cdots+W_{\mathsf{N}}; \tag{23a}$$

$$F_2=W_{\mathsf{y}+1}+\cdots+W_{\mathsf{M}}+2(W_{\mathsf{M}+1}+\cdots+W_{\mathsf{N}}). \tag{23b}$$

This can be done by letting

$$A_1=2F_1-F_2=2(W_1+\cdots+W_{\mathsf{y}})+W_{\mathsf{y}+1}+\cdots+W_{\mathsf{M}}$$

which can be computed by servers in $\left[\frac{\mathsf{M}}{2}\right]$, and letting

$$A_2=F_2-F_1=-(W_1+\cdots+W_{\mathsf{y}})+W_{\mathsf{M}+1}+\cdots+W_{\mathsf{N}}$$

which can be computed by servers in $\left[\frac{\mathsf{M}}{2}+1:\mathsf{M}\right]$.

We then focus on the servers in $[\mathsf{M}+1:\mathsf{N}]$. For each pair of datasets $\mathcal{P}_i=\{\mathsf{y}+i,\mathsf{M}+i\}$ where $i\in[\mathsf{N}-\mathsf{M}]$, we let $P_i=W_{\mathsf{y}+i}+2W_{\mathsf{M}+i}$. Hence, we can express F_2 in (23b) as $P_1+\cdots+P_{\mathsf{N}-\mathsf{M}}$. Next we apply the computing phase of the proposed scheme for the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem. In the proposed scheme for the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem, server $n'\in[\mathsf{N}-\mathsf{M}]$ computes a linear combination of the $\mathsf{N}-\mathsf{M}$ messages, where the coefficients of the messages that server n' cannot compute are 0. We assume that the vector of the coefficients in this linear combination is $\mathbf{v}_{n'}$, containing $\mathsf{N}-\mathsf{M}$ elements.

Go back to the (N,M) non-secure problem. We let each server $n\in[\mathsf{M}+1:\mathsf{N}]$ compute

$$A_{n-\mathsf{M}+2}=\mathbf{v}_{n-\mathsf{M}}\ [P_1;\ldots;P_{\mathsf{N}-\mathsf{M}}],$$

where $\mathbf{v}_{n-\mathsf{M}}$ represents the vector of the coefficients in the linear combination sent by server $n-\mathsf{M}$ in the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem.

1) Decoding Phase: If the set of $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1$ responding servers contains one server in $\left[\frac{\mathsf{M}}{2}\right]$ and one server in $\left[\frac{\mathsf{M}}{2}+1:\mathsf{M}\right]$, from A_1 and A_2 the user can recover the task function.

We then consider the case where from the answers of the responding servers, the user can only receive one of A_1 and A_2. In this case, the set of N_{r} responding servers contains at least $\mathsf{N}_{\mathrm{r}}-\frac{\mathsf{M}}{2}=\mathsf{N}-\frac{3\mathsf{M}}{2}+1$ servers in $[\mathsf{M}+1:\mathsf{N}]$. Notice that in the $\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$ non-secure problem, the answers of any $\mathsf{N}-\mathsf{M}-\frac{\mathsf{M}}{2}+1=\mathsf{N}-\frac{3\mathsf{M}}{2}+1$ servers can re-construct the task function. Hence, in the (N,M) non-secure problem, the answers of any $\mathsf{N}-\frac{3\mathsf{M}}{2}+1$ servers in $[\mathsf{M}+1:\mathsf{N}]$ can re-construct $P_1+\cdots+P_{\mathsf{N}-\mathsf{M}}=F_2$. Together with $A_1=2F_1-F_2$ or with $A_2=F_2-F_1$, the user can recover the task function F_1.

It can be seen that the number of linearly independent combinations transmitted by servers in $[\mathsf{M}+1:\mathsf{N}]$ is $h\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)$, the linear space of which contains F_2. In addition, the number of linearly independent combinations transmitted by servers in $[\mathsf{M}]$ is two, the linear space of which also contains F_2. Hence, the number of totally transmitted linearly independent combinations is $h(\mathsf{N},\mathsf{M})=2+h\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)-1=h\left(\mathsf{N}-\mathsf{M},\frac{\mathsf{M}}{2}\right)+1$, coinciding with (19).

D. Scheme 4 for (20)

We first provide an example to illustrate the main idea.

Example 3: We consider the $(\mathsf{N},\mathsf{M})=(8,5)$ *non-secure problem. It can be seen that in this example* $\mathsf{N}_{\mathrm{r}}=\mathsf{N}-\mathsf{M}+1=4$.

Assignment phase. We assign the datasets as follows.

Server 1	*Server 2*	*Server 3*	*Server 4*
D_1	D_1	D_1	D_1
D_2	D_2	D_2	D_2
D_3	D_3	D_6	D_6
D_4	D_4	D_7	D_7
D_5	D_5	D_8	D_8
Server 5	*Server 6*	*Server 7*	*Server 8*
D_1	D_3	D_3	D_3
D_2	D_4	D_4	D_4
D_6	D_5	D_5	D_5
D_7	D_6	D_7	D_8
D_8	D_7	D_8	D_6

Computing phase. We let each user send one linear combination of messages, such that the user can recover $\mathbf{F}\ [W_1;\ldots;W_{\mathsf{N}}]$ *from the answers of any* N_{r} *responding servers, where*

$$\begin{aligned}\mathbf{F}=\begin{bmatrix}\mathbf{f}_1\\ \mathbf{f}_2\\ \mathbf{f}_3\end{bmatrix}&=\begin{bmatrix}1&1&1&1&1&1&1&1\\ 0&0&2&2&2&1&1&1\\ 0&0&0&0&0&*&*&*\end{bmatrix}\\&=\begin{bmatrix}1&1&1&1&1&1&1&1\\ 0&0&2&2&2&1&1&1\\ 0&0&0&0&0&1&2&3\end{bmatrix},\end{aligned} \tag{24}$$

and each '$*$*' is uniformly i.i.d. over* $\mathbb{F}_{\mathsf{q}}$ *and in this example we assume that the last three '*$*$*' in* $\mathbf{f}_3$ *are simply* 1, 2, *and* 3, *respectively. We also define that* $[F_1;F_2;F_3]=\mathbf{F}\ [W_1;\ldots;W_{\mathsf{N}}]$.

