This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

Data assignment phase: Each dataset D_k where $k \in [\mathsf{K}]$ is assigned to a subset of N workers in an uncoded manner. Define $\mathcal{Z}_n \subseteq [\mathsf{K}]$ as the set of datasets assigned to worker $n \in [\mathsf{N}]$. The assignment constraint is that

$$|\mathcal{Z}_n| \leq \mathsf{M} := \frac{\mathsf{K}}{\mathsf{N}} \left(\mathsf{N} - \mathsf{N}_\mathrm{r} + \mathsf{m}\right), \ \forall n \in [\mathsf{N}],$$

where M represents the computation cost, and m represents the computation cost factor.[5]

The assignment function of worker n is denoted by φ_n, where

$$\mathcal{Z}_n = \varphi_n(\mathbf{F}), \quad \varphi_n : [\mathbb{F}_\mathsf{q}]^{\mathsf{K}_\mathrm{c}\mathsf{K}} \to \binom{[\mathsf{K}]}{\leq \mathsf{M}},$$

and $\binom{[\mathsf{K}]}{\leq \mathsf{M}}$ represents the set of all subsets of $[\mathsf{K}]$ of size not larger than M. In addition, for each dataset D_k where $k \in [\mathsf{K}]$, we define $\mathcal{H}_k$ as the set of workers to whom dataset D_k is assigned. For each set of datasets $\mathcal{K}$ where $\mathcal{K} \subseteq [\mathsf{K}]$, we define $\mathcal{H}_\mathcal{K} := \cup_{k\in[\mathcal{K}]}\mathcal{H}_k$ as the set of workers to whom there exists some dataset in $\mathcal{K}$ assigned.

Computing phase: Each worker $n \in [\mathsf{N}]$ first computes the message $W_k = f_k(D_k)$ for each $k \in \mathcal{Z}_n$. Worker n then computes

$$X_n = \psi_n(\{W_k : k \in \mathcal{Z}_n\}, \mathbf{F})$$

where the encoding function ψ_n is

$$\psi_n : [\mathbb{F}_\mathsf{q}]^{|\mathcal{Z}_n|\mathsf{L}} \times [\mathbb{F}_\mathsf{q}]^{\mathsf{K}_\mathrm{c}\mathsf{K}} \to [\mathbb{F}_\mathsf{q}]^{\mathsf{T}_n},$$

and T_n represents the length of X_n. Finally, worker n sends X_n to the master.

Decoding phase: The master only waits for the N_r fastest workers' answers to compute $g(W_1, \ldots, W_\mathsf{K})$. In other words, the computation scheme can tolerate $\mathsf{N} - \mathsf{N}_\mathrm{r}$ stragglers. Since the master does not know a priori which workers are stragglers, the computation scheme should be designed so that from the answers of any N_r workers, the master should recover $g(W_1, \ldots, W_\mathsf{K})$. More precisely, for any subset of workers $\mathcal{A} \subseteq [\mathsf{N}]$ where $|\mathcal{A}| = \mathsf{N}_\mathrm{r}$, with the definition

$$X_\mathcal{A} := \{X_n : n \in \mathcal{A}\},$$

there exists a decoding function $\phi_\mathcal{A}$ such that $\hat{g}_\mathcal{A} = \phi_\mathcal{A}\big(X_\mathcal{A}, \mathbf{F}\big)$, where the decoding function is

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

The worst-case probability of error is defined as

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

In addition, 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 downloaded by the master from any N_r responding workers. The communication cost R is achievable if there exists a computation scheme with assignment, encoding, and decoding functions such that

$$\lim_{\mathsf{q}\to\infty} \lim_{\mathsf{L}\to\infty} \varepsilon = 0.$$

Since the probability of each demand matrix is identical, the above constraint implies that any achievable computing scheme should work for most demand matrices with dimension $\mathsf{K}_\mathrm{c} \times \mathsf{K}$.

The objective is to characterize the optimal tradeoff between the computation and communication costs $(\mathsf{m}, \mathsf{R}^\star)$, i.e., for each $\mathsf{m} \in [\mathsf{N}_\mathrm{r}]$, we aim to find the minimum communication cost $\mathsf{R}^\star$.

As shown in [5, Section II], since the elements of the demand matrix $\mathbf{F}$ are uniformly i.i.d. over a large enough field $\mathbb{F}_\mathsf{q}$, the desired task contains K_c linearly independent combinations of messages with high probability, where each message contains L uniformly i.i.d. symbols on $\mathbb{F}_\mathsf{q}$; thus a simple cut-set bound argument yields

$$\mathsf{R}^\star \geq \mathsf{K}_\mathrm{c}. \tag{1}$$

The cyclic assignment was widely used in the existing works on the distributed computing problems [5], [17]–[21]. For each dataset D_k where $k \in [\mathsf{K}]$, we assign D_k to the workers in $\mathcal{H}_k$ where (recall that in this paper we let $a \bmod b = b$ if b divides a)

$$\begin{aligned}\mathcal{H}_k = \big\{&k \bmod \mathsf{N}, (k-1) \bmod \mathsf{N}, \ldots, \\ &(k - \mathsf{N} + \mathsf{N}_\mathrm{r} - \mathsf{m} + 1) \bmod \mathsf{N}\big\}.\end{aligned} \tag{2}$$

Thus the set of datasets assigned to worker $n \in [\mathsf{N}]$ is

$$\begin{aligned}\mathcal{Z}_n = \bigcup_{p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]} \big\{&(n \bmod \mathsf{N}) + p\mathsf{N}, ((n+1) \bmod \mathsf{N}) + p\mathsf{N}, \ldots, \\ &((n + \mathsf{N} - \mathsf{N}_\mathrm{r} + \mathsf{m} - 1) \bmod \mathsf{N}) + p\mathsf{N}\big\}\end{aligned} \tag{3}$$

with cardinality $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_\mathrm{r} + \mathsf{m})$. For example, if $\mathsf{K} = \mathsf{N} = 4$, $\mathsf{N}_\mathrm{r} = 3$ and $\mathsf{m} = 2$, by the cyclic assignment with $p = 0$ in (3), we have

$$\mathcal{H}_1 = \{1,3,4\},\ \mathcal{H}_2 = \{1,2,4\},\ \mathcal{H}_3 = \{1,2,3\},\ \mathcal{H}_4 = \{2,3,4\};$$
$$\mathcal{Z}_1 = \{1,2,3\},\ \mathcal{Z}_2 = \{2,3,4\},\ \mathcal{Z}_3 = \{3,4,1\},\ \mathcal{Z}_4 = \{4,1,2\}.$$

For each $\mathsf{m} \in [\mathsf{N}_\mathrm{r}]$, the minimum communication cost under the cyclic assignment in (3) is denoted by $\mathsf{R}^\star_\mathrm{cyc}$. Clearly, we have $\mathsf{R}^\star_\mathrm{cyc} \geq \mathsf{R}^\star$.

Remark 1. *It will be clear that the assumption that the desired function's coefficients (i.e., the elements in demand matrix $\mathbf{F}$) are uniformly i.i.d. over a large enough field, is needed for the information theoretic converse bounds, and to prove the decodability of the proposed computing scheme with vanishing probability of error by the Schwartz-Zippel lemma [25]–[27].*[6] *As shown in [5, Remark 3], for some specific demand matrices, the optimal communication costs can be strictly higher than $\mathsf{R}^\star$. It is one of our on-going works to study the arbitrary demand matrices.*

[5] It was proved in [5] that in order to tolerate $\mathsf{N}-\mathsf{N}_\mathrm{r}$ stragglers, the minimum computation cost is $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_\mathrm{r} + 1)$.

[6] The Schwartz-Zippel lemma [25]–[27] shows that the realization of a multivariate polynomial is not equal to zero with high probability if the coefficients of this polynomial are not all zero and each variable in the polynomial is uniformly i.i.d. over a large enough field.

In contrast, the assumption that the symbols in each message are uniformly i.i.d., is only needed for the information theoretic converse bounds, while the proposed computing scheme in this paper works for any arbitrary component functions $f_k(D_k)$ where $k \in [\mathsf{K}]$. □

Special cases: The sub-case of the considered problem for $\mathsf{K_c} = 1$ and any m was studied in [20], [21] and the sub-case for $\mathsf{m} = 1$ and any $\mathsf{K_c}$ was studied in [5].

- $\mathsf{K_c} = 1$. It was proved in [20], [21] that when $\mathsf{K_c} = 1$, the communication cost $\frac{\mathsf{N_r}}{\mathsf{m}}$ is optimal under the constraint of linear coding in the computing phase and symmetric transmission (i.e., the number of symbols transmitted by each worker is the same).
- $\mathsf{m} = 1$. The communication cost by the computing scheme in [5] is $\mathsf{N_r K_c}$ when $\mathsf{K_r} \leq \frac{\mathsf{K}}{\mathsf{N}}$; is $\frac{\mathsf{K N_r}}{\mathsf{N}}$ when $\frac{\mathsf{K}}{\mathsf{N}} \leq \mathsf{K_c} \leq \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}$; is $\mathsf{K_c}$ when $\mathsf{K_c} \geq \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}$. The communication cost is exactly optimal when $\mathsf{K} = \mathsf{N}$, or when $\mathsf{K_c} \in \left[\left\lceil \frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N_r}+1}} \right\rceil\right]$, or when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r} : \mathsf{K}\right]$. In addition, it is optimal under the constraint of the cyclic assignment when N divides K.

III. Main Results

A. Novel Converse and Achievable Bounds

We first provide a converse bound under the constraint of the cyclic assignment, which will be proved in Section IV.

Theorem 1. *For the $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{m})$ distributed linearly separable computation problem,*

- *when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)\right]$, by defining $\mathsf{u} := \left\lceil \frac{\mathsf{K_c N}}{\mathsf{K}} \right\rceil$, we have*

$$\mathsf{R}^{\star}_{cyc} \geq \frac{\mathsf{N_r K_c}}{\mathsf{m} + \mathsf{u} - 1}. \tag{4a}$$

- *when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1) : \mathsf{K}\right]$, we have*

$$\mathsf{R}^{\star}_{cyc} \geq \mathsf{R}^{\star} \geq \mathsf{K_c}. \tag{4b}$$

□

We then introduce the computation-communication costs tradeoff by the novel computing scheme in the following theorem.

Theorem 2. *For the $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{m})$ distributed linearly separable computation problem where*

$$40 \geq \mathsf{N} \geq \frac{\mathsf{m} + \mathsf{u} - 1}{\mathsf{u}} + \mathsf{u}(\mathsf{N_r} - \mathsf{m} - \mathsf{u} + 1), \tag{5}$$

the computation-communication costs tradeoff $(\mathsf{m}, \mathsf{R_{ach}})$ is achievable, where

- *when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$,*

$$\mathsf{R_{ach}} = \frac{\mathsf{K_c N_r}}{\mathsf{m}} \tag{6a}$$

- *when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)\right]$,*

$$\mathsf{R_{ach}} = \frac{\mathsf{N_r K u}}{\mathsf{N}(\mathsf{m} + \mathsf{u} - 1)}; \tag{6b}$$

- *when $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1) : \mathsf{K}\right]$,*

$$\mathsf{R_{ach}} = \mathsf{K_c}. \tag{6c}$$

□

Notice that the RHS of the constraint (5)

$$\mathsf{N} \geq \frac{\mathsf{m} + \mathsf{u} - 1}{\mathsf{u}} + \mathsf{u}(\mathsf{N_r} - \mathsf{m} - \mathsf{u} + 1), \tag{7}$$

will be explained in Remark 3 from a viewpoint of linear space dimension. It can be seen that in the first case of the proposed computing scheme (i.e., $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$), we have $\mathsf{u} = 1$ and thus the constraint (7) always holds. In the third case of the proposed computing scheme (i.e., $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1) : \mathsf{K}\right]$), we have $\mathsf{u} \geq \mathsf{N_r} - \mathsf{m} + 1$ and thus the constraint in (7) always holds.

While proving the decodability of the proposed computing scheme in Theorem 2, we use the Schwartz-Zippel lemma [25]–[27] in Appendix A. For the non-zero polynomial condition for the Schwartz-Zippel lemma, we numerically verify all cases that $40 \geq \mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}(\mathsf{N_r} - \mathsf{m} - \mathsf{u} + 1)$, and conjecture in the rest of the paper that the condition holds for any case where $\mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}(\mathsf{N_r} - \mathsf{m} - \mathsf{u} + 1)$, i.e., in Theorem 2 we replace the constraint (5) by (7).

In Section V, due to the space limitation, we will only provide the computing scheme for the second case (6b) (i.e., $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)\right]$). By the exactly same method as described in [5, Sections IV-B and IV-C], the computing schemes for the first and third cases can be obtained by the direct extensions of the computing scheme for the second case. More precisely,

- $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$. When $\mathsf{K_c} = 1$, it can be easily shown (see [5, Section IV-B]) that the $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, 1, \mathsf{m})$ distributed linearly separable computation problem is equivalent to the $(\mathsf{N}, \mathsf{N}, \mathsf{N_r}, 1, \mathsf{m})$ distributed linearly separable computation problem, which needs the communication cost $\frac{\mathsf{N_r}}{\mathsf{m}}$ from (6b). For $\mathsf{K_c} \in \left[2 : \frac{\mathsf{K}}{\mathsf{N}}\right]$, we can treat the $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{m})$ distributed linearly separable computation problem as $\mathsf{K_c}$ independent $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, 1, \mathsf{m})$ distributed linearly separable computation problems; thus the communication cost is $\frac{\mathsf{K_c N_r}}{\mathsf{m}}$, coinciding with (6a).
- $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1) : \mathsf{K}\right]$. When $\mathsf{K_c} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)$, from (6b) it can be seen that the communication cost is $\frac{\mathsf{N_r K u}}{\mathsf{N}(\mathsf{m}+\mathsf{u}-1)} = \frac{\mathsf{K u}}{\mathsf{N}} = \mathsf{K_c}$, coinciding with (6c). When $\mathsf{K_c} > \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)$, as in [5, Section IV-C], we can divide each demanded linear combination into $\binom{\mathsf{K_c}-1}{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)-1}$ equal-length sub-combinations, each of which has $\frac{\mathsf{L}}{\binom{\mathsf{K_c}-1}{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)-1}}$ symbols. We then treat the $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{m})$ distributed linearly separable computation problem as $\binom{\mathsf{K_c}}{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)}$ independent $\left(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1), \mathsf{m}\right)$ distributed linearly separable computation sub-problems, where in each sub-problem we let the master recover $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)$ sub-combinations, with the communication cost $\frac{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)}{\binom{\mathsf{K_c}-1}{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)-1}}$; thus the total communication cost is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

$\binom{\mathsf{K_c}}{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)} \frac{\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)}{\mathsf{K_c}-1} = \mathsf{K_c}$, coinciding with (6c).

By comparing the proposed converse bound in Theorem 1 and the proposed scheme in Theorem 2, we can directly obtain the following (order) optimality results.

Theorem 3. *For the* $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})$ *distributed linearly separable computation problem where* $\mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}(\mathsf{N_r} - \mathsf{m} - \mathsf{u} + 1)$*,*

- *when* $\mathsf{K} = \mathsf{N}$*, we have*

$$\mathsf{R}^{\star}_{cyc} = \mathsf{R}_{\rm ach} = \begin{cases} \frac{\mathsf{N_r}\mathsf{K_c}}{\mathsf{m}+\mathsf{u}-1}, & \text{if } \mathsf{K_c} \in [\mathsf{N_r}-\mathsf{m}+1]; \\ \mathsf{K_c}, & \text{if } \mathsf{K_c} \in [\mathsf{N_r}-\mathsf{m}+1:\mathsf{K}]; \end{cases}$$

- *when* $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$*, we have*

$$\mathsf{R}^{\star}_{cyc} = \mathsf{R}_{\rm ach} = \frac{\mathsf{N_r}\mathsf{K_c}}{\mathsf{m}};$$

- *when* $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}+1 : \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)-1\right]$*, we have*

$$\mathsf{R}^{\star}_{cyc} \geq \frac{\mathsf{K_c}}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}}\mathsf{R}_{\rm ach} \geq \frac{\mathsf{R}_{\rm ach}}{2};$$

- *when* $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1) : \mathsf{K}\right]$*, we have*

$$\mathsf{R}^{\star} = \mathsf{R}^{\star}_{cyc} = \mathsf{R}_{\rm ach} = \mathsf{K_c}.$$

□

In words, for the considered problem satisfying the constraint in (7), when $\mathsf{K_c} \in [\mathsf{N_r}-\mathsf{m}+1 : \mathsf{K}]$, the proposed computing scheme is exactly optimal; when $\mathsf{K} = \mathsf{N}$ or $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$, the proposed computing scheme is optimal under the constraint of the cyclic assignment; when N divides K and $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}+1 : \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)-1\right]$, the proposed scheme is order optimal within a factor of $\frac{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}}{\mathsf{K_c}} \leq 2$ under the constraint of the cyclic assignment. Note that when $\mathsf{K_c} = 1$, the proposed computing scheme achieves the same communication load as in [20], [21], which was proved to be optimal under the constraint of linear coding in the computing phase and symmetric transmission. Instead, we prove that it is optimal only under the constraint of the cyclic assignment.

Remark 2. *When the elements in* $\mathbf{F}$ *and* $[W_1;\ldots;W_{\mathsf{K}}]$ *are on the field of real numbers, the proposed computing scheme in Theorem 2 can work with high probability if each element in* $\mathbf{F}$ *is uniformly i.i.d. over a large enough finite set of real numbers. For example, real numbers in finite arithmetic (either fixed points or floating points) can be in a discrete and large finite set. Note that, the decodability proof of the proposed computing scheme is based on the Schwartz-Zippel lemma [25]–[27], while this lemma is valid for any field if each variable in the multivariate polynomial (i.e., some element in* $\mathbf{F}$ *or some dummy variable) is uniformly i.i.d. over a large enough finite set. Furthermore, by a simple extension, the proposed computing scheme can also work with high probability if each element in* $\mathbf{F}$ *is uniformly i.i.d. over an interval of real numbers. This is because for a non-zero multivariate polynomial with finite degree where the range of the variables is an interval of real number, the set of roots of this polynomial has measure* 0*.* □

B. Numerical Evaluations

We end this section by providing some numerical evaluations on the proposed converse and achievable bounds. In Fig. 1, we provide some numerical evaluations on the proposed converse and achievable bounds. For the sake of comparison, we introduce a baseline scheme. For the case where $\mathsf{K_c} = 1$, the computing scheme in [20], [21] needs the communication cost $\frac{\mathsf{N_r}}{\mathsf{m}}$ for each $\mathsf{m} \in [\mathsf{N}]$. Hence, a simple baseline scheme can be obtained by treating the considered problem as $\mathsf{K_c}$ independent sub-problems, where in each sub-problem the master recover one of its desired linear combination. Thus the communication cost for the baseline scheme is

$$\mathsf{R}_{\rm base} = \mathsf{K_c}\mathsf{N_r}/\mathsf{m}, \ \forall \mathsf{m} \in [\mathsf{N_r}]. \tag{8}$$

In Fig. 1a, we consider the distributed linearly separable computation problem where $\mathsf{K} = 20$, $\mathsf{N} = 10$, $\mathsf{N_r} = 8$, and $\mathsf{K_c} = 8$. In this example, the constraint in (7) always holds. It can be seen from Fig. 1a that the proposed computing scheme outperforms the baseline scheme and coincides with the proposed converse bound.

In Fig. 1b, we consider the distributed linearly separable computation problem where $\mathsf{K} = 20$, $\mathsf{N} = 10$, $\mathsf{N_r} = 7$, $\mathsf{m} = 2$. For each $\mathsf{K_c} \in [20]$, we plot the communication costs. In this example, the constraint in (7) also always holds. It can be seen from Fig. 1b that the proposed computing scheme outperforms the baseline scheme. The propose scheme coincides with the proposed converse bound when $\mathsf{K_c} \leq \frac{\mathsf{K}}{\mathsf{N}} = 2$, or when $\mathsf{K_c}$ divides $\frac{\mathsf{K}}{\mathsf{N}}$, or when $\mathsf{K_c} \geq \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1) = 12$.

The focus of the paper is on some large enough finite field, where the proposed computing scheme in Theorem 2 works with high probability. However, in practice the field size is limited. In Table I, we illustrate the probabilities that the proposed scheme works on the different finite fields by the Monte Carlo simulation. For each considered system, we randomly generate 10^4 demand matrices and count the number of demand matrices for which the proposed computing can work. In Table Ia we consider that $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m}) = (6,6,5,2,2)$, and in Table Ib we consider that $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m}) = (11,11,7,2,2)$. Both tables show that the success probability of the proposed computing scheme increases as q grows. In addition, it increases faster in the smaller computing system than in the larger system.

IV. Proof of Theorem 1

When $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1) : \mathsf{K}\right]$, the converse bound in Theorem 1 is the cut-set converse bound in (1). Hence, in the following we focus on the case $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)\right]$.

We will use an example to illustrate the main idea.

Example 1. *In this example, we have* $\mathsf{N} = \mathsf{K} = 5$*,* $\mathsf{N_r} = 4$*,* $\mathsf{m} = 2$*, and* $\mathsf{K_c} = \mathsf{u} = 2$*.*

The number of datasets assigned to each worker is $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N_r}+\mathsf{m}) = 3$*. Each dataset is assigned to* 3 *workers. With the cyclic assignment, we assign*

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

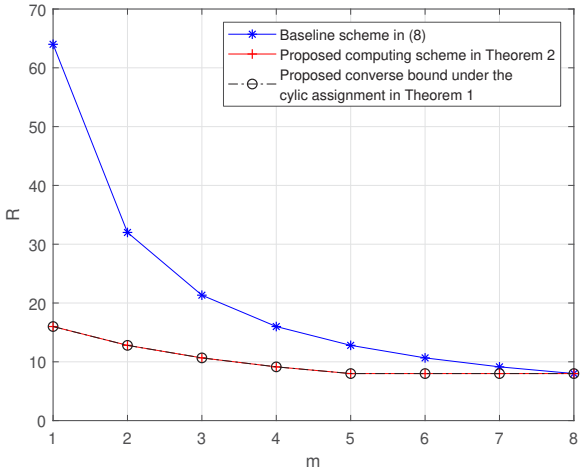
$\mathsf{q}=2$	$\mathsf{q}=7$	$\mathsf{q}=11$	$\mathsf{q}=13$	$\mathsf{q}=19$	$\mathsf{q}=29$
0	0.0637	0.2846	0.3674	0.5134	0.6566
$\mathsf{q}=71$	$\mathsf{q}=113$	$\mathsf{q}=173$	$\mathsf{q}=229$	$\mathsf{q}=541$	$\mathsf{q}=3571$
0.8398	0.8961	0.9328	0.9504	0.978	0.9968

(a) $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})=(6,6,5,2,2)$.

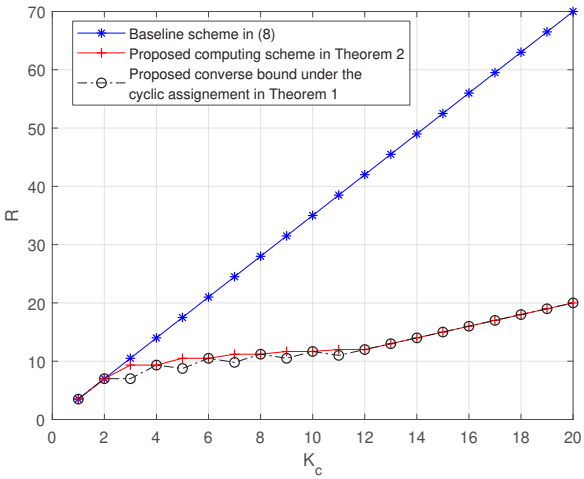
$\mathsf{q}=2$	$\mathsf{q}=29$	$\mathsf{q}=71$	$\mathsf{q}=83$	$\mathsf{q}=101$	$\mathsf{q}=113$
0	0	0.0903	0.1913	0.3679	0.4771
$\mathsf{q}=131$	$\mathsf{q}=149$	$\mathsf{q}=173$	$\mathsf{q}=229$	$\mathsf{q}=541$	$\mathsf{q}=3571$
0.6324	0.7529	0.8292	0.9048	0.9606	0.9943

(b) $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})=(11,11,7,2,2)$.

TABLE I: The probabilities that the proposed scheme works on different finite fields, for the $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})$ distributed linearly separable computation problem.



(a) The computation-communication costs tradeoff for the case $\mathsf{K}=20$, $\mathsf{N}=10$, $\mathsf{N_r}=8$, $\mathsf{K_c}=8$.



(b) The communication costs for the case $\mathsf{K}=20$, $\mathsf{N}=10$, $\mathsf{N_r}=7$, $\mathsf{m}=2$.

Fig. 1: Numerical evaluations for the considered distributed linearly separable computation problem.

We consider the demand matrix $\mathbf{F}$ *whose dimension is* 2×5 *with elements uniformly i.i.d. over large field* $\mathbb{F}_{\mathsf{q}}$. *Hence, the sub-matrix including each* $\mathcal{K}_{\mathrm{c}}=2$ *columns is full rank with high probability.*

Notice that in this example the number of stragglers is $\mathsf{N}-\mathsf{N_r}=1$. *We first consider that worker* 5 *is the straggler; thus the master should recover* $\mathbf{F}[W_1;\ldots;W_5]$ *from the answers of workers in* $\mathcal{A}=[4]$. *In addition, each dataset is assigned to* $\mathsf{N}-\mathsf{N_r}+\mathsf{m}=3$ *workers. Hence, there must exist one dataset assigned to all the straggler(s) which is also assigned to* m *responding workers. In this example, all of* D_1, D_2, *and* D_5 *belong to such datasets. Now we select one of them, e.g.,* D_2. *Note that* D_2 *is assigned to workers* $\mathcal{H}_2=\{1,2,5\}$. *We then consider the next dataset* $D_{(2+1) \bmod \mathsf{K}}=D_3$. *The set of workers storing dataset* D_3 *(denoted by* $\mathcal{H}_3$*) is obtained by right-shifting* $\mathcal{H}_2$ *by one position, i.e.,* $\mathcal{H}_3=\{1,2,3\}$. *Hence, there is exactly one new worker in* $\mathcal{H}_3$ *who is not in* $\mathcal{H}_2\cap\mathcal{A}$, *which is worker* 3. *So we have*

$$|(\mathcal{H}_2\cup\mathcal{H}_3)\cap\mathcal{A}|=\mathsf{m}+(2-1)=3=\mathsf{m}+\mathsf{u}-1;$$

in other words, in the set of responding workers $\mathcal{A}$, *the number of workers who can compute* W_2 *or* W_3 *is equal to* 3. *In addition, the sub-matrix of* $\mathbf{F}$ *including the columns in* $\{2,3\}$ *is full rank (with rank* $\mathsf{K_c}=2$*). Recall that each message has* L *uniformly i.i.d. symbols. Hence, the number of transmitted symbols by workers in* $(\mathcal{H}_2\cup\mathcal{H}_3)\cap\mathcal{A}$ *should be no less than* $2\mathsf{L}$*; thus*

$$\sum_{n\in\left((\mathcal{H}_2\cup\mathcal{H}_3)\cap\mathcal{A}\right)} T_n=T_1+T_2+T_3 \tag{9a}$$

$$\geq H\left(\mathbf{F}[W_1;\ldots;W_5]|W_1,W_4,W_5\right)\geq \mathsf{K_c}\mathsf{L}=2\mathsf{L}. \tag{9b}$$