We let servers $1,2$ *with datasets* $D_1,\ldots,D_5$ *compute* $X_1=X_2=F_1-F_2=W_1+W_2-W_3-W_4-W_5$.

For servers in $[3:5]$ *with datasets* D_1,D_2,D_6,D_7,D_8, *we construct their transmissions such that from the answers of any two of them we can recover* $2F_1-F_2=2W_1+2W_2+W_6+W_7+W_8$ *and* $F_3=2W_6+W_7$. *Notice that both of* $2F_1-F_2$ *and* F_3 *can be computed by each server in* $[3:5]$. *Hence, each server in* $[3:5]$ *computes a random linear combination of* $(2F_1-F_2)$ *and* F_3. *For example, we let servers* $3,4,5$ *compute* X_3,X_4,X_5, *respectively, where* $X_3=(2F_1-F_2)+F_3$; $X_4=(2F_1-F_2)+2F_3$; $X_5=(2F_1-F_2)+4F_3$.

For servers in $[6:8]$, *we construct their transmissions such that from the answers of any two of them we can recover* F_2 *and* F_3. *This can be done by letting servers* $6,7,8$ *compute* X_6,X_7,X_8, *respectively, where* $X_6=3F_2-F_3=6W_3+6W_4+6W_5+2W_6+W_7$; $X_7=F_2-F_3=2W_3+2W_4+2W_5-W_7-2W_8$; $X_8=2F_2-F_3=4W_3+4W_4+4W_5+W_6-W_8$.

Decoding phase. For any set of $\mathsf{N_r}=4$ *servers,* $\mathcal{A}$*, we are in one of the following three cases:*

- *Case 1:* $\mathcal{A}$ *contains at least two servers in* $[3:5]$*. From the answers of any two servers in* $[3:5]$*, the user can recover* $2F_1-F_2$ *and* F_3*. Besides,* $\mathcal{A}$ *contains at least either one server in* $[2]$ *or one server in* $[6:8]$*. It can be seen that each of* X_1,X_2,X_6,X_7,X_8 *is linearly independent of* $2F_1-F_2$ *and* F_3*. Hence, the user then recovers* F_1*.*
- *Case 2:* $\mathcal{A}$ *contains at least two servers in* $[6:8]$*. From the answers of any two servers in* $[6:8]$*, the user can recover* F_2 *and* F_3*. Besides,* $\mathcal{A}$ *contains at least one server in* $[5]$*. It can be seen that in the transmitted linear combination of each server in* $[5]$ *contains* F_1*. Hence, the user then recovers* F_1*.*
- *Case 3:* $\mathcal{A}$ *contains servers* $1,2$*, one server in* $[3:5]$*, and one server in* $[6:8]$*. In this case, we can also check that the user receives three independent linear combinations in* F_1,F_2,F_3*, such that it can recover* F_1*.*

It can be seen that the number of totally transmitted linearly independent combinations is $h(8,5)=3$*, coinciding with* (20). □

We now consider the (N,M) non-secure problem where $1.5\mathsf{M}\le\mathsf{N}<2\mathsf{M}$ and M is odd, and aim to construct a scheme (Scheme 4) to prove (20). We also define that $\mathsf{N}=2\mathsf{M}-\mathsf{y}$.

1) Assignment Phase: The assignment is given at the top of the next page. In the assignment, we divide the N datasets into three parts, where the first part contains $D_1,\ldots,D_t$ (later we will explain the reason to choose $t=\frac{\mathsf{M}-1}{2}$) which are all assigned to servers in $[\mathsf{M}]$; the second part contains $D_{t+1},\ldots,D_{\mathsf{M}}$ which are all assigned to servers in $[\mathsf{y}]\cup[\mathsf{M}+1:\mathsf{N}]$; the third part contains $D_{\mathsf{M}+1},\ldots,D_{\mathsf{N}}$, which are assigned to servers in $[\mathsf{y}+1:\mathsf{M}]$ in a cyclic way where each server obtains $\mathsf{M}-t$ neighbouring datasets in $\left[\frac{\mathsf{M}-1}{2}+1:\mathsf{N}\right]$. The datasets $D_{\mathsf{M}+1},\ldots,D_{\mathsf{N}}$ are also assigned to servers in $[\mathsf{M}+1:\mathsf{N}]$ in a cyclic way where each server obtains t neighbouring datasets in $\left[\frac{\mathsf{M}-1}{2}+1:\mathsf{N}\right]$.

As we assign the datasets in $[\mathsf{M}+1:\mathsf{N}]$ to the servers in $[\mathsf{y}+1:\mathsf{M}]$ in a cyclic way where each server obtains $\mathsf{M}-t$ datasets, we can choose $\mathsf{N}-\mathsf{M}-(\mathsf{M}-t)+1=\mathsf{N}-2\mathsf{M}+t+1$ neighbouring servers in $[\mathsf{y}+1:\mathsf{M}]$ satisfying the constraint in (8); in addition, server 1 has $D_{t+1},\ldots,D_{\mathsf{M}}$, which are not assigned to the servers in $[\mathsf{y}+1:\mathsf{M}]$. Hence, the ordered set of the above $\mathsf{N}-2\mathsf{M}+t+2$ servers satisfies the constraint in (8).

Similarly, we assign the datasets in $[\mathsf{M}+1:\mathsf{N}]$ to the servers in $[\mathsf{M}+1:\mathsf{N}]$ in a cyclic way where each server obtains t datasets, we can choose $\mathsf{N}-\mathsf{M}-t+1$ neighbouring servers in $[\mathsf{M}+1:\mathsf{N}]$ satisfying the constraint in (8); in addition, server 1 has $D_1,\ldots,D_t$, which are not assigned to the servers in $[\mathsf{M}+1:\mathsf{N}]$. Hence, the ordered set of the above $\mathsf{N}-\mathsf{M}-t+2$ servers satisfies the constraint in (8).

Similar to the derivation of (34c), by the chain rule of entropy, under the above assignment,

$$H(X_{[\mathsf{N}]})/\mathsf{L}\ \ge\ \max\{\mathsf{N}-2\mathsf{M}+t+2,\mathsf{N}-\mathsf{M}-t+2\}\overset{t=\frac{\mathsf{M}-1}{2}}{=}\mathsf{N}-\mathsf{M}-\frac{\mathsf{M}-1}{2}+2=\frac{\mathsf{M}+5}{2}-y.$$

Hence, we let $t=\frac{\mathsf{M}-1}{2}$.

2) Computing Phase: We design the computing phase such that the total number of independent transmitted linear combinations of messages by all servers is $\frac{\mathsf{M}+5}{2}-\mathsf{y}$. These linear combinations are in $\mathbf{F}\ [W_1;\ldots;W_{\mathsf{N}}]$ where

$$\mathbf{F}=\begin{bmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}\end{bmatrix}=\underbrace{\begin{bmatrix}1,\ldots,1\\ 0,\ldots,0\\ 0,\ldots,0\\ \ddots\\ 0,\ldots,0\end{bmatrix}}_{\mathbf{F}_1}\underbrace{\begin{bmatrix}1,\ldots,1\\ a,\ldots,a\\ 0,\ldots,0\\ \ddots\\ 0,\ldots,0\end{bmatrix}}_{\mathbf{F}_2}\underbrace{\begin{bmatrix}1,\ldots,1\\ 1,\ldots,1\\ *,\ldots,*\\ \ddots\\ *,\ldots,*\end{bmatrix}}_{\mathbf{F}_3}.$$

Notice that a represents a symbol uniformly over $\mathbb{F}_{\mathsf{q}}\backslash\{0,1\}$, and '$*$' represents a symbol uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. We divide matrix $\mathbf{F}$ into three column-wise sub-matrices, $\mathbf{F}_1$ with dimension $\left(\frac{\mathsf{M}+5}{2}-\mathsf{y}\right)\times\frac{\mathsf{M}-1}{2}$ which corresponds to the messages in $\left[\frac{\mathsf{M}-1}{2}\right]$, $\mathbf{F}_2$ with dimension $\left(\frac{\mathsf{M}+5}{2}-\mathsf{y}\right)\times\frac{\mathsf{M}+1}{2}$ which corresponds to the messages in $\left[\frac{\mathsf{M}+1}{2}:\mathsf{M}\right]$, and $\mathbf{F}_3$ with dimension $\left(\frac{\mathsf{M}+5}{2}-\mathsf{y}\right)\times(\mathsf{N}-\mathsf{M})$ which corresponds to the messages in $[\mathsf{M}+1:\mathsf{N}]$. We also define that $F_i=\mathbf{f}_i[W_1;\ldots;W_{\mathsf{N}}]$ for each $i\in\left[\frac{\mathsf{M}+5}{2}-\mathsf{y}\right]$. Thus the transmission of each server could be expressed as a linear combination of $\left[F_1;\ldots;F_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}\right]$.

As each server $n\in[\mathsf{y}]$ cannot compute $W_{\mathsf{M}+1},\ldots,W_{\mathsf{N}}$, we let it compute

$$\begin{aligned}&\mathbf{s}_n\ \mathbf{F}\ [W_1;\ldots;W_{\mathsf{N}}]\\ &=[1,-1,0,\ldots,0]\ \mathbf{F}\ [W_1;\ldots;W_{\mathsf{N}}] &\text{(25a)}\\ &=W_1+\cdots+W_{\frac{\mathsf{M}-1}{2}}+(1-a)(W_{\frac{\mathsf{M}+1}{2}}+\cdots+W_{\mathsf{M}}), &\text{(25b)}\end{aligned}$$

such that the coefficients of $W_{\mathsf{M}+1},\ldots,W_{\mathsf{N}}$ which it cannot compute are 0.

For the servers in $[\mathsf{y}+1:\mathsf{M}]$, we construct their transmissions such that from the answers of any $\frac{\mathsf{M}+3}{2}-\mathsf{y}$ servers in $[\mathsf{y}+1:\mathsf{M}]$, the user can recover $aF_1-F_2,F_3,\ldots,F_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}$. More precisely, we let server $n\in[\mathsf{y}+1:\mathsf{M}]$ compute

$$\mathbf{s}_n\left[a\mathbf{f}_1-\mathbf{f}_2;\mathbf{f}_3;\ldots;\mathbf{f}_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}\right][W_1;\ldots;W_{\mathsf{N}}], \tag{26a}$$

where

$$\left[a\mathbf{f}_1-\mathbf{f}_2;\mathbf{f}_3;\ldots;\mathbf{f}_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}\right]=\begin{bmatrix}a&\ldots&a&0&\ldots&0&a-1&\ldots&a-1\\ 0&\ldots&0&0&\ldots&0&*&\ldots&*\\ &\ddots&&&\ddots&&&\ddots&\\ 0&\ldots&0&0&\ldots&0&*&\ldots&*\end{bmatrix}. \tag{26b}$$

$(\mathbf{F}'_3)_{\left(\frac{\mathsf{M}+3}{2}-\mathsf{y}\right)\times(\mathsf{N}-\mathsf{M})}$

We design $\mathbf{s}_n$ as follows. Notice that $W_1,\ldots,W_{\frac{\mathsf{M}-1}{2}}$ can be computed by server n; and that in the linear combination (26a) the coefficients of $W_{\frac{\mathsf{M}-1}{2}+1},\ldots,W_{\mathsf{M}}$ are 0. Hence, in order to guarantee that in (26a) the coefficients of the messages which server n cannot compute are 0, we only need to consider the messages in $W_{\mathsf{M}+1},\ldots,W_{\mathsf{N}}$, whose related columns are in $\mathbf{F}'_3$. Server n cannot compute $\mathsf{N}-\mathsf{M}-\frac{\mathsf{M}+1}{2}=\frac{\mathsf{M}-1}{2}-\mathsf{y}$ messages in $W_{\mathsf{M}+1},\ldots,W_{\mathsf{N}}$; thus the column-wise sub-matrix of $\mathbf{F}'_3$