Similarly, considering that worker 4 *is the straggler, we have*

$$T_5+T_1+T_2\geq \mathsf{K_c}\mathsf{L}=2\mathsf{L}. \tag{10}$$

Considering that worker 3 *is the straggler, we have*

$$T_4+T_5+T_1\geq \mathsf{K_c}\mathsf{L}=2\mathsf{L}. \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

Considering that worker 2 *is the straggler, we have*

$$T_3 + T_4 + T_5 \geq \mathsf{K_c L} = 2\mathsf{L}. \tag{12}$$

Considering that worker 1 *is the straggler, we have*

$$T_2 + T_3 + T_4 \geq \mathsf{K_c L} = 2\mathsf{L}. \tag{13}$$

By summing (9b)-(13)*, we have*

$$T_1 + T_2 + T_3 + T_4 + T_5 \geq \frac{10}{3}\mathsf{L},$$

which leads that

$$\mathsf{R}^{\star}_{cyc} \geq \max_{\mathcal{A}\subseteq[5]:|\mathcal{A}|=\mathsf{N_r}=4} \frac{\sum_{j\in\mathcal{A}} T_j}{\mathsf{L}} \geq \frac{8}{3},$$

as the converse bound in (4a). □

We are now ready to generalize the proposed converse bound under the constraint of the cyclic assignment in Example 1. Recall that we consider the case where $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)\right]$ and that $\mathsf{u} = \left\lceil \frac{\mathsf{K_c N}}{\mathsf{K}} \right\rceil$. The demand matrix $\mathbf{F}$ has the dimension $\mathsf{K_c} \times \mathsf{K}$ with elements uniformly i.i.d. over a large enough finite field. Hence, the sub-matrix including each $\mathcal{K}_c$ columns is full rank with high probability. By the cyclic assignment, as shown in (2), each dataset D_k is assigned to workers $\mathcal{H}_k = \big\{k \bmod \mathsf{N}, (k-1) \bmod \mathsf{N}, \ldots, (k - \mathsf{N} + \mathsf{N_r} - \mathsf{m} + 1) \bmod \mathsf{N}\big\}$.

We consider the set of stragglers who are adjacent. Thus each time we choose one integer $n \in [\mathsf{N}]$, let $\mathcal{S}_n := \{n \bmod \mathsf{N}, (n-1) \bmod \mathsf{N}, \ldots, (n - \mathsf{N} + \mathsf{N_r} + 1) \bmod \mathsf{N}\}$ where $|\mathcal{S}_n| = \mathsf{N} - \mathsf{N_r}$, be the set of stragglers. The master should recover $\mathbf{F}[W_1; \ldots; W_{\mathsf{K}}]$ from the answers of workers in $[\mathsf{N}] \backslash \mathcal{S}_n$. From the cyclic assignment, there are exactly $\frac{\mathsf{K}}{\mathsf{N}}$ datasets, denoted by $\mathcal{U}_0 = \big\{((n+\mathsf{m}) \bmod \mathsf{N}) + p\mathsf{N} : p \in \big[0 : \frac{\mathsf{K}}{\mathsf{N}} - 1\big]\big\}$, which are exclusively assigned to the workers in

$$\begin{aligned}
&\mathcal{H}_{\mathcal{U}_0} = \mathcal{S}_n \cup \{(n+1) \bmod \mathsf{N}, (n+2) \bmod \mathsf{N}, \\
&\ldots, (n+\mathsf{m}) \bmod \mathsf{N}\} \\
&= \{(n - \mathsf{N} + \mathsf{N_r} + 1) \bmod \mathsf{N}, (n - \mathsf{N} + \mathsf{N_r} + 2) \bmod \mathsf{N}, \\
&\ldots, (n+\mathsf{m}) \bmod \mathsf{N}\}.
\end{aligned}$$

Then for each $i \in [\mathsf{u} - 1]$, the datasets in $\mathcal{U}_i = \big\{((n + \mathsf{m} + i) \bmod \mathsf{N}) + p\mathsf{N} : p \in \big[0 : \frac{\mathsf{K}}{\mathsf{N}} - 1\big]\big\}$, are exclusively assigned to the workers in

$$\begin{aligned}
&\mathcal{H}_{\mathcal{U}_i} = \{(n - \mathsf{N} + \mathsf{N_r} + i + 1) \bmod \mathsf{N}, (n - \mathsf{N} + \mathsf{N_r} + i + 2) \\
&\bmod \mathsf{N}, \ldots, (n + \mathsf{m} + i) \bmod \mathsf{N}\}.
\end{aligned}$$

It can be seen that there are totally $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}$ datasets in $\cup_{i\in[0:\mathsf{u}-1]}\mathcal{U}_i$, which are exclusively assigned to the workers in

$$\begin{aligned}
&\cup_{i\in[0:\mathsf{u}-1]}\, \mathcal{H}_{\mathcal{U}_i} = \{(n - \mathsf{N} + \mathsf{N_r} + 1) \bmod \mathsf{N}, \\
&(n - \mathsf{N} + \mathsf{N_r} + 2) \bmod \mathsf{N}, \ldots, (n + \mathsf{m} + \mathsf{u} - 1) \bmod \mathsf{N}\} \\
&= \mathcal{S}_n \cup \{(n+1) \bmod \mathsf{N}, \ldots, (n + \mathsf{m} + \mathsf{u} - 1) \bmod \mathsf{N}\}.
\end{aligned}$$

Note that since $\mathsf{u} \leq \mathsf{N_r} - \mathsf{m} + 1$, we have $\mathcal{S}_n \cap \{(n+1) \bmod \mathsf{N}, \ldots, (n + \mathsf{m} + \mathsf{u} - 1) \bmod \mathsf{N}\} = \emptyset$. In other words, the number of responding workers in $\cup_{i\in[0:\mathsf{u}-1]}\mathcal{H}_{\mathcal{U}_i}$ is

$$\begin{aligned}
&\big|\big(\cup_{i\in[0:\mathsf{u}-1]}\mathcal{H}_{\mathcal{U}_i}\big) \cap ([\mathsf{N}] \setminus \mathcal{S}_n)\big| \\
&= |\{(n+1) \bmod \mathsf{N}, \ldots, (n + \mathsf{m} + \mathsf{u} - 1) \bmod \mathsf{N}\}| \\
&= \mathsf{m} + \mathsf{u} - 1.
\end{aligned}$$

Since $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u} \geq \mathsf{K_c}$, the sub-matrix of the demand matrix including the columns in $\cup_{i\in[0:\mathsf{u}-1]}\mathcal{U}_i$ has a rank equal to $\mathsf{K_c}$ with high probability. Hence, the number of transmitted symbols by workers in $\{(n+1) \bmod \mathsf{N}, \ldots, (n + \mathsf{m} + \mathsf{u} - 1) \bmod \mathsf{N}\}$ should be no less than $\mathsf{K_c L}$; thus

$$\sum_{j\in\{(n+1) \bmod \mathsf{N}, \ldots, (n+\mathsf{m}+\mathsf{u}-1) \bmod \mathsf{N}\}} T_j \geq \mathsf{K_c L}. \tag{14}$$

By considering all $n \in [\mathsf{N}]$ and summing all the inequalities as in (14), we have

$$\sum_{j\in[\mathsf{N}]} T_j \geq \frac{\mathsf{N K_c}}{\mathsf{m} + \mathsf{u} - 1}\mathsf{L},$$

which leads that

$$\mathsf{R}^{\star}_{\text{cyc}} \geq \frac{\max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N_r}} \sum_{j\in\mathcal{A}} T_j}{\mathsf{L}} \geq \frac{\mathsf{N_r K_c}}{\mathsf{m} + \mathsf{u} - 1},$$

as the converse bound in (4a).

V. Proof of (6b)

We focus on the case where $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r} - \mathsf{m} + 1)\right]$. We first illustrate the main idea in the following example.

Example 2. *In this example, we have* $\mathsf{N} = \mathsf{K} = 6$*,* $\mathsf{N_r} = 5$*,* $\mathsf{m} = 2$*, and* $\mathsf{K_c} = 2$*. Since* $\mathsf{N} = \mathsf{K}$ *in this example, we have* $\mathsf{u} = \mathsf{K_c} = 2$*. For the sake of simplicity, while illustrating the proposed scheme through this example, we assume that the field is a large enough prime field. It will be proved that in general this assumption is not necessary in our proposed scheme. We assume that the demand matrix is*

$$\mathbf{F} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & f_{1,4} & f_{1,5} & f_{1,6} \\ f_{2,1} & f_{2,2} & f_{2,3} & f_{2,4} & f_{2,5} & f_{2,6} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

Data assignment phase: The number of datasets assigned to each worker is $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + \mathsf{m}) = 3$*. We use the cyclic assignment, to assign*

Worker 1	*Worker 2*	*Worker 3*	*Worker 4*	*Worker 5*	*Worker 6*
D_1	D_2	D_3	D_4	D_5	D_6
D_2	D_3	D_4	D_5	D_6	D_1
D_3	D_4	D_5	D_6	D_1	D_2

Computing phase: Since the communication cost is no less than $\mathsf{N_r}\frac{\mathsf{K_c}}{\mathsf{m}+\mathsf{u}-1} = \frac{10}{3}$ *from the converse bound* (4a)*, we divide each message* W_k *where* $k \in [6]$ *into* $\mathsf{m}+\mathsf{u}-1 = 3$ *non-overlapping and equal-length sub-messages,* $W_k = \{W_{k,j} : j \in [3]\}$*. Hence, the task function becomes* $(\mathsf{m}+\mathsf{u}-1)\mathsf{K_c} = 6$ *linear combinations of sub-messages. Each worker should send* $\mathsf{K_c} = 2$ *linear combinations of sub-messages. From the answers of* $\mathsf{N_r} = 5$ *workers, the master totally receives* $\mathsf{N_r K_c} = 10$ *linear combinations of sub-messages. Hence, we generate* $\mathsf{v} = 10 - 6 = 4$ *virtually demanded linear combinations of sub-messages; thus the effective demand matrix (i.e., containing original and virtual demands) is*

$$\mathbf{F}'[W_{1,1}; \ldots; W_{6,1}; W_{1,2}; \ldots; W_{6,3}]$$

where $\mathbf{F}'$ *has the dimension* $\mathsf{N_r K_c} \times \mathsf{K}(\mathsf{m}+\mathsf{u}-1) = 10 \times 18$*, with the form in* (15)*.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

$$\mathbf{F}' = \underbrace{\begin{bmatrix} 1 & \cdots & 1 \\ 0 & \cdots & 5 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ a_{1,1} & \cdots & a_{1,6} \\ a_{2,1} & \cdots & a_{2,6} \\ a_{3,1} & \cdots & a_{3,6} \\ a_{4,1} & \cdots & a_{4,6} \end{bmatrix}}_{\mathbf{F}'_1} \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 0 & \cdots & 5 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ a_{1,7} & \cdots & a_{1,12} \\ a_{2,7} & \cdots & a_{2,12} \\ a_{3,7} & \cdots & a_{3,12} \\ a_{4,7} & \cdots & a_{4,12} \end{bmatrix}}_{\mathbf{F}'_2} \underbrace{\begin{bmatrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 0 & \cdots & 5 \\ a_{1,13} & \cdots & a_{1,18} \\ a_{2,13} & \cdots & a_{2,18} \\ a_{3,13} & \cdots & a_{3,18} \\ a_{4,13} & \cdots & a_{4,18} \end{bmatrix}}_{\mathbf{F}'_3}. \tag{15}$$

The transmissions of the 6 *workers can be expressed as*

$$\mathbf{S}\ \mathbf{F}'\ [W_{1,1};\ldots;W_{6,1};W_{1,2};\ldots;W_{6,3}] = [\mathbf{s}^{1,1};\mathbf{s}^{1,2};\mathbf{s}^{2,1};\ldots;\mathbf{s}^{6,2}]\mathbf{F}'[W_{1,1};\ldots;W_{6,1};W_{1,2};\ldots;W_{6,3}],$$

where the row vector $\mathbf{s}^{n,j}$ *represents the* j^{th} *transmission vector of worker* n*; in other words,* $\mathbf{s}^{n,j}\mathbf{F}'[W_{1,1};\ldots;W_{6,1};W_{1,2};\ldots;W_{6,3}]$ *represents the* j^{th} *transmitted linear combination by worker* n*. We can further expand* $\mathbf{S}$ *as in* (16)*.*

Now the j^{th} *transmitted linear combination by worker* n *can be expressed as*

$$\mathbf{s}^{n,j}\mathbf{d}_1 W_{1,1} + \mathbf{s}^{n,j}\mathbf{d}_2 W_{2,1} + \cdots + \mathbf{s}^{n,j}\mathbf{d}_6 W_{6,1} + \mathbf{s}^{n,j}\mathbf{d}_7 W_{1,2} + \cdots + \mathbf{s}^{n,j}\mathbf{d}_{18} W_{6,3}, \tag{17}$$

where $\mathbf{d}_i$ *represents the* i^{th} *column of* $\mathbf{F}'$*. Recall that* $\overline{\mathcal{Z}_n} \subseteq [\mathsf{K}]$ *represents the set of messages which are not assigned to worker* n*. Hence, to guarantee that the linear combination in* (17) *can be transmitted by worker* n*, we should have*

$$\mathbf{s}^{n,j}\mathbf{d}_{k+(t-1)\mathsf{K}} = 0,\ \forall n \in [6], j \in [2], t \in [3], k \in \overline{\mathcal{Z}_n}. \tag{18}$$