Server 1	$\cdots$	Server y	Server y $+1$	Server y $+2$	$\cdots$	Server M	Server M+1	Server M $+2$	$\cdots$	Server N
D_1	$\cdots$	D_1	D_1	D_1	$\cdots$	D_1	$D_{\frac{\mathsf{M}-1}{2}+1}$	$D_{\frac{\mathsf{M}-1}{2}+1}$	$\cdots$	$D_{\frac{\mathsf{M}-1}{2}+1}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$D_{\frac{\mathsf{M}-1}{2}}$	$\cdots$	$D_{\frac{\mathsf{M}-1}{2}}$	$D_{\frac{\mathsf{M}-1}{2}}$	$D_{\frac{\mathsf{M}-1}{2}}$	$\cdots$	$D_{\frac{\mathsf{M}-1}{2}}$	D_{M}	D_{M}	$\cdots$	D_{M}
$D_{\frac{\mathsf{M}-1}{2}+1}$	$\cdots$	$D_{\frac{\mathsf{M}-1}{2}+1}$	$D_{\mathsf{M}+1}$	$D_{\mathsf{M}+2}$	$\cdots$	D_{N}	$D_{\mathsf{M}+1}$	$D_{\mathsf{M}+2}$	$\cdots$	D_{N}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
D_{M}	$\cdots$	D_{M}	$D_{\frac{3\mathsf{M}+1}{2}}$	$D_{\frac{3\mathsf{M}+1}{2}+1}$	$\cdots$	$D_{\frac{3\mathsf{M}+1}{2}-1}$	$D_{\frac{3\mathsf{M}-1}{2}}$	$D_{\frac{3\mathsf{M}-1}{2}+1}$	$\cdots$	$D_{\frac{3\mathsf{M}-1}{2}-1}$

corresponding to these $\frac{\mathsf{M}-1}{2}-\mathsf{y}$ messages has the dimension $\left(\frac{\mathsf{M}+3}{2}-\mathsf{y}\right)\times\left(\frac{\mathsf{M}-1}{2}-\mathsf{y}\right)$. In addition, each '$*$' is uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. Hence, the left-hand side nullspace of this sub-matrix contains $\frac{\mathsf{M}+3}{2}-\mathsf{y}-\left(\frac{\mathsf{M}-1}{2}-\mathsf{y}\right)=2$ vectors, each of which has $\frac{\mathsf{M}+3}{2}-\mathsf{y}$ elements. We let $\mathbf{s}_n$ be a random linear combination of these two vectors, where each of the two coefficients is uniformly i.i.d over $\mathbb{F}_\mathsf{q}$.

Finally, we focus on the servers in $[\mathsf{M}+1:\mathsf{N}]$. We construct their transmissions such that from any the answers of any $\frac{\mathsf{M}+3}{2}-\mathsf{y}$ servers in $[\mathsf{M}+1:\mathsf{N}]$, the user can recover $F_2,F_3,\ldots,F_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}$. More precisely, we let server $n\in[\mathsf{M}+1:\mathsf{N}]$ compute

$$\mathbf{s}_n\left[\mathbf{f}_2;\mathbf{f}_3;\ldots;\mathbf{f}_{\frac{\mathsf{M}+5}{2}-\mathsf{y}}\right][W_1;\ldots;W_\mathsf{N}]. \tag{27}$$

We design $\mathbf{s}_n$ as follows. Notice that in the linear combination (27) the coefficients of $W_1,\ldots,W_{\frac{\mathsf{M}-1}{2}}$ are 0; and that $W_{\frac{\mathsf{M}+1}{2}},\ldots,W_\mathsf{M}$ can be computed by server n. Hence, in order to guarantee that in (27) the coefficients of the messages which server n cannot compute are 0, we only need to consider the messages in $W_{\mathsf{M}+1},\ldots,W_\mathsf{N}$, whose related columns are in $\mathbf{F}_3^{([2:\frac{\mathsf{M}+5}{2}-y])_\mathrm{r}}$.[6] Server n cannot compute $\mathsf{N}-\mathsf{M}-\frac{\mathsf{M}-1}{2}=\frac{\mathsf{M}+1}{2}-\mathsf{y}$ messages in $W_{\mathsf{M}+1},\ldots,W_\mathsf{N}$; thus the column-wise sub-matrix of $\mathbf{F}_3^{([2:\frac{\mathsf{M}+5}{2}-y])_\mathrm{r}}$ corresponding to these $\frac{\mathsf{M}+1}{2}-\mathsf{y}$ messages has the dimension $\left(\frac{\mathsf{M}+3}{2}-\mathsf{y}\right)\times\left(\frac{\mathsf{M}+1}{2}-\mathsf{y}\right)$. In addition, each '$*$' is uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. Hence, the left-hand side nullspace of this sub-matrix contains $\frac{\mathsf{M}+3}{2}-\mathsf{y}-\left(\frac{\mathsf{M}+1}{2}-\mathsf{y}\right)=1$ vector, which has $\frac{\mathsf{M}+3}{2}-\mathsf{y}$ elements. We let $\mathbf{s}_n$ be this vector.

Due to the limitation of the pages, we skip the proof of the decodability and please refer to [15, Section IV-D] for further details.

By the above scheme, the number of linearly independent transmissions by all servers is equal to the number of rows in $\mathbf{F}$, i.e., $\frac{\mathsf{M}+5}{2}-\mathsf{y}=\mathsf{N}-\frac{3\mathsf{M}-5}{2}$, coinciding with (20).

E. Scheme 5 for (21)

Finally, we consider the case where $\mathsf{M}<\mathsf{N}<1.5\mathsf{M}$, and aim to construct a scheme (Scheme 5) to prove (21). Scheme 5 is a recursive scheme which is based on the proposed scheme for the $(\mathsf{M},2\mathsf{M}-\mathsf{N})$ non-secure problem. We assume that the latter scheme has been designed before, whose number of totally transmitted linearly independent combinations of messages is $h(\mathsf{M},2\mathsf{M}-\mathsf{N})$.

[6] Recall that $\mathbf{M}^{(\mathcal{S})_\mathrm{r}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the rows of $\mathbf{M}$ with indices in $\mathcal{S}$.

1) Assignment Phase: We first assign datasets $D_1,\ldots,D_{\mathsf{N}-\mathsf{M}}$ to each server in $[\mathsf{M}]$. Then we assign datasets $D_{\mathsf{N}-\mathsf{M}+1},\ldots,D_\mathsf{N}$ to each server in $[\mathsf{M}+1:\mathsf{N}]$. So far, each server in $[\mathsf{M}+1:\mathsf{N}]$ has obtained M datasets, while each server in $[\mathsf{M}]$ has obtained $\mathsf{N}-\mathsf{M}<\mathsf{M}$ datasets. In addition, each dataset in $[\mathsf{N}-\mathsf{M}+1:\mathsf{N}]$ has been assigned to $\mathsf{N}-\mathsf{M}<\mathsf{M}$ servers. Hence, in the next step we should assign each dataset D_k where $k\in[\mathsf{N}-\mathsf{M}+1:\mathsf{N}]$ to $\mathsf{M}-(\mathsf{N}-\mathsf{M})=2\mathsf{M}-\mathsf{N}$ servers in $[\mathsf{M}]$, such that each server in $[\mathsf{M}]$ obtains $\mathsf{M}-(\mathsf{N}-\mathsf{M})=2\mathsf{M}-\mathsf{N}$ datasets in $[\mathsf{N}-\mathsf{M}+1:\mathsf{N}]$. Thus we can apply the assignment phase of the proposed scheme for the $(\mathsf{M},2\mathsf{M}-\mathsf{N})$ non-secure problem, to assign datasets $D_{\mathsf{N}-\mathsf{M}+1},\ldots,D_\mathsf{N}$ to servers in $[\mathsf{M}]$.

2) Computing Phase: Let us first focus on the $(\mathsf{M},2\mathsf{M}-\mathsf{N})$ non-secure problem, where the M messages are assumed to be $W_1'',\ldots,W_\mathsf{M}''$. In the proposed scheme for the $(\mathsf{M},2\mathsf{M}-\mathsf{N})$ non-secure problem, each server computes a linear combination of the M messages. Considering the transmitted linear combinations by all servers, the number of linearly independent combinations is denoted by $h(\mathsf{M},2\mathsf{M}-\mathsf{N})$ and these $h(\mathsf{M},2\mathsf{M}-\mathsf{N})$ linear combinations can be expressed as

$$\mathbf{F}_4\ [W_1'';\ldots;W_\mathsf{M}'']. \tag{28}$$

The transmission of each server $n'\in[\mathsf{M}]$ can be expressed as

$$\mathbf{s}_{n'}\ \mathbf{F}_4\ [W_1'';\ldots;W_\mathsf{M}''].$$

Let us then go back to the (N,M) non-secure problem. We construct the answer of the N servers, such that the transmissions of all servers totally contain $h(\mathsf{M},2\mathsf{M}-\mathsf{N})$ linearly independent combinations and these $h(\mathsf{M},2\mathsf{M}-\mathsf{N})$ linear combinations can be expressed as $\mathbf{F}[W_1;\ldots;W_\mathsf{N}]$, where (each a_i where $i\in[h(\mathsf{M},2\mathsf{M}-\mathsf{N})-1]$ represents a symbol uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$)

$$\mathbf{F}=\left[\begin{array}{ccc|ccc} 1 & \cdots & 1 & 1 & \cdots & 1 \\ a_1 & \cdots & a_1 & + & \cdots & + \\ & \ddots & & & \ddots & \\ a_{h(\mathsf{M},2\mathsf{M}-\mathsf{N})-1} & \cdots & a_{h(\mathsf{M},2\mathsf{M}-\mathsf{N})-1} & + & \cdots & + \end{array}\right].$$

$\mathbf{F}_5$ (left block) $\quad$ $\mathbf{F}_4$ in (28) (right block)

Each '$+$' represents an element of $\mathbf{F}_4$ in (28). Notice that the dimension of $\mathbf{F}_5$ is $\left(h(\mathsf{M},2\mathsf{M}-\mathsf{N})-1\right)\times(\mathsf{N}-\mathsf{M})$ and the dimension of $\mathbf{F}_4$ is $\left(h(\mathsf{M},2\mathsf{M}-\mathsf{N})-1\right)\times\mathsf{M}$.

For each server $n\in[\mathsf{M}]$, by the construction of the assignment phase, datasets $D_1,\ldots,D_{\mathsf{N}-\mathsf{M}}$ are assigned to it and the assignment on the datasets $D_{\mathsf{N}-\mathsf{M}+1},\ldots,D_\mathsf{N}$ is

from the assignment phase of the proposed scheme for the $(\mathsf{M}, 2\mathsf{M}-\mathsf{N})$ non-secure problem. Hence, we let server n compute $\mathbf{s}_n\mathbf{F}[W_1;\ldots;W_{\mathsf{N}}]$, where $\mathbf{s}_n$ is the same as the transmission vector of server n in the $(\mathsf{M}, 2\mathsf{M}-\mathsf{N})$ non-secure problem.

For each server $n \in [\mathsf{M}+1:\mathsf{N}]$, it cannot compute $W_1,\ldots,W_{\mathsf{N}-\mathsf{M}}$, which correspond to the column-wise sub-matrix $\mathbf{F}_5$, whose rank is 1. So the left-hand side null space of $\mathbf{F}_5$ contains $h(\mathsf{M}, 2\mathsf{M}-\mathsf{N})-1$ linearly independent vectors, each of which has $h(\mathsf{M}, 2\mathsf{M}-\mathsf{N})$ elements. We let the transmission vector of server n, denoted by $\mathbf{s}_n$, be a random linear combinations of these $h(\mathsf{M}, 2\mathsf{M}-\mathsf{N})-1$ linearly independent vectors, where the $h(\mathsf{M}, 2\mathsf{M}-\mathsf{N})-1$ coefficients are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$; in other words, server n computes $\mathbf{s}_n\mathbf{F}[W_1;\ldots;W_{\mathsf{N}}]$.