In addition, for each set $\mathcal{A} \subseteq [6]$ *where* $|\mathcal{A}| = 5$*, by receiving the linear combinations transmitted by the workers in* $\mathcal{A}$*, the master should recover the desired linear combinations. Hence, we should have (recalling that* $\mathcal{A}(i)$ *represents the* i^{th} *smallest element of* $\mathcal{A}$*)*

$$[\mathbf{s}^{\mathcal{A}(1),1};\mathbf{s}^{\mathcal{A}(1),2};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(5),2}] \text{ is full rank}, \tag{19}$$

$\forall \mathcal{A} \subseteq [6]$ *where* $|\mathcal{A}| = 5$*. Our objective is to determine the elements in* $\mathbf{S}$ *and in* $\mathbf{F}'$ *such that the constraints in* (18) *and* (19) *are satisfied.*

We divide matrix $\mathbf{F}'$ *into* 3 *sub-matrices,* $\mathbf{F}'_1, \mathbf{F}'_2, \mathbf{F}'_3$ *each of which has the dimension* 10×6*, as illustrated in* (15)*. We also divide matrix* $\mathbf{S}$ *into* 4 *sub-matrices,* $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ *each of which has the dimension* 12×2 *and* $\mathbf{S}_4$ *with dimension* 12×4*, as illustrated in* (16)*. In other words,* $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ *correspond to the* $(\mathsf{m}+\mathsf{u}-1)\mathsf{K}_\mathrm{c} = 6$ *real demanded linear combinations of sub-messages, while* $\mathbf{S}_4$ *corresponds to the* $\mathsf{v} = 4$ *virtual demanded linear combinations of sub-messages.*

The proposed computing scheme in the computing phase contains three main steps:[7]

- *Step 1: We first choose the values for the elements in* $\mathbf{S}_4$*.*
- *Step 2: After determining* $\mathbf{S}_4$*, the constraints in* (18) *become linear in terms of the remaining variables (i.e., the elements in* $\mathbf{F}'_1, \mathbf{F}'_2, \mathbf{F}'_3, \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$*). Hence, we can obtain the values for these remaining variables by solving the systems of linear equations.*
- *Step 3: After determining all the variables, we check the constraints in* (19) *such that the proposed scheme is decodable.*

Step 1: We choose the values for $\mathbf{S}_4$ *with the following form,*

$$\mathbf{S}_4 = \begin{bmatrix} b_1^{1,1} & b_2^{1,1} & b_3^{1,1} & b_4^{1,1} \\ b_1^{1,2} & b_2^{1,2} & b_3^{1,2} & b_4^{1,2} \\ b_1^{2,1} & b_2^{2,1} & b_3^{2,1} & b_4^{2,1} \\ b_1^{2,2} & b_2^{2,2} & b_3^{2,2} & b_4^{2,2} \\ b_1^{3,1} & b_2^{3,1} & b_3^{3,1} & b_4^{3,1} \\ b_1^{3,2} & b_2^{3,2} & b_3^{3,2} & b_4^{3,2} \\ b_1^{4,1} & b_2^{4,1} & b_3^{4,1} & b_4^{4,1} \\ b_1^{4,2} & b_2^{4,2} & b_3^{4,2} & b_4^{4,2} \\ b_1^{5,1} & b_2^{5,1} & b_3^{5,1} & b_4^{5,1} \\ b_1^{5,2} & b_2^{5,2} & b_3^{5,2} & b_4^{5,2} \\ b_1^{6,1} & b_2^{6,1} & b_3^{6,1} & b_4^{6,1} \\ b_1^{6,2} & b_2^{6,2} & b_3^{6,2} & b_4^{6,2} \end{bmatrix} = \begin{bmatrix} * & * & 0 & 0 \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ 0 & 0 & * & * \\ * & * & 0 & 0 \\ 0 & 0 & * & * \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 2 & 2 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{20}$$

[7] Notice that the computing schemes in [20], [21] for the case $\mathsf{K}_\mathrm{c} = 1$ and in [5] for the case where $\mathsf{m} = 1$ cannot be used in this example to achieved the converse bound. The idea of the computing schemes in [20], [21] is first to randomly determine the elements in $\mathbf{S}$, and then to determine the coefficients of the virtually demanded linear combinations in $\mathbf{F}'$ in order to satisfy the constraints in (18). One can check that if we randomly choose all the elements in $\mathbf{S}$, there does not exist any solution on $\mathbf{F}'$ which satisfies the constraints in (18), because there will be more linearly independent constraints than the variables. The idea of the computing scheme in [5] is first to randomly determine the coefficients of the virtually demanded linear combinations in $\mathbf{F}'$, and then to determine the elements in $\mathbf{S}$ in order to satisfy the constraints in (18). However, if we randomly determine the coefficients of the virtually demanded linear combinations in $\mathbf{F}'$, the sub-matrix of $\mathbf{F}'$ including the columns corresponding to the sub-messages which each worker cannot compute has the dimension $\mathsf{v} \times (\mathsf{m}+\mathsf{u}-1)(\mathsf{N}_\mathrm{r}-\mathsf{m}) = 10 \times 9$. Hence, the left-hand side null-space of this sub-matrix only has one linearly independent vector; thus each worker can only transmit one linearly independent linear combination of sub-messages, where the coefficients of the unknown sub-messages are 0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

$$\mathbf{S}=\begin{bmatrix}\mathbf{s}^{1,1}\\ \mathbf{s}^{1,2}\\ \mathbf{s}^{2,1}\\ \mathbf{s}^{2,2}\\ \vdots\\ \mathbf{s}^{6,2}\end{bmatrix}=\underbrace{\begin{bmatrix}s_1^{1,1} & s_2^{1,1}\\ s_1^{1,2} & s_2^{1,2}\\ s_1^{2,1} & s_2^{2,1}\\ \vdots & \vdots\\ s_1^{6,2} & s_2^{6,2}\end{bmatrix}}_{\mathbf{S}_1}\underbrace{\begin{bmatrix}s_3^{1,1} & s_4^{1,1}\\ s_3^{1,2} & s_4^{1,2}\\ s_3^{2,1} & s_4^{2,1}\\ \vdots & \vdots\\ s_3^{6,2} & s_4^{6,2}\end{bmatrix}}_{\mathbf{S}_2}\underbrace{\begin{bmatrix}s_5^{1,1} & s_6^{1,1}\\ s_5^{1,2} & s_6^{1,2}\\ s_5^{2,1} & s_6^{2,1}\\ \vdots & \vdots\\ s_5^{6,2} & s_6^{6,2}\end{bmatrix}}_{\mathbf{S}_3}\underbrace{\begin{bmatrix}b_1^{1,1} & b_2^{1,1} & b_3^{1,1} & b_4^{1,1}\\ b_1^{1,2} & b_2^{1,2} & b_3^{1,2} & b_4^{1,2}\\ b_1^{2,1} & b_2^{2,1} & b_3^{2,1} & b_4^{2,1}\\ \vdots & \vdots & \vdots & \vdots\\ b_1^{6,2} & b_2^{6,2} & b_3^{6,2} & b_4^{6,2}\end{bmatrix}}_{\mathbf{S}_4}. \tag{16}$$

where each '$*$' represents a uniform i.i.d. symbol on $\mathbb{F}_q$. More precisely, for the first linear combination transmitted by each worker $n\in[6]$, we choose $b_1^{n,1}$ and $b_2^{n,1}$ uniformly i.i.d. over $\mathbb{F}_q$, while letting $b_3^{n,1}$ and $b_4^{n,1}$ be zero. For the second linear combination transmitted by each worker n, we choose $b_3^{n,2}$ and $b_4^{n,2}$ uniformly i.i.d. over $\mathbb{F}_q$, while letting $b_1^{n,2}$ and $b_2^{n,2}$ be zero. The above choice on $\mathbf{S}_4$ will guarantee that the constraints in (18) become linearly independent in terms of the remaining variables to be decided in the next step.[8]

Step 2: Let us focus on the constraints in (18) for $t=1$, which corresponds to the elements in $\mathbf{S}_1$ and $\mathbf{F}'_1$.

When $(t,j)=(1,1)$, the constraints in (18) become

$$s_1^{n,1}f_{1,k}+s_2^{n,1}f_{2,k}+b_1^{n,1}a_{1,k}+b_2^{n,1}a_{2,k}+b_3^{n,1}a_{3,k}+b_4^{n,1}a_{4,k}=0,\ \forall n\in[6],k\in\overline{\mathcal{Z}_n},$$

where $f_{1,k}$ represents the k^{th} element in the first demand vector, $f_{2,k}$ represents the k^{th} element in the second demand vector, and the values of $b_i^{n,1}$ where $i\in[4]$ have been chosen in (20). For example, if $n=1$, we have the set of datasets which are not assigned to worker 1 is $\overline{\mathcal{Z}_1}=\{4,5,6\}$. Hence, we have the following three constraints

$$\begin{aligned}&s_1^{1,1}f_{1,4}+s_2^{1,1}f_{2,4}+b_1^{1,1}a_{1,4}+b_2^{1,1}a_{2,4}+b_3^{1,1}a_{3,4}+b_4^{1,1}a_{4,4}\\&=1s_1^{1,1}+3s_2^{1,1}+0a_{1,4}+2a_{2,4}=0,\\&s_1^{1,1}f_{1,5}+s_2^{1,1}f_{2,5}+b_1^{1,1}a_{1,5}+b_2^{1,1}a_{2,5}+b_3^{1,1}a_{3,5}+b_4^{1,1}a_{4,5}\\&=1s_1^{1,1}+4s_2^{1,1}+0a_{1,5}+2a_{2,5}=0,\\&s_1^{1,1}f_{1,6}+s_2^{1,1}f_{2,6}+b_1^{1,1}a_{1,6}+b_2^{1,1}a_{2,6}+b_3^{1,1}a_{3,6}+b_4^{1,1}a_{4,6}\\&=1s_1^{1,1}+5s_2^{1,1}+0a_{1,6}+2a_{2,6}=0.\end{aligned}$$

Similarly, if $n=2$, with $\overline{\mathcal{Z}_2}=\{1,5,6\}$ we have the following three constraints

$$\begin{aligned}&s_1^{2,1}f_{1,1}+s_2^{2,1}f_{2,1}+b_1^{2,1}a_{1,1}+b_2^{2,1}a_{2,1}+b_3^{2,1}a_{3,1}+b_4^{2,1}a_{4,1}\\&=1s_1^{2,1}+0s_2^{2,1}+2a_{1,1}+2a_{2,1}=0,\\&s_1^{2,1}f_{1,5}+s_2^{2,1}f_{2,5}+b_1^{2,1}a_{1,5}+b_2^{2,1}a_{2,5}+b_3^{2,1}a_{3,5}+b_4^{2,1}a_{4,5}\\&=1s_1^{2,1}+4s_2^{2,1}+2a_{1,5}+2a_{2,5}=0,\\&s_1^{2,1}f_{1,6}+s_2^{2,1}f_{2,6}+b_1^{2,1}a_{1,6}+b_2^{2,1}a_{2,6}+b_3^{2,1}a_{3,6}+b_4^{2,1}a_{4,6}\\&=1s_1^{2,1}+5s_2^{2,1}+2a_{1,6}+2a_{2,6}=0.\end{aligned}$$

If $n=3$, with $\overline{\mathcal{Z}_3}=\{1,2,6\}$ we have the following three constraints

$$\begin{aligned}&s_1^{3,1}f_{1,1}+s_2^{3,1}f_{2,1}+b_1^{3,1}a_{1,1}+b_2^{3,1}a_{2,1}+b_3^{3,1}a_{3,1}+b_4^{3,1}a_{4,1}\\&=1s_1^{3,1}+0s_2^{3,1}+1a_{1,1}+2a_{2,1}=0,\\&s_1^{3,1}f_{1,2}+s_2^{3,1}f_{2,2}+b_1^{3,1}a_{1,2}+b_2^{3,1}a_{2,2}+b_3^{3,1}a_{3,2}+b_4^{3,1}a_{4,2}\\&=1s_1^{3,1}+1s_2^{3,1}+1a_{1,2}+2a_{2,2}=0,\\&s_1^{3,1}f_{1,6}+s_2^{3,1}f_{2,6}+b_1^{3,1}a_{1,6}+b_2^{3,1}a_{2,6}+b_3^{3,1}a_{3,6}+b_4^{3,1}a_{4,6}\\&=1s_1^{3,1}+5s_2^{3,1}+1a_{1,6}+2a_{2,6}=0.\end{aligned}$$

If $n=4$, with $\overline{\mathcal{Z}_4}=\{1,2,3\}$ we have the following three constraints

$$\begin{aligned}&s_1^{4,1}f_{1,1}+s_2^{4,1}f_{2,1}+b_1^{4,1}a_{1,1}+b_2^{4,1}a_{2,1}+b_3^{4,1}a_{3,1}+b_4^{4,1}a_{4,1}\\&=1s_1^{4,1}+0s_2^{4,1}+0a_{1,1}+1a_{2,1}=0,\\&s_1^{4,1}f_{1,2}+s_2^{4,1}f_{2,2}+b_1^{4,1}a_{1,2}+b_2^{4,1}a_{2,2}+b_3^{4,1}a_{3,2}+b_4^{4,1}a_{4,2}\\&=1s_1^{4,1}+1s_2^{4,1}+0a_{1,2}+1a_{2,2}=0,\\&s_1^{4,1}f_{1,3}+s_2^{4,1}f_{2,3}+b_1^{4,1}a_{1,3}+b_2^{4,1}a_{2,3}+b_3^{4,1}a_{3,3}+b_4^{4,1}a_{4,3}\\&=1s_1^{4,1}+2s_2^{4,1}+0a_{1,3}+1a_{2,3}=0.\end{aligned}$$