Due to the limitation of the pages, we skip the proof of the decodability and please refer to [15, Section IV-E] for further details.

In conclusion, we have $h(\mathsf{N},\mathsf{M}) = h(\mathsf{M}, 2\mathsf{M}-\mathsf{N})$, which coincides with (21).

V. Conclusion

We formulated the secure distributed linearly separable computation problem, where the user should recover the desired task function without retrieving any other information about the datasets. It is interesting to see that to add this security constraint into a non-secure computing scheme, we do not need to increase the communication cost if the original computing scheme is based on linear coding. We then focused on the problem where $\mathsf{K}_{\mathrm{c}} = 1$ and the computation cost is minimum. In this case, while achieving the optimal communication cost, we aim to minimize the size of the randomness variable which is independent of the datasets and is introduced in the system to preserve the security. For this purpose, we proposed an information theoretic converse bound on the randomness size for each possible assignment. A novel secure computing scheme was proposed, which outperforms the optimal computing schemes with the well-know fractional repetition assignment and cyclic assignment in terms of the randomness size and leads some additional optimality results.

Appendix A
Proof of Theorem 1

We consider the non-sercure distributed linearly separable computation problem in [6] where $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+\mathsf{m})$ for $\mathsf{m} \in [\mathsf{N}_{\mathrm{r}}]$ and the user requests $\mathsf{K}_{\mathrm{c}} \in [\mathsf{K}]$ linearly independent combinations of messages. We now describe on a general linear coding computing scheme $\mathscr{S}$. In this scheme, we divide each message W_k where $k \in [\mathsf{K}]$ into ℓ non-overlapping and equal-length sub-messages, $W_k = \{W_{k,i} : i \in [\ell]\}$. Server $n \in [\mathsf{N}]$ sends $\frac{\ell T_n}{\mathsf{L}}$ linearly independent combinations of $W_{1,1}, W_{1,2}, \ldots, W_{\mathsf{K},\ell}$.

Considering the transmitted linear combinations by all servers, the number of linearly independent combinations is denoted by λ and these λ linear combinations can be expressed as $\mathbf{F}[W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$, where the i^{th} row of $\mathbf{F}$ is $\mathbf{f}_i$, for each $i \in [\lambda]$. Notice that any linear scheme can be transformed in the above manner. Among these λ linear combinations, $\mathbf{f}_i[W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$ where $i \in [\ell\mathsf{K}_{\mathrm{c}}]$ represent the desired task function of the user. The transmission of each server $n \in [\mathsf{N}]$ can be express as

$$\mathbf{S}_n \ \mathbf{F}\ [W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}], \tag{29}$$

where the dimension of $\mathbf{S}_n$ is $\frac{\ell T_n}{\mathsf{L}} \times \lambda$.

For any set of N_{r} responding servers (denoted by $\mathcal{A} = \{\mathcal{A}(1),\ldots,\mathcal{A}(\mathsf{N}_{\mathrm{r}})\}$), the user receives $\mathbf{S}_{\mathcal{A}}\ \mathbf{F}\ [W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$ where $\mathbf{S}_{\mathcal{A}}$ represents the row-wise sub-matrix of $\left[\mathbf{S}_{\mathcal{A}(1)};\ldots;\mathbf{S}_{\mathcal{A}(\mathsf{N}_{\mathrm{r}})}\right]$ which has the same rank as $\left[\mathbf{S}_{\mathcal{A}(1)};\ldots;\mathbf{S}_{\mathcal{A}(\mathsf{N}_{\mathrm{r}})}\right]$. Assume $\mathbf{S}_{\mathcal{A}}$ contains $r_{\mathcal{A}}$ rows. In the decoding phase, the user multiplies $\mathbf{S}_{\mathcal{A}}\mathbf{F}[W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$ by $\mathbf{D}_{\mathcal{A}}$ with dimension $\ell\mathsf{K}_{\mathrm{c}} \times r_{\mathcal{A}}$ where

$$\mathbf{D}_{\mathcal{A}}\ \mathbf{S}_{\mathcal{A}} = [\mathbf{I}_{\ell\mathsf{K}_{\mathrm{c}}}, \mathbf{0}_{\ell\mathsf{K}_{\mathrm{c}}\times(r_{\mathcal{A}}-\ell\mathsf{K}_{\mathrm{c}})}], \tag{30}$$

$\mathbf{I}_n$ represents the identity matrix with dimension $n \times n$, and $\mathbf{0}_{m\times n}$ represents the zero matrix with dimension $m \times n$. Hence, the user can recover the desired task function from $\mathbf{D}_{\mathcal{A}}\mathbf{S}_{\mathcal{A}}\mathbf{F}[W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$.

Now we take the security constraint (7) into consideration, and extend the above general linear coding scheme. We introduce $\lambda - \ell\mathsf{K}_{\mathrm{c}}$ independent randomness variables $Q_1,\ldots,Q_{\lambda-\ell\mathsf{K}_{\mathrm{c}}}$, where $Q_i, i \in [\lambda-\ell\mathsf{K}_{\mathrm{c}}]$, is uniformly i.i.d. over $[\mathbb{F}_{\mathsf{q}}]^{\frac{\mathsf{L}}{\ell}}$. We then let $\mathbf{F}' = [\mathbf{F},\mathbf{S}]$, where $\mathbf{S} = [\mathbf{0}_{\ell\mathsf{K}_{\mathrm{c}}\times(\lambda-\ell\mathsf{K}_{\mathrm{c}})};\mathbf{S}']$ and $\mathbf{S}'$ is full-rank with dimension $(\lambda-\ell\mathsf{K}_{\mathrm{c}})\times(\lambda-\ell\mathsf{K}_{\mathrm{c}})$.

We let each server $n \in [\mathsf{N}]$ transmit

$$\mathbf{S}_n\ \mathbf{F}'\ [W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell};Q_1;\ldots;Q_{\lambda-\ell\mathsf{K}_{\mathrm{c}}}], \tag{31}$$

where $\mathbf{S}_n$ is the same as that in (29). It can be seen that in the transmitted linear combinations (31), the coefficients of the sub-messages which server n cannot compute are still 0 as the original non-secure scheme.