If $n=5$, with $\overline{\mathcal{Z}_5}=\{2,3,4\}$ we have the following three constraints

$$\begin{aligned}&s_1^{5,1}f_{1,2}+s_2^{5,1}f_{2,2}+b_1^{5,1}a_{1,2}+b_2^{5,1}a_{2,2}+b_3^{5,1}a_{3,2}+b_4^{5,1}a_{4,2}\\&=1s_1^{5,1}+1s_2^{5,1}+1a_{1,2}+0a_{2,2}=0,\\&s_1^{5,1}f_{1,3}+s_2^{5,1}f_{2,3}+b_1^{5,1}a_{1,3}+b_2^{5,1}a_{2,3}+b_3^{5,1}a_{3,3}+b_4^{5,1}a_{4,3}\\&=1s_1^{5,1}+2s_2^{5,1}+1a_{1,3}+0a_{2,3}=0,\\&s_1^{5,1}f_{1,4}+s_2^{5,1}f_{2,4}+b_1^{5,1}a_{1,4}+b_2^{5,1}a_{2,4}+b_3^{5,1}a_{3,4}+b_4^{5,1}a_{4,4}\\&=1s_1^{5,1}+3s_2^{5,1}+1a_{1,4}+0a_{2,4}=0.\end{aligned}$$

If $n=6$, with $\overline{\mathcal{Z}_6}=\{3,4,5\}$ we have the following three constraints

$$\begin{aligned}&s_1^{6,1}f_{1,3}+s_2^{6,1}f_{2,3}+b_1^{6,1}a_{1,3}+b_2^{6,1}a_{2,3}+b_3^{6,1}a_{3,3}+b_4^{6,1}a_{4,3}\\&=1s_1^{6,1}+2s_2^{6,1}+2a_{1,3}+2a_{2,3}=0,\\&s_1^{6,1}f_{1,4}+s_2^{6,1}f_{2,4}+b_1^{6,1}a_{1,4}+b_2^{6,1}a_{2,4}+b_3^{6,1}a_{3,4}+b_4^{6,1}a_{4,4}\\&=1s_1^{6,1}+3s_2^{6,1}+2a_{1,4}+2a_{2,4}=0,\\&s_1^{6,1}f_{1,5}+s_2^{6,1}f_{2,5}+b_1^{6,1}a_{1,5}+b_2^{6,1}a_{2,5}+b_3^{6,1}a_{3,5}+b_4^{6,1}a_{4,5}\\&=1s_1^{6,1}+4s_2^{6,1}+2a_{1,5}+2a_{2,5}=0.\end{aligned}$$

Hence, there are totally $6\times3=18$ constraints on 24 variables, which are

$$a_{1,1},\ldots,a_{1,6},a_{2,1},\ldots,a_{2,6},s_1^{1,1},s_2^{1,1},s_1^{2,1},s_2^{2,1},\ldots,s_2^{6,1}. \tag{21}$$

[8] Note that we can also choose each element in $\mathbf{S}_4$ uniformly i.i.d. over $\mathbb{F}_q$ to find a realization of $\mathbf{S}_4$ which leads to these linearly independences. However, by the Schwartz-Zippel lemma [25]–[27], the probability to obtain a 'good' choice of $\mathbf{S}_4$ decreases, since the total degree of the corresponding polynomial in the Schwartz-Zippel lemma increases.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

Since the number of variables is more than the number of constraints, we fix $24-18=6$ *variables. More precisely, we give a value uniformly i.i.d. over* $\mathbb{F}_{\mathsf{q}}$ *to each of the following* 6 *variables (the positions of these* 6 *variables are found through programming),*

$$s_1^{1,1}=0,\ s_2^{2,1}=1,\ s_1^{3,1}=1,\ s_2^{4,1}=1,\ s_1^{5,1}=0,\ s_2^{6,1}=1. \tag{22}$$

After determining the 6 *variables in* (22)*, the above* 18 *constraints are linearly independent on the remaining* 18 *variables, such that by solving a system of linear equations we have*

$$a_{1,1}=1/4, a_{1,2}=5/8, a_{1,3}=5/4, a_{1,4}=15/8, a_{1,5}=21/8,$$
$$a_{1,6}=27/8, a_{2,1}=-5/8, a_{2,2}=-13/8, a_{2,3}=-21/8,$$
$$a_{2,4}=-15/4, a_{2,5}=-5, a_{2,6}=-25/4, s_2^{1,1}=5/2,$$
$$s_1^{2,1}=3/4, s_2^{3,1}=13/8, s_1^{4,1}=5/8, s_2^{5,1}=-5/8, s_1^{6,1}=3/4.$$

Note that for any element a *on* $\mathbb{F}_{\mathsf{q}}$*,* $1/a$ *represents the multiplicative inverse of* a *on* $\mathbb{F}_{\mathsf{q}}$*.*

Similarly, by considering all pairs (t,j) *where* $t\in[3]$ *and* $j\in[2]$*, we can determine* (23)*.*

Step 3: For each subset of workers $\mathcal{A}\subseteq[6]$ *where* $|\mathcal{A}|=5$*, it can be seen that the constraints in* (19) *holds. For example, if* $\mathcal{A}=[5]$*, the sub-matrix* $\mathbf{S}^{([10])_{\mathsf{r}}}$ *including the first* 10 *rows of* $\mathbf{S}$ *is full rank. Hence, we let each worker* $n\in[\mathsf{N}]$ *compute and send two linear combinations of sub-messages,* $\mathbf{s}^{n,1}\mathbf{F}'[W_{1,1};\ldots;W_{6,3}]$ *and* $\mathbf{s}^{n,2}\mathbf{F}'[W_{1,1};\ldots;W_{6,3}]$*.*

Decoding phase: Assume that the set of responding workers is $\mathcal{A}$ *where* $\mathcal{A}\subseteq[6]$ *and* $|\mathcal{A}|=5$*. The master receives*

$$\mathbf{X}_{\mathcal{A}}=\left[\mathbf{s}^{\mathcal{A}(1),1};\mathbf{s}^{\mathcal{A}(1),2};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(5),2}\right]\mathbf{F}'\ [W_{1,1};\ldots;W_{6,1};W_{1,2};\ldots;W_{6,3}].$$

Since $\left[\mathbf{s}^{\mathcal{A}(1),1};\mathbf{s}^{\mathcal{A}(1),2};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(5),2}\right]$ *is full rank, the master then computes*

$$\left[\mathbf{s}^{\mathcal{A}(1),1};\mathbf{s}^{\mathcal{A}(1),2};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(5),2}\right]^{-1}\mathbf{X}_{\mathcal{A}}$$

to obtain $\mathbf{F}'\ [W_{1,1};\ldots;W_{6,1};W_{1,2};\ldots;W_{6,3}]$*, which contains its demanded linear combinations.*

Performance: Since each worker sends $\frac{2\mathsf{L}}{3}$ *symbols, the communication cost is* $\frac{10\mathsf{L}}{3\mathsf{L}}=\frac{10}{3}$*, coinciding with the converse bound in* (4b)*.* □

We are ready to generalize the proposed distributed computing scheme in Example 2. First we focus on $\mathsf{K}_{\mathrm{c}}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}$, where $\mathsf{u}\in[\mathsf{N}_{\mathrm{r}}-\mathsf{m}+1]$ and $\mathsf{N}\geq\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N}_{\mathrm{r}}-\mathsf{m}-\mathsf{u}+1)$. During the data assignment phase, we use the cyclic assignment.

Computing phase: Since the communication cost is no less than $\mathsf{N}_{\mathrm{r}}\frac{\mathsf{K}_{\mathrm{c}}}{\mathsf{m}+\mathsf{u}-1}$, from the converse bound (4b), we divide each message W_k where $k\in[\mathsf{K}]$ into $\mathsf{m}+\mathsf{u}-1$ non-overlapping and equal-length sub-messages, $W_k=\{W_{k,j}: j\in[\mathsf{m}+\mathsf{u}-1]\}$. Hence, the task function becomes $(\mathsf{m}+\mathsf{u}-1)\mathsf{K}_{\mathrm{c}}$ linear combinations of sub-messages. Each worker should send K_{c} linear combinations of sub-messages. From the answers of N_{r} workers, the master totally receives $\mathsf{N}_{\mathrm{r}}\mathsf{K}_{\mathrm{c}}$ linear combinations of sub-messages. Hence, we generate

$$\mathsf{v}=\mathsf{N}_{\mathrm{r}}\mathsf{K}_{\mathrm{c}}-(\mathsf{m}+\mathsf{u}-1)\mathsf{K}_{\mathrm{c}}=\mathsf{K}_{\mathrm{c}}(\mathsf{N}_{\mathrm{r}}-\mathsf{m}-\mathsf{u}+1)$$

virtually requested linear combinations of sub-messages; thus the effective demand matrix $\mathbf{F}'$ has the dimension $\mathsf{N}_{\mathrm{r}}\mathsf{K}_{\mathrm{c}}\times\mathsf{K}(\mathsf{m}+\mathsf{u}-1)$, with the form in (24).

The transmissions of the K workers can be expressed as

$$\mathbf{S}\ \mathbf{F}'\ [W_{1,1};\ldots;W_{\mathsf{K},1};W_{1,2};\ldots;W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}]=[\mathbf{s}^{1,1};\ldots;\mathbf{s}^{1,\mathsf{K}_{\mathrm{c}}};\mathbf{s}^{2,1};\ldots;\mathbf{s}^{\mathsf{N},\mathsf{K}_{\mathrm{c}}}]\ \mathbf{F}'\ [W_{1,1};\ldots;W_{\mathsf{K},1};W_{1,2};\ldots;W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}],$$

where $\mathbf{s}^{n,j}\mathbf{F}'[W_{1,1};\ldots;W_{\mathsf{K},1};W_{1,2};\ldots;W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}]$ represents the j^{th} transmitted linear combination by worker n. We can further expand $\mathbf{S}$ as in (25),

By defining $\mathbf{d}_i$ as the i^{th} column of $\mathbf{F}'$, the j^{th} transmitted linear combination by worker n can be expressed as

$$\mathbf{s}^{n,j}\mathbf{d}_1W_{1,1}+\cdots+\mathbf{s}^{n,j}\mathbf{d}_{\mathsf{K}}W_{\mathsf{K},1}+\mathbf{s}^{n,j}\mathbf{d}_{\mathsf{K}+1}W_{1,2}+\cdots+\mathbf{s}^{n,j}\mathbf{d}_{(\mathsf{m}+\mathsf{u}-1)\mathsf{K}}W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}. \tag{26}$$

To guarantee that the linear combination in (26) can be transmitted by worker n, the coefficients of the sub-messages which worker n cannot compute should be 0; that is

$$\mathbf{s}^{n,j}\mathbf{d}_{k+(t-1)\mathsf{K}}=0,\ \forall n\in[\mathsf{N}],j\in[\mathsf{K}_{\mathrm{c}}],t\in[\mathsf{m}+\mathsf{u}-1],k\in\overline{\mathcal{Z}_n}. \tag{27}$$

In addition, for each set $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_{\mathrm{r}}$, by receiving the linear combinations transmitted by the workers in $\mathcal{A}$, the master should recover the desired linear combinations. Hence, we should have

$$[\mathbf{s}^{\mathcal{A}(1),1};\ldots;\mathbf{s}^{\mathcal{A}(1),\mathsf{K}_{\mathrm{c}}};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(\mathsf{N}_{\mathrm{r}}),\mathsf{K}_{\mathrm{c}}}]\text{ is full rank,} \tag{28}$$

$\forall\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_{\mathrm{r}}$. Our objective is to determine the elements in $\mathbf{S}$ (i.e., $s_i^{n,j}$ where $n\in[\mathsf{N}]$, $j\in[\mathsf{K}_{\mathrm{c}}]$, $i\in[(\mathsf{m}+\mathsf{u}-1)\mathsf{K}_{\mathrm{c}}]$; $b_i^{n,j}$ where $n\in[\mathsf{N}]$, $j\in[\mathsf{K}_{\mathrm{c}}]$, $i\in[\mathsf{v}]$) and in $\mathbf{F}'$ (i.e., $a_{i,k}$ where $i\in[\mathsf{v}]$ and $k\in[(\mathsf{m}+\mathsf{u}-1)\mathsf{K}]$) such that the constraints in (27) and (28) are satisfied.

We divide matrix $\mathbf{F}'$ into $\mathsf{m}+\mathsf{u}-1$ sub-matrices, $\mathbf{F}'_1,\ldots,\mathbf{F}'_{\mathsf{m}+\mathsf{u}-1}$ each of which has the dimension $\mathsf{N}_{\mathrm{r}}\mathsf{K}_{\mathrm{c}}\times\mathsf{K}$, as illustrated in (24). We also divide matrix $\mathbf{S}$ into $\mathsf{m}+\mathsf{u}$ sub-matrices, $\mathbf{S}_1,\ldots,\mathbf{S}_{\mathsf{m}+\mathsf{u}-1}$ each of which has the dimension $\mathsf{N}\mathsf{K}_{\mathrm{c}}\times\mathsf{K}_{\mathrm{c}}$ and $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ with dimension $\mathsf{N}\mathsf{K}_{\mathrm{c}}\times\mathsf{v}$, as illustrated in (25). As in Example 2, the proposed computing scheme contains three main steps:

- Step 1: We first choose the values for the elements in $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$.
- Step 2: After determining the elements in $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$, the constraints in (27) become linear in terms of the remaining variables, which are then determined by solving the systems of linear equations.
- Step 3: After determining all the variables, we check the constraints in (28) such that the proposed scheme is decodable.

Step 1: We choose the values for $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ with the form in (29b) (as $\mathbf{S}_4$ in Example 2), where each '$*$' represents a uniformly i.i.d. symbol on $\mathbb{F}_{\mathsf{q}}$. More precisely, for the j^{th} linear combination transmitted by worker n where $j\in[\mathsf{K}_{\mathrm{c}}]$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

$$\mathbf{S}=\begin{bmatrix} 0 & 5/2 & 0 & 0 & 0 & -11/4 & 0 & 2 & 0 & 0\\ 1 & -14 & 1 & 27 & 0 & 0 & 0 & 0 & 2 & 0\\ 3/4 & 1 & 0 & 0 & 41/8 & 1 & 2 & 2 & 0 & 0\\ 40 & 0 & -82 & 1 & 0 & 0 & 0 & 0 & 0 & 2\\ 1 & 13/8 & 0 & 0 & 1 & -9/16 & 1 & 2 & 0 & 0\\ 1 & -10 & 0 & 39/2 & 0 & 0 & 0 & 0 & 2 & 1\\ 5/8 & 1 & 0 & 0 & -25/16 & 0 & 0 & 1 & 0 & 0\\ -19/2 & 0 & 41/2 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & -5/8 & 0 & 0 & 1 & 41/16 & 1 & 0 & 0 & 0\\ 1 & -10 & 1 & 37/2 & 0 & 0 & 0 & 0 & 2 & 1\\ 3/4 & 1 & 0 & 0 & 73/8 & 0 & 2 & 2 & 0 & 0\\ -23/2 & 1 & 31/2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}; \tag{23a}$$

$$[a_{1,1},\ldots,a_{1,18}]=\left[\frac{1}{4},\frac{5}{8},\frac{5}{4},\frac{15}{8},\frac{21}{8},\frac{27}{8},0,0,0,0,0,0,\frac{-33}{8},\frac{-57}{16},\frac{-49}{8},\frac{139}{16},\frac{161}{16},\frac{-191}{16}\right]; \tag{23b}$$

$$[a_{2,1},\ldots,a_{2,18}]=\left[\frac{-5}{8},\frac{-13}{8},\frac{-21}{8},\frac{-15}{4},-5,\frac{-25}{4},0,0,0,0,0,0,\frac{25}{16},\frac{25}{16},\frac{25}{16},\frac{33}{8},\frac{11}{2},\frac{55}{8}\right]; \tag{23c}$$

$$[a_{3,1},\ldots,a_{3,18}]=\left[\frac{19}{2},\frac{19}{2},\frac{19}{2},\frac{41}{2},\frac{55}{2},\frac{69}{2},\frac{-41}{2},\frac{-43}{2},\frac{-45}{2},-41,\frac{-109}{2},-68,0,0,0,0,0,0\right]; \tag{23d}$$

$$[a_{4,1},\ldots,a_{4,18}]=\left[-20,-10,0,-12,-20,-20,41,\frac{47}{2},7,\frac{51}{2},39,\frac{77}{2},0,0,0,0,0,0\right]. \tag{23e}$$

$$\mathbf{F}'=\begin{bmatrix} f_{1,1} & \cdots & f_{1,\mathsf{K}} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ f_{\mathsf{K_c},1} & \cdots & f_{\mathsf{K_c},\mathsf{K}} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0\\ 0 & \cdots & 0 & f_{1,1} & \cdots & f_{1,\mathsf{K}} & \cdots & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & f_{\mathsf{K_c},1} & \cdots & f_{\mathsf{K_c},\mathsf{K}} & \cdots & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & f_{1,1} & \cdots & f_{1,\mathsf{K}}\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & f_{\mathsf{K_c},1} & \cdots & f_{\mathsf{K_c},\mathsf{K}}\\ a_{1,1} & \cdots & a_{1,\mathsf{K}} & a_{1,\mathsf{K}+1} & \cdots & a_{1,2\mathsf{K}} & \cdots & a_{1,(\mathsf{m}+\mathsf{u}-2)\mathsf{K}+1} & \cdots & a_{1,(\mathsf{m}+\mathsf{u}-1)\mathsf{K}}\\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ a_{\mathsf{v},1} & \cdots & a_{\mathsf{v},\mathsf{K}} & a_{\mathsf{v},\mathsf{K}+1} & \cdots & a_{\mathsf{v},2\mathsf{K}} & \cdots & a_{\mathsf{v},(\mathsf{m}+\mathsf{u}-2)\mathsf{K}+1} & \cdots & a_{\mathsf{v},(\mathsf{m}+\mathsf{u}-1)\mathsf{K}} \end{bmatrix}. \tag{24}$$

$\mathbf{F'}_1$ $\quad$ $\mathbf{F'}_2$ $\quad$ $\mathbf{F'}_{\mathsf{m}+\mathsf{u}-1}$

$$\mathbf{S}=\begin{bmatrix} \mathbf{s}^{1,1}\\ \vdots\\ \mathbf{s}^{1,\mathsf{K_c}}\\ \mathbf{s}^{2,1}\\ \vdots\\ \mathbf{s}^{\mathsf{N},\mathsf{K_c}} \end{bmatrix}=\begin{bmatrix} s_1^{1,1} & \cdots & s_{\mathsf{K_c}}^{1,1} & \cdots & s_{(\mathsf{m}+\mathsf{u}-2)\mathsf{K_c}+1}^{1,1} & \cdots & s_{(\mathsf{m}+\mathsf{u}-1)\mathsf{K_c}}^{1,1} & b_1^{1,1} & \cdots & b_{\mathsf{v}}^{1,1}\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ s_1^{1,\mathsf{K_c}} & \cdots & s_{\mathsf{K_c}}^{1,\mathsf{K_c}} & \cdots & s_{(\mathsf{m}+\mathsf{u}-2)\mathsf{K_c}+1}^{1,\mathsf{K_c}} & \cdots & s_{(\mathsf{m}+\mathsf{u}-1)\mathsf{K_c}}^{1,\mathsf{K_c}} & b_1^{1,\mathsf{K_c}} & \cdots & b_{\mathsf{v}}^{1,\mathsf{K_c}}\\ s_1^{2,1} & \cdots & s_{\mathsf{K_c}}^{2,1} & \cdots & s_{(\mathsf{m}+\mathsf{u}-2)\mathsf{K_c}+1}^{2,1} & \cdots & s_{(\mathsf{m}+\mathsf{u}-1)\mathsf{K_c}}^{2,1} & b_1^{2,1} & \cdots & b_{\mathsf{v}}^{2,1}\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\ s_1^{\mathsf{N},\mathsf{K_c}} & \cdots & s_{\mathsf{K_c}}^{\mathsf{N},\mathsf{K_c}} & \cdots & s_{(\mathsf{m}+\mathsf{u}-2)\mathsf{K_c}+1}^{\mathsf{N},\mathsf{K_c}} & \cdots & s_{(\mathsf{m}+\mathsf{u}-1)\mathsf{K_c}}^{\mathsf{N},\mathsf{K_c}} & b_1^{\mathsf{N},\mathsf{K_c}} & \cdots & b_{\mathsf{v}}^{\mathsf{N},\mathsf{K_c}} \end{bmatrix}. \tag{25}$$

$\mathbf{S}_1$ $\quad$ $\mathbf{S}_{\mathsf{m}+\mathsf{u}-1}$ $\quad$ $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

$$\mathbf{S}_{\mathsf{m}+\mathsf{u}} = \begin{bmatrix} b_1^{1,1} & \cdots & b_{\frac{\mathsf{v}}{\mathsf{K_c}}}^{1,1} & b_{\frac{\mathsf{v}}{\mathsf{K_c}}+1}^{1,1} & \cdots & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}}^{1,1} & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}+1}^{1,1} & \cdots & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}}^{1,1} & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}+1}^{1,1} & \cdots & b_{\mathsf{v}}^{1,1} \\ b_1^{1,2} & \cdots & b_{\frac{\mathsf{v}}{\mathsf{K_c}}}^{1,2} & b_{\frac{\mathsf{v}}{\mathsf{K_c}}+1}^{1,2} & \cdots & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}}^{1,2} & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}+1}^{1,2} & \cdots & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}}^{1,2} & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}+1}^{1,2} & \cdots & b_{\mathsf{v}}^{1,2} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_1^{1,\mathsf{K_c}} & \cdots & b_{\frac{\mathsf{v}}{\mathsf{K_c}}}^{1,\mathsf{K_c}} & b_{\frac{\mathsf{v}}{\mathsf{K_c}}+1}^{1,\mathsf{K_c}} & \cdots & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}}^{1,\mathsf{K_c}} & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}+1}^{1,\mathsf{K_c}} & \cdots & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}}^{1,\mathsf{K_c}} & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}+1}^{1,\mathsf{K_c}} & \cdots & b_{\mathsf{v}}^{1,\mathsf{K_c}} \\ b_1^{2,1} & \cdots & b_{\frac{\mathsf{v}}{\mathsf{K_c}}}^{2,1} & b_{\frac{\mathsf{v}}{\mathsf{K_c}}+1}^{2,1} & \cdots & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}}^{2,1} & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}+1}^{2,1} & \cdots & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}}^{2,1} & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}+1}^{2,1} & \cdots & b_{\mathsf{v}}^{2,1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ b_1^{\mathsf{N},\mathsf{K_c}} & \cdots & b_{\frac{\mathsf{v}}{\mathsf{K_c}}}^{\mathsf{N},\mathsf{K_c}} & b_{\frac{\mathsf{v}}{\mathsf{K_c}}+1}^{\mathsf{N},\mathsf{K_c}} & \cdots & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}}^{\mathsf{N},\mathsf{K_c}} & b_{\frac{2\mathsf{v}}{\mathsf{K_c}}+1}^{\mathsf{N},\mathsf{K_c}} & \cdots & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}}^{\mathsf{N},\mathsf{K_c}} & b_{\frac{(\mathsf{K_c}-1)\mathsf{v}}{\mathsf{K_c}}+1}^{\mathsf{N},\mathsf{K_c}} & \cdots & b_{\mathsf{v}}^{\mathsf{N},\mathsf{K_c}} \end{bmatrix} \tag{29a}$$

$$= \begin{bmatrix} * & \cdots & * & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & * & \cdots & * & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & * & \cdots & * \\ * & \cdots & * & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & * & \cdots & * \end{bmatrix}, \tag{29b}$$

and $n \in [\mathsf{N}]$, we choose each of $b_{\frac{(j-1)\mathsf{v}}{\mathsf{K_c}}+1}^{n,j}, \ldots, b_{\frac{j\mathsf{v}}{\mathsf{K_c}}}^{n,j}$ uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$, while setting the other variables in this linear combination be 0. The above choice on $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ will guarantee that the constraints in (27) become linearly independent in terms of the remaining variables to be determined in the next step.

Step 2: We then fix one $t \in [\mathsf{m}+\mathsf{u}-1]$ and one $j \in [\mathsf{K_c}]$; thus the constraints in (27) become

$$0 = \mathbf{s}^{n,j}\mathbf{d}_{k+(t-1)\mathsf{K}} \tag{30a}$$

$$= \sum_{i_1 \in [\mathsf{K_c}]} f_{i_1,k}\ s_{(t-1)\mathsf{K_c}+i_1}^{n,j} + \sum_{i_2 \in [\mathsf{v}]} b_{i_2}^{n,j}\ a_{i_2,(t-1)\mathsf{K}+k} \tag{30b}$$

$$= \sum_{i_1 \in [\mathsf{K_c}]} f_{i_1,k}\ s_{(t-1)\mathsf{K_c}+i_1}^{n,j} + \sum_{i_3 \in \left[\frac{(j-1)\mathsf{v}}{\mathsf{K_c}}+1:\frac{j\mathsf{v}}{\mathsf{K_c}}\right]} b_{i_3}^{n,j}\ a_{i_3,(t-1)\mathsf{K}+k},\ \forall n \in [\mathsf{N}], k \in \overline{\mathcal{Z}_n}. \tag{30c}$$

Notice that in (30c) the coefficients $f_{i_1,k}$ are the elements in the demand matrix $\mathbf{F}$ and $b_{i_3}^{n,j}$ have been already determined in Step 1. Hence, the constraints (30c) are linear in terms of the variables

$$s_{(t-1)\mathsf{K_c}+i_1}^{n,j} \text{ and } a_{i_3,k_1},\quad \forall n \in [\mathsf{N}], i_1 \in [\mathsf{K_c}],$$
$$i_3 \in \left[\frac{(j-1)\mathsf{v}}{\mathsf{K_c}}+1 : \frac{j\mathsf{v}}{\mathsf{K_c}}\right], k_1 \in [(t-1)\mathsf{K}+1 : t\mathsf{K}]. \tag{31}$$

Next, we determine the values of the variables in (31) by solving the system of linear equations. In (31), there are totally

$$\mathsf{N}\mathsf{K_c} + \frac{\mathsf{v}}{\mathsf{K_c}}\mathsf{K} = \mathsf{N}\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u} + (\mathsf{N_r}-\mathsf{m}-\mathsf{u}+1)\mathsf{K} = \mathsf{K}(\mathsf{N_r}-\mathsf{m}+1)$$

variables while in (30c) there are totally

$$\mathsf{N}\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}) = \mathsf{K}(\mathsf{N_r}-\mathsf{m})$$

constraints. Hence, in order to determine all the variables in (31) while satisfying the constraints in (30c), we first fix $\mathsf{K}(\mathsf{N_r}-\mathsf{m}+1) - \mathsf{K}(\mathsf{N_r}-\mathsf{m}) = \mathsf{K}$ variables. More precisely, for each $n \in [\mathsf{N}]$ and each $i \in [\mathsf{K}/\mathsf{N}]$, we first choose each of

$$s_{(t-1)\mathsf{K_c}+(i-1)\mathsf{u}+(n \bmod \mathsf{u})}^{n,j}, \tag{32}$$

uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. Note that $s_{(t-1)\mathsf{K_c}+(i-1)\mathsf{u}+(n \bmod \mathsf{u})}^{n,j}$ is in the $\big((n-1)\mathsf{K_c}+j\big)^{\text{th}}$ row and the $\big((t-1)\mathsf{K_c}+(i-1)\mathsf{u}+(n \bmod \mathsf{u})\big)^{\text{th}}$ column of $\mathbf{S}$. Hence, among all the $\mathsf{K}(\mathsf{N_r}-\mathsf{m}+1)$ variables in (31), we have determined $\mathsf{N}\frac{\mathsf{K}}{\mathsf{N}} = \mathsf{K}$ variables. Thus there are $\mathsf{K}(\mathsf{N_r}-\mathsf{m})$ variables to be solved by $\mathsf{K}(\mathsf{N_r}-\mathsf{m})$ linear equations in (30c). It will be proved by the Schwartz-Zippel Lemma [25]–[27] in Appendix A that with high probability, these $\mathsf{K}(\mathsf{N_r}-\mathsf{m})$ linear equations are linearly independent over these remaining $\mathsf{K}(\mathsf{N_r}-\mathsf{m})$ variables.[9] As a result, by solving the system of linear equations we can determine all the remaining variables in (31).