For any set of N_{r} responding servers $\mathcal{A}$, the user receives $\mathbf{S}_{\mathcal{A}}\ \mathbf{F}'\ [W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell}]$, and recovers its desired task from $\mathbf{D}_{\mathcal{A}}\mathbf{S}_{\mathcal{A}}\mathbf{F}'[W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell};Q_1;\ldots;Q_{\lambda-\ell\mathsf{K}_{\mathrm{c}}}]$, since (30) holds.

Finally, we will prove that the above scheme is secure. From the answers of all servers, the user can only recover totally λ linearly independent combinations, which are

$$\mathbf{F}'\ [W_{1,1};W_{1,2};\ldots;W_{\mathsf{K},\ell};Q_1;\ldots;Q_{\lambda-\ell\mathsf{K}_{\mathrm{c}}}]. \tag{32}$$

In addition, $\mathbf{S}'$ is full-rank (with rank equal to $\lambda-\ell\mathsf{K}_{\mathrm{c}}$). Hence, the user can only recover the desired task function (i.e., the first $\ell\mathsf{K}_{\mathrm{c}}$ linear combinations in (32)) without $Q_1,\ldots,Q_{\lambda-\ell\mathsf{K}_{\mathrm{c}}}$, and thus the above scheme is secure. We introduce $\lambda-\ell\mathsf{K}_{\mathrm{c}}$ randomness variables, each of which has $\frac{\mathsf{L}}{\ell}$ symbols. Hence, the total needed randomness size is $\eta = \frac{(\lambda-\ell\mathsf{K}_{\mathrm{c}})\mathsf{L}/\ell}{\mathsf{L}} = \frac{\lambda}{\ell} - \mathsf{K}_{\mathrm{c}} = H_{\mathscr{S}} - \mathsf{K}_{\mathrm{c}}$.

Appendix B
Proof of Theorem 3

By the security constraint in (7), the user can only obtain $W_1+\cdots+W_{\mathsf{K}}$ without knowing any other information about the messages after receiving the answers of all servers. Recall

that $X_{\mathcal{S}} = \{X_n : n \in \mathcal{S}\}$. Intuitively by [20], we need a key with length at least $H(X_{[\mathsf{N}]}) - H(W_1 + \cdots + W_{\mathsf{K}})$ such that except $W_1 + \cdots + W_{\mathsf{K}}$, the other information about $W_1, \ldots, W_{\mathsf{K}}$ transmitted in $X_{[\mathsf{N}]}$ is hidden. More precisely, from (7) we have

$$\begin{aligned} 0 &= I\big(W_1, \ldots, W_{\mathsf{K}}; X_{[\mathsf{N}]} | W_1 + W_2 + \cdots + W_{\mathsf{K}}\big) && \text{(33a)} \\ &= H(X_{[\mathsf{N}]} | W_1 + W_2 + \cdots + W_{\mathsf{K}}) - H(X_{[\mathsf{N}]} | W_1, \ldots, W_{\mathsf{K}}) && \text{(33b)} \\ &\geq H(X_{[\mathsf{N}]} | W_1 + W_2 + \cdots + W_{\mathsf{K}}) \\ &\quad - H(Q, W_1, \ldots, W_{\mathsf{K}} | W_1, \ldots, W_{\mathsf{K}}) && \text{(33c)} \\ &= H(X_{[\mathsf{N}]} | W_1 + W_2 + \cdots + W_{\mathsf{K}}) - H(Q) && \text{(33d)} \\ &= H(X_{[\mathsf{N}]}) - I(X_{[\mathsf{N}]}; W_1 + W_2 + \cdots + W_{\mathsf{K}}) - H(Q) && \text{(33e)} \\ &\geq H(X_{[\mathsf{N}]}) - H(W_1 + W_2 + \cdots + W_{\mathsf{K}}) - H(Q) && \text{(33f)} \end{aligned}$$

where (33c) comes from that the $X_{[\mathsf{N}]}$ is a function of $Q, W_1, \ldots, W_{\mathsf{K}}$, (33d) comes from that Q is independent of $W_1, \ldots, W_{\mathsf{K}}$. Hence, from (33f) and we have

$$\begin{aligned} \eta \mathsf{L} &\geq H(Q) \geq H(X_{[\mathsf{N}]}) - H(W_1 + \cdots + W_{\mathsf{K}}) && \text{(34a)} \\ &= H(X_{[\mathsf{N}]}) - \mathsf{L} \geq H(X_{s_1}, \ldots, X_{s_v}) - \mathsf{L} && \text{(34b)} \\ &= H(X_{s_1}) + H(X_{s_2} | X_{s_1}) + \cdots \\ &\quad + H(X_{s_v} | X_{s_1}, \ldots, X_{s_{v-1}}) - \mathsf{L}, && \text{(34c)} \end{aligned}$$

Let us then focus on each entropy term in (34c), $H(X_{s_i} | X_{s_1}, \ldots, X_{s_{i-1}})$ where $i \in [v]$. Recall that server s_i can compute some message (assumed to be message W_j) which cannot be computed by servers $s_1, \ldots, s_{i-1}$, and that each message cannot be computed by $\mathsf{N}_{\mathrm{r}} - 1$ servers. We assume that the set of servers which cannot compute W_j is $\overline{\mathcal{A}_j}$. Obviously, $\{s_1, \ldots, s_{i-1}\} \subseteq \overline{\mathcal{A}_j}$. Now consider that the set of responding servers is $\overline{\mathcal{A}_j} \cup \{s_i\}$, totally containing N_{r} servers. As W_j can only be computed by server s_i among the servers in $\overline{\mathcal{A}_j} \cup \{s_i\}$, and from the answers of servers in $\overline{\mathcal{A}_j} \cup \{s_i\}$ the user should recover $W_1 + \cdots + W_{\mathsf{K}}$, we have

$$H(X_{s_i} | X_{s_1}, \ldots, X_{s_{i-1}}) \geq H(X_{s_i} | X_k : k \in \overline{\mathcal{A}_j}) \geq \mathsf{L}. \tag{35}$$

Hence, we take (35) into (34c) to obtain, $\eta \mathsf{L} \geq v\mathsf{L} - \mathsf{L}$, which proves (9).