By considering all the pairs (t, j) where $t \in [\mathsf{m}+\mathsf{u}-1]$ and $j \in [\mathsf{K_c}]$, we can determine all the elements in $\mathbf{S}$ and $\mathbf{F}'$.

Step 3: It will be proved by the Schwartz-Zippel Lemma [25]–[27] in Appendix A that the constraints in (28) hold with high probability. Hence, we let each worker n compute and send $\mathsf{K_c}$ linear combinations, i.e., $\mathbf{s}^{n,j}\mathbf{F}'[W_{1,1}; \ldots; W_{\mathsf{K},1}; W_{1,2}; \ldots; W_{\mathsf{K},\mathsf{m}+\mathsf{t}-1}]$ where $j \in [\mathsf{K_c}]$.

Decoding phase: Assume that the set of responding workers is $\mathcal{A}$ where $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = \mathsf{N_r}$. The master receives

$$\mathbf{X}_{\mathcal{A}} = [\mathbf{s}^{\mathcal{A}(1),1}; \ldots; \mathbf{s}^{\mathcal{A}(1),\mathsf{K_c}}; \mathbf{s}^{\mathcal{A}(2),1}; \ldots; \mathbf{s}^{\mathcal{A}(\mathsf{N_r}),\mathsf{K_c}}]$$
$$\mathbf{F}'\ [W_{1,1}; \ldots; W_{\mathsf{K},1}; W_{1,2}; \ldots; W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}].$$

[9] Note that in Example 2, we focus on a specific demand and thus the Schwartz-Zippel Lemma is not needed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2021.3107432, IEEE Transactions on Communications

Since $[\mathbf{s}^{\mathcal{A}(1),1};\ldots;\mathbf{s}^{\mathcal{A}(1),\mathsf{K_c}};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(\mathsf{N_r}),\mathsf{K_c}}]$ is full rank, the master then computes

$$[\mathbf{s}^{\mathcal{A}(1),1};\ldots;\mathbf{s}^{\mathcal{A}(1),\mathsf{K_c}};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(\mathsf{N_r}),\mathsf{K_c}}]^{-1}\mathbf{X}_{\mathcal{A}}$$

to obtain $\mathbf{F}'[W_{1,1};\ldots;W_{\mathsf{K},1};W_{1,2};\ldots;W_{\mathsf{K},\mathsf{m}+\mathsf{u}-1}]$, which contains its demanded linear combinations.

Performance: Since each worker sends $\frac{\mathsf{K_c}\mathsf{L}}{\mathsf{m}+\mathsf{u}-1}$ symbols, the communication cost is $\frac{\mathsf{N_r}\mathsf{K_c}\mathsf{L}}{(\mathsf{m}+\mathsf{u}-1)\mathsf{L}}=\frac{\mathsf{N_r}\mathsf{K_c}}{\mathsf{m}+\mathsf{u}-1}$, coinciding with (6a).

Remark 3. *The proposed scheme works for the case where*

$$\mathsf{N}\geq\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N_r}-\mathsf{m}-\mathsf{u}+1),\tag{33}$$

which can be explained intuitively in the following way. It will be proved in Appendix A that if the proposed scheme works for the $\big(\mathsf{N},\mathsf{N},\mathsf{N_r},\mathsf{u},\mathsf{m}\big)$ *distributed linearly separable computation problem (i.e., the number of messages is equal to* N*) with high probability, then with high probability the proposed scheme also works for the* $\big(\mathsf{K},\mathsf{N},\mathsf{N_r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\big)$ *distributed linearly separable computation problem where* N *divides* K*. Hence, let us then analyse the case* $\mathsf{K}=\mathsf{N}$*. In this case, note that* $\mathsf{K_c}=\mathsf{u}$*.*

We fix one $t\in[\mathsf{m}+\mathsf{u}-1]$ *in the constraints* (27)*. In Step 2 of the computing phase, we should solve the following problem:*

Problem t*: Determine the values of the variables*

$$s^{n,j}_{(t-1)\mathsf{u}+i_1}\text{ and }a_{i_3,k},\ \forall n\in[\mathsf{N}],j\in[\mathsf{u}],i_1\in[\mathsf{u}],$$
$$i_3\in[\mathsf{v}],k\in[(t-1)\mathsf{K}:t\mathsf{K}],$$

satisfying the constraints

$$\sum_{i_1\in[\mathsf{u}]}f_{i_1,k}\ s^{n,j}_{(t-1)\mathsf{u}+i_1}+\sum_{i_3\in\left[\frac{(j-1)\mathsf{v}}{\mathsf{u}}+1:\frac{j\mathsf{v}}{\mathsf{u}}\right]}b^{n,j}_{i_3}\ a_{i_3,(t-1)\mathsf{K}+k}=0,$$

$\forall j\in[\mathsf{u}],n\in[\mathsf{N}],k\in\overline{\mathcal{Z}_n}$.

Notice that by solving Problem t*, for each* $i\in[\mathsf{v}]$*, we can determine*

$$[s^{1,1}_{(t-1)\mathsf{u}+i};\ldots;s^{1,\mathsf{u}}_{(t-1)\mathsf{u}+i};s^{2,1}_{(t-1)\mathsf{u}+i};\ldots;s^{\mathsf{N},\mathsf{u}}_{(t-1)\mathsf{u}+i}],$$

which is the $((t-1)\mathsf{u}+i)^{th}$ *column of* $\mathbf{S}$*. Another important observation is that, Problem* t_1 *is totally equivalent to Problem* t_2 *for any* $t_1\neq t_2$*. Thus, we can introduce the following unified problem.*

Unified Problem: Determine the values of the variables

$$p^{n,j}_{i_1}\text{ and }q_{i_3,k},\ \forall n\in[\mathsf{N}],j\in[\mathsf{u}],i_1\in[\mathsf{u}],i_3\in[\mathsf{v}],k\in[\mathsf{K}],$$

satisfying the constraints

$$\sum_{i_1\in[\mathsf{u}]}f_{i_1,k}\ p^{n,j}_{i_1}+\sum_{i_3\in\left[\frac{(j-1)\mathsf{v}}{\mathsf{u}}+1:\frac{j\mathsf{v}}{\mathsf{u}}\right]}b^{n,j}_{i_3}\ q_{i_3,k}=0,\tag{34}$$

$\forall j\in[\mathsf{u}],n\in[\mathsf{N}],k\in\overline{\mathcal{Z}_n}$.

In the unified problem, there are

$$\mathsf{Nuu}+\mathsf{vK}=\mathsf{Nu}(\mathsf{u}+\mathsf{N_r}-\mathsf{m}-\mathsf{u}+1)=\mathsf{Nu}(\mathsf{N_r}-\mathsf{m}+1)$$

variables and $\mathsf{Nu}(\mathsf{N_r}-\mathsf{m})$ *constraints. Hence, the number of linearly independent solutions of the unified problem is no less than* $\mathsf{Nu}(\mathsf{N_r}-\mathsf{m}+1)-\mathsf{Nu}(\mathsf{N_r}-\mathsf{m})=\mathsf{Nu}$*, where the equality holds when the constraints in the unified problem is linearly independent. To guarantee that all the columns in* $\mathbf{S}$ *are linearly independent, we should assign* $\mathsf{m}+\mathsf{u}-1$ *linearly independent solutions to Problems* $1,2,\ldots,\mathsf{m}+\mathsf{u}-1$*.*

In addition, among all the linearly independent solutions of the unified problem, there are uv *trivial solutions which we cannot pick. More precisely, for each* $i\in[\mathsf{v}]$ *and* $d\in[\mathsf{u}]$*, one possible solution is to set (recall that* $\mathbf{f}_d$ *represents the* d^{th} *demand vector)*

$$(q_{i,1},q_{i,2},\ldots,q_{i,\mathsf{K}})=\mathbf{f}_d,$$

while setting $q_{i_3,k}=0$ *if* $i_3\neq i$*. In addition, we set*

$$p^{n,j}_i=-b^{n,j}_i,\ \forall n\in[\mathsf{N}],j\in[\mathsf{u}],$$

while setting $p^{n,j}_{i_1}=0$ *if* $i_1\neq i$*. It can be easily checked that by the above choice of variables, the constraints in* (34) *hold. Hence, the above choice is one possible solution of the unified problem. There are totally* uv *such possible solutions. However, any combination of such* uv *solutions cannot be chosen as a solution of Problem* t*. This is because in each of the above solutions, there is a column of* $\mathbf{S}$ *(i.e.,* $[p^{1,1}_i;\ldots;p^{1,\mathsf{u}}_i;p^{1,2}_i;\ldots;p^{\mathsf{N},\mathsf{u}}_i]$*), which can be expressed by a fixed column of* $\mathbf{S}$ *(i.e.,* $[b^{1,1}_i;\ldots;b^{1,\mathsf{u}}_i;b^{1,2}_i;\ldots;b^{\mathsf{N},\mathsf{u}}_i]$*). Hence, the full rank constraints in* (28) *cannot hold.*

As a result, if we have

$$\mathsf{Nu}\geq\mathsf{m}+\mathsf{u}-1+\mathsf{uv}=\mathsf{m}+\mathsf{u}-1+\mathsf{u}^2(\mathsf{N_r}-\mathsf{m}-\mathsf{u}+1)$$

which is equivalent to (33)*, it can be guaranteed that we can assign one linearly independent non-trivial solution to each Problem* t*.* □

For each $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{u}-1)<\mathsf{K_c}<\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}$ where $\mathsf{u}\in[\mathsf{N_r}-\mathsf{m}+1]$, we first generate $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}-\mathsf{K_c}$ demand vectors whose elements are uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$, and add these vectors into the demand matrix $\mathbf{F}$. Next, we use the above distributed computing scheme with $\mathsf{K_c'}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}$. Hence, the communication cost is $\frac{\mathsf{N_r}\mathsf{K_c'}}{\mathsf{m}+\mathsf{u}-1}=\frac{\mathsf{N_r}\mathsf{K}\mathsf{u}}{\mathsf{N}(\mathsf{m}+\mathsf{u}-1)}$, coinciding with (6a).

Remark 4. *For the proposed computing scheme for the case* $\big[\frac{\mathsf{K}}{\mathsf{N}}:\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1)\big]$*, the decoding complexity (i.e., the number of multiplications) of the master is* $\mathcal{O}\left(\mathsf{K_c}\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}\mathsf{N_r}\mathsf{L}\right)$*. Similarly, when* $\mathsf{K_c}\in\big[\frac{\mathsf{K}}{\mathsf{N}}\big]$*, the decoding complexity is* $\mathcal{O}\left(\mathsf{K_c}\mathsf{N_r}\mathsf{L}\right)$*. When* $\mathsf{K_c}\in\big[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N_r}-\mathsf{m}+1):\mathsf{K}\big]$*, the decoding complexity is* $\mathcal{O}\left(\mathsf{K_c}\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r^2}\mathsf{L}+\mathsf{K_c}\big(\frac{\mathsf{K_c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}-1}\big)\mathsf{L}\right)$*.* □

VI. Conclusions and Future Research Directions

In this paper, we studied the computation-communication costs tradeoff for the distributed linearly separable computation problem. A converse bound under the constraint of the cyclic assignment was proposed, and we also proposed a novel distributed computing scheme under some parameter regimes. Some exact optimality results were derived with or without the constraint of the cyclic assignment. The proposed computing scheme was also proved to be generally order optimal within a factor of 2 under the constraint of the cyclic assignment. The simplest open which the proposed scheme cannot work is the case where $\mathsf{K}=\mathsf{N}=\mathsf{N_r}=5$, $\mathsf{K_c}=2$, and $\mathsf{m}=2$. Further works include the design of the distributed computing scheme

for the open cases and the derivation of the converse bound for any dataset assignment.

Ongoing works include the generalization of the proposed scheme under any system parameters and the extension to the systems with partial stragglers as in [17], [28] or/and with partial computation recovery as in [29], [30].

APPENDIX A
FEASIBILITY PROOF OF THE PROPOSED COMPUTING SCHEME IN SECTION V

In the following, we first show that for the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{m}) = \big(\mathsf{N},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem, where $\mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}(\mathsf{N}_\mathrm{r}-\mathsf{m}-\mathsf{u}+1)$, the proposed computing scheme works with high probability. Next we show that if the proposed scheme works for the $\big(\mathsf{N},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem with high probability, then with high probability the proposed scheme also works for the $\big(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem, where $\frac{\mathsf{K}}{\mathsf{N}}$ is a positive integer.