REFERENCES

[1] J. Xu, S.-L. Huang, L. Song, and T. Lan, "Live gradient compensation for evading stragglers in distributed learning," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 3368–3376.

[2] S. Dutta, V. Cadambe, and P. Grover, "'Short-Dot': Computing large linear transforms distributedly using coded short dot products," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6171–6193, Jul. 2019.

[3] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, early access, Nov. 15, 2021, doi: 10.1109/TIT.2021.3127910.

[4] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5610–5619.

[5] H. Cao, Q. Yan, X. Tang, and G. Han, "Adaptive gradient coding," 2020, *arXiv:2006.04845*.

[6] K. Wan, H. Sun, M. Ji, and G. Caire, "On the tradeoff between computation and communication costs for distributed linearly separable computation," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7390–7405, Nov. 2021.

[7] M. Ben-Or and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[8] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 11–19.

[9] K. Bonawitz *et al.*, "Practical secure aggregation for federated learning on user-held data," 2016, *arXiv:1611.04482*.

[10] K. Bonawitz, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[11] 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 *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1215–1225.

[12] W.-T. Chang and R. Tandon, "On the upload versus download cost for secure and private matrix multiplication," 2019, *arXiv:1906.10684*.

[13] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using Reed–Solomon codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 2027–2031.

[14] N. Raviv, R. Tandon, A. Dimakis, and I. Tamo, "Gradient coding from cyclic MDS codes and expander graphs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2018, pp. 4302–4310.

[15] K. Wan, H. Sun, M. Ji, and G. Caire, "On secure distributed linearly separable computation," 2021, *arXiv:2102.00926*.

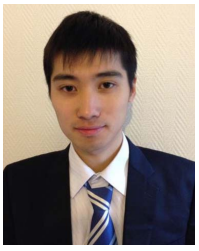
[16] A. Behrouzi-Far and E. Soljanin, "Efficient replication for straggler mitigation in distributed computing," 2020, *arXiv:2006.02318*.

[17] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, Oct. 1980.

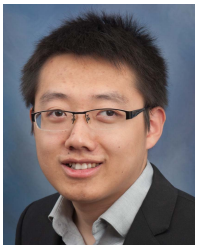
[18] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *Symbolic and Algebraic Computation. EUROSAM 1979* (Lecture Notes in Computer Science), vol. 72, E. W. Ng, Ed. Berlin, Germany: Springer, 1979, pp. 216–226.

[19] R. A. Demillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Inf. Process. Lett.*, vol. 7, no. 4, pp. 193–195, Jun. 1978.

[20] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, no. 4, pp. 656–715, Sep. 1949.



Kai Wan (Member, IEEE) received the B.E. degree in optoelectronics from the Huazhong University of Science and Technology, China, in 2012, and the M.Sc. and Ph.D. degrees in communications from Université Paris-Saclay, France, in 2014 and 2018, respectively. He is currently a Post-Doctoral Researcher with the Communications and Information Theory Chair (CommIT), Technische Universität Berlin, Berlin, Germany. His research interests include information theory, coding techniques, and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, privacy, and security. He has been serving as an Associate Editor for IEEE COMMUNICATIONS LETTERS since August 2021.



Hua Sun (Member, IEEE) received the B.E. degree in communications engineering from the Beijing University of Posts and Telecommunications, China, in 2011, and the M.S. degree in electrical and computer engineering and the Ph.D. degree in electrical engineering from the University of California at Irvine, USA, in 2013 and 2017, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage. He was a recipient of the NSF CAREER Award in 2021 and the UNT College of Engineering Distinguished Faculty Fellowship in 2021. His coauthored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016 and the IEEE GLOBECOM Best Paper Award in 2016.

Mingyue Ji (Member, IEEE) received the B.E. degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2006, the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Sweden, in 2008, the M.Sc. degree in electrical engineering from the University of California at Santa Cruz in 2010, and the Ph.D. degree from the Ming Hsieh Department of Electrical Engineering, University of Southern California, in 2015. He was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Ltd.) from 2015 to 2016. He is now an Assistant Professor with the Electrical and Computer Engineering Department and an Adjunct Assistant Professor with the School of Computing, The University of Utah. He is interested in the broad area of information theory, coding theory, concentration of measure and statistics with the applications of caching networks, wireless communications, distributed storage and computing systems, distributed machine learning, and (statistical) signal processing. He received the IEEE Communications Society Leonard G. Abraham Prize for the Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, the Best Paper Award at 2021 IEEE GLOBECOM Conference, the Best Paper Award at 2015 IEEE ICC Conference, and the Best Student Paper Award at 2010 IEEE European Wireless (EW) Conference and USC Annenberg Fellowship from 2010 to 2014. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS since 2020.

Giuseppe Caire (Fellow, IEEE) was born in Torino in 1965. He received the B.Sc. degree in electrical engineering from the Politecnico di Torino in 1990, the M.Sc. degree in electrical engineering from Princeton University in 1992, and the Ph.D. degree from the Politecnico di Torino in 1994.

He has been a Post-Doctoral Research Fellow with the European Space Agency (ESTEC), Noordwijk, The Netherlands, from 1994 to 1995; an Assistant Professor in telecommunications at the Politecnico di Torino; an Associate Professor at the University of Parma, Italy; a Professor with the Department of Mobile Communications, Eurecom Institute, Sophia-Antipolis, France; and a Professor of electrical engineering with the Viterbi School of Engineering, University of Southern California at Los Angeles. He is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science, Technical University of Berlin, Germany. His main research interests are in the field of communications theory, information theory, channel, and source coding, with particular focus on wireless communications. He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and in 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, the ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, and the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020. He was a recipient of the 2021 Leibinz Prize of the German National Science Foundation (DFG). He has served in the Board of Governors for the IEEE Information Theory Society from 2004 to 2007 and as an Officer from 2008 to 2013. He was the President of the IEEE Information Theory Society in 2011.