A. $\mathsf{K}=\mathsf{N}$

The feasibility of the proposed computing scheme is proved by the Schwartz-Zippel Lemma [25]–[27] as we used in [5, Appendix C] for the computing scheme where $\mathsf{m}=1$. For the sake of simplicity, in the following we provide the sketch of the feasibility proof.

Recall that in Step 2 of the proposed computing scheme, for each pair (t,j) where $t\in[\mathsf{m}+\mathsf{u}-1]$ and $j\in[\mathsf{u}]$, we need to determine the values of the variables in (31) while satisfying the linear constraints in (30c). In addition, among all the variables in (31), we choose the values of the variables in (32) uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. Then there are remaining $\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})$ variables (the vector of these $\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})$ variables is assumed to be $\mathbf{b}$) and $\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})$ linear equations over these variables, and thus we can express these linear equations as (recall that $(\mathbf{M})_{m\times n}$ indicates that the dimension of matrix $\mathbf{M}$ is $m\times n$) $(\mathbf{A})_{\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})\times\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})}\ (\mathbf{b})_{\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})\times 1} = (\mathbf{c})_{\mathsf{K}(\mathsf{N}_\mathrm{r}-\mathsf{m})\times 1}$, where the coefficients in $\mathbf{A}$ and $\mathbf{c}$ are composed of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32) which are all generated uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. Hence, the determinant of $\mathbf{A}$ can be seen as a multivariate polynomial whose variables are the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32). Since the variables of the polynomial are uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$ where $\mathsf{q}\to\infty$, by the Schwartz-Zippel Lemma [25]–[27], if we can further show that this polynomial is a non-zero multivariate polynomial (i.e., a multivariate polynomial whose coefficients are not all 0), the probability that the polynomial is equal to 0 over all possible realization of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32), goes to 0. In other words, the determinant of $\mathbf{A}$ is non-zero with high probability. So the next step is to show this polynomial is non-zero. This means that we need to find one realization of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32), such that this polynomial is not equal to zero. By random generation of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32), we have tested all cases where $\mathsf{N}=\mathsf{K}\leq 40$ satisfying the constraint $\mathsf{N}\geq\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N}_\mathrm{r}-\mathsf{m}-\mathsf{u}+1)$. Hence, for each pair (t,j), the probability that Step 2 of the proposed computing scheme is feasible goes to 1. By the probability union bound, the probability that Step 2 of the proposed computing scheme is feasible for all pairs of (t,j), also goes to 1.

Moreover, by using the Cramer's rule, each element in $\mathbf{b}$ can be seen as a ratio of two polynomials whose variables are the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32), where the polynomial in the denominator is non-zero with high probability. As a result, each element in $\mathbf{S}$ can be seen as ratio of two polynomials of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32) for all pairs (t,j). So in Step 3 for each $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_\mathrm{r}$, the determinant of the matrix $[\mathbf{s}^{\mathcal{A}(1),1};\ldots;\mathbf{s}^{\mathcal{A}(1),\mathsf{K}_\mathrm{c}};\mathbf{s}^{\mathcal{A}(2),1};\ldots;\mathbf{s}^{\mathcal{A}(\mathsf{N}_\mathrm{r}),\mathsf{K}_\mathrm{c}}]$ can be expressed as $Y_\mathcal{A}=\sum_{i\in[(\mathsf{N}_\mathrm{r}\mathsf{u})!]}\frac{P_i}{Q_i}$, where P_i and Q_i are polynomial whose variables are the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32) for all pairs (t,j). We want to prove that $Y_\mathcal{A}\prod_{i\in[(\mathsf{N}_\mathrm{r}\mathsf{u})!]}Q_i$ is a non-zero polynomial such that we can use the Schwartz-Zippel Lemma [25]–[27] to show that the determinant $Y_\mathcal{A}$ is not equal to zero with high probability. Again, by random generation of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32) for all pairs (t,j), we have tested all cases where $\mathsf{N}=\mathsf{K}\leq 40$ satisfying the constraint $\mathsf{N}\geq\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N}_\mathrm{r}-\mathsf{m}-\mathsf{u}+1)$. In these cases, with the random choices, both $\prod_{i\in[(\mathsf{N}_\mathrm{r}\mathsf{u})!]}Q_i$ and $Y_\mathcal{A}$ are not equal to zero, and thus $Y_\mathcal{A}\prod_{i\in[(\mathsf{N}_\mathrm{r}\mathsf{u})!]}Q_i$ is not equal to 0.

In conclusion, we prove the feasibility of the proposed computing scheme in Steps 2 and 3 with high probability, for the case where $\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N}_\mathrm{r}-\mathsf{m}-\mathsf{u}+1)\leq\mathsf{K}=\mathsf{N}\leq 40$.

B. N *divides* K

We then consider the $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{m}) = \big(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem, where $\mathsf{N}\geq\frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}}+\mathsf{u}(\mathsf{N}_\mathrm{r}-\mathsf{m}-\mathsf{u}+1)$ and $\frac{\mathsf{K}}{\mathsf{N}}$ is a positive integer. Similar to the proof for the case where $\mathsf{K}=\mathsf{N}$, we also aim to find a specific realization of the elements in $\mathbf{F}$, $\mathbf{S}_{\mathsf{m}+\mathsf{u}}$ and (32) for all pairs (t,j), such that Steps 2 and 3 of the proposed scheme are feasible (i.e., the determinant polynomials are non-zero).

We construct the demand matrix (i.e., $\mathbf{F}$ with dimension $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}\times\mathsf{K}$) as follows,

$$\mathbf{F}=\begin{bmatrix}(\mathbf{F}_1)_{\mathsf{u}\times\mathsf{N}} & \mathbf{0}_{\mathsf{u}\times\mathsf{N}} & \cdots & \mathbf{0}_{\mathsf{u}\times\mathsf{N}}\\ \mathbf{0}_{\mathsf{u}\times\mathsf{N}} & (\mathbf{F}_2)_{\mathsf{u}\times\mathsf{N}} & \cdots & \mathbf{0}_{\mathsf{u}\times\mathsf{N}}\\ \vdots & \vdots & \vdots & \vdots\\ \mathbf{0}_{\mathsf{u}\times\mathsf{N}} & \mathbf{0}_{\mathsf{u}\times\mathsf{N}} & \cdots & (\mathbf{F}_{\mathsf{K}/\mathsf{N}})_{\mathsf{u}\times\mathsf{N}}\end{bmatrix},$$

where each element in $\mathbf{F}_i, i\in\left[\frac{\mathsf{K}}{\mathsf{N}}\right]$ is generated uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. In the above construction, the $\big(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem is divided into $\frac{\mathsf{K}}{\mathsf{N}}$ independent/disjoint $\big(\mathsf{N},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation sub-problems. Since the determinant polynomials are non-zero with high probability for each sub-problem as we proved in Appendix A-A, it can be seen that the determinant polynomials for the $\big(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\big)$ distributed linearly separable computation problem are also non-zero with high probability.

REFERENCES

[1] E. Amazon, "Amazon web services," *Available in: http://aws. amazon. com/es/ec2/(November 2012)*, 2015.

[2] K. S. P. T. and L. U. Gonzalez, *Building Your Next Big Thing with Google Cloud Platform: A Guide for Developers and Enterprise Architects*. Apress, 2015.

[3] B. Wilder, *Cloud architecture patterns: using microsoft azure*. " O'Reilly Media, Inc.", 2012.

[4] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, Mar. 2018.

[5] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *arXiv:2007.00345*, Jul. 2020.

[6] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," *in IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017.

[7] S. Wang, J. Liu, and N. Shroff, "Coded sparse matrix multiplication," *in Proc. 35th Intl. Conf. on Mach. Learning (ICML)*, pp. 5139–5147, 2018.

[8] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," *in Advances in Neural Information Processing Systems (NIPS)*, pp. 4406–4416, 2017.

[9] ——, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Infor. Theory*, vol. 66, no. 3, pp. 1920–1933, Mar. 2020.

[10] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Infor. Theory*, vol. 66, no. 1, pp. 278–301, Jan. 2020.

[11] M. Aliasgari, O. Simeone, and J. Kliewer, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Trans. Infor. Forensics and Security*, vol. 15, pp. 2722–2734, 2020.

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

[13] Z. Jia and S. A. Jafar, "Cross subspace alignment codes for coded distributed batch computation," *IEEE Trans. Infor. Theory*, Mar. 2021.

[14] ——, "On the capacity of secure distributed matrix multiplication," *arXiv:1908.06957*, Aug. 2019.

[15] Q. Yu and A. S. Avestimehr, "Entangled polynomial codes for secure, private, and batch distributed matrix multiplication: Breaking the cubic barrier," *arXiv:2001.05101*, Jan. 2020.

[16] A. Ramamoorthy, A. B. Das, and L. Tang, "Straggler-resistant distributed matrix computation via coding theory: Removing a bottleneck in large-scale data processing," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 136–145, May 2020.

[17] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," *in Advances in Neural Information Processing Systems (NIPS)*, p. 3368–3376, 2017.

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

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

[20] M. Ye and E. Abbe, "Communication computation efficient gradient coding," *in Advances in Neural Information Processing Systems (NIPS)*, pp. 5610–5619, 2018.

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

[22] 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, Oct. 2019.

[23] Y. Yang, M. Interlandi, P. Grover, S. Kar, S. Amizadeh, and M. Weimer, "Coded elastic computing," *in IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2654–2658, Jul. 2019.

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

[25] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM (JACM)*, vol. 27, no. 4, pp. 701–717, 1980.

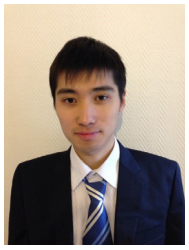
[26] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *International symposium on symbolic and algebraic manipulation*. Springer, 1979, pp. 216–226.

[27] R. A. Demillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Information Processing Letters*, vol. 7, no. 4, pp. 193–195, 1978.

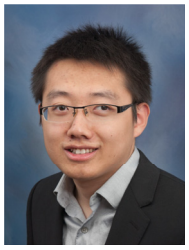
[28] A. B. Das, L. Tang, and A. Ramamoorthy, "C3les: Codes for coded computation that leverage stragglers," *in IEEE Inf. Theory Workshop (ITW 2018)*, Nov. 2018.

[29] E. Ozfatura, S. Ulukus, and D. Gunduz, "Coded distributed computing with partial recovery," *available at arXiv:2007.02191*, Jul. 2020.

[30] S. Sarmasarkar, V. Lalitha, and N. Karamchandani, "On gradient coding with partial recovery," *available at arXiv:2102.10163*, Feb. 2021.



Kai Wan (S'15 – M'18) received the B.E. degree in Optoelectronics from Huazhong University of Science and Technology, China, in 2012, the M.Sc. and Ph.D. degrees in Communications from Université Paris-Saclay, France, in 2014 and 2018. He is currently a post-doctoral researcher with the Communications and Information Theory Chair (CommIT) at 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 served as an Associate Editor of IEEE Communications Letters from Aug. 2021.



Hua Sun (S'12 – M'17) received the B.E. degree in Communications Engineering from 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 University of California Irvine, USA, in 2013 and 2017, respectively. He is an Assistant Professor in the Department of Electrical Engineering at the University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage. He is a recipient of the NSF CAREER award in 2021. His co-authored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016, and an IEEE GLOBECOM Best Paper Award in 2016.

Mingyue Ji (S'09-M'15) received the B.E. in Communication Engineering from Beijing University of Posts and Telecommunications (China), in 2006, the M.Sc. degrees in Electrical Engineering from Royal Institute of Technology (Sweden) and from University of California, Santa Cruz, in 2008 and 2010, respectively, and the PhD from the Ming Hsieh Department of Electrical Engineering at University of Southern California in 2015. He subsequently was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Limited) in 2015-2016. He is now an Assistant Professor of Electrical and Computer Engineering Department and an Adjunct Assistant Professor of School of Computing at the University of Utah. He received the IEEE Communications Society Leonard G. Abraham Prize for the best IEEE JSAC paper in 2019, the best paper award in IEEE ICC 2015 conference, the best student paper award in IEEE European Wireless 2010 Conference and USC Annenberg Fellowship from 2010 to 2014. He has served as an Associate Editor of IEEE Transactions on Communications from 2020. He is interested 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.

Giuseppe Caire (S'92 – M'94 – SM'03 – F'05) was born in Torino in 1965. He received the B.Sc. in Electrical Engineering from Politecnico di Torino in 1990, the M.Sc. in Electrical Engineering from Princeton University in 1992, and the Ph.D. from Politecnico di Torino in 1994. He has been a post-doctoral research fellow with the European Space Agency (ESTEC, Noordwijk, The Netherlands) in 1994-1995, Assistant Professor in Telecommunications at the Politecnico di Torino, Associate Professor at the University of Parma, Italy, Professor with the Department of Mobile Communications at the Eurecom Institute, Sophia-Antipolis, France, a Professor of Electrical Engineering with the Viterbi School of Engineering, University of Southern California, Los Angeles, and he is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science at the Technical University of Berlin, Germany.

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, an ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for best IEEE JSAC paper in 2019, the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020, and he is a recipient of the 2021 Leibinz Prize of the German National Science Foundation (DFG). Giuseppe Caire is a Fellow of IEEE since 2005. He has served in the Board of Governors of the IEEE Information Theory Society from 2004 to 2007, and as officer from 2008 to 2013. He was President of the IEEE Information Theory Society in 2011. His main research interests are in the field of communications theory, information theory, channel and source coding with particular focus on wireless communications.