Probabilistic Group Testing in Distributed Computing with Attacked Workers

Sarthak Jain, Martina Cardone, Soheil Mohajer
University of Minnesota, Minneapolis, MN 55455, USA, Email: {jain0122, mcardone, soheil}@umn.edu

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206705

***Abstract*—The problem of distributed matrix-vector product is considered, where the server distributes the task of the computation among n worker nodes, out of which L are compromised (but non-colluding) and may return incorrect results. Specifically, it is assumed that the compromised workers are unreliable, that is, at any given time, each compromised worker may return an incorrect and correct result with probabilities α and $1-\alpha$, respectively. Thus, the tests are noisy. This work proposes a new probabilistic group testing approach to identify the unreliable/compromised workers with $O\left(\frac{L\log(n)}{\alpha}\right)$ tests. Moreover, using the proposed group testing method, sparse parity-check codes are constructed and used in the considered distributed computing framework for encoding, decoding and identifying the unreliable workers. This methodology has two distinct features: (i) the cost of identifying the set of L unreliable workers at the server can be shown to be considerably lower than existing distributed computing methods, and (ii) the encoding and decoding functions are easily implementable and computationally efficient.**

I. Introduction

In distributed computing, an expensive task can be *encoded* into sub-tasks of lower complexity, which can then be distributed to several worker nodes, in order to improve the overall computational speed [1]–[7]. However, a subset of the workers may be compromised and may return incorrect results of the sub-tasks assigned to them. In other words, these compromised workers are unreliable, i.e., they sometimes return incorrect results (i.e., they behave maliciously) and at other times behave like reliable workers and return correct results, making them difficult to identify. In such scenarios, it is desirable that the encoding into sub-tasks is performed in a way that two properties hold: (i) *Identification*: there is an efficient mechanism to identify the set of unreliable/compromised worker nodes; and (ii) *Decodability*: from the results of the sub-tasks assigned to the reliable worker nodes, the result of the original task can be recovered efficiently.

In this work, we propose a probabilistic group testing based distributed computing scheme for the task of matrix-vector computation, which ensures both efficient identification and decodability. Group testing was first introduced in [8] and extensively studied in areas ranging from medicine [9] to computer science [1], to efficiently identify a set of defective items in a large set of n items, by testing groups of items at a time rather than testing the items individually. Two types of group testing methodologies are prevalent: (i) *probabilistic group testing*; and (ii) *combinatorial group testing*. Combinatorial group testing is deterministic and identifies defective items with a zero probability of error [10]. In probabilistic group testing, this probability of error goes to 0 as $n \to \infty$, whereas for finite n, it can be made arbitrarily small by increasing the number of tests [11]–[14]. Unlike traditional group testing, in this work, the goal is to identify the set of L (out of n) unreliable workers, where the unreliable workers do not always behave maliciously, but they sometimes hide their true identity. A test can therefore be negative even if there were one or more unreliable workers in the tested group. Group testing with unreliable items has been widely studied [15]–[19]. However, the probabilistic model for the behavior of unreliable workers that we adopt is different from the models considered in the above works. This difference is primarily due to the fact that the behavior of an unreliable worker (i.e., either acting maliciously or acting reliably) remains the same for all the tests related to a specific computing task. Thus, naively adopting the probabilistic group testing methods of [15]–[18] for our model, produce sub-optimal results.

The contribution of this work is threefold: (i) we propose a probabilistic group testing scheme to efficiently identify the L unreliable workers (Section II); (ii) using the proposed group testing scheme, we construct sparse parity-check codes which are used for encoding and enable an efficient identification of the unreliable workers (Section III); and (iii) we construct a low-complexity decoding function for retrieving the original matrix-vector product from the results of the reliable workers (Section III). The proposed scheme can be shown to outperform the MDS-code based schemes presented in [1], [2] for unreliable worker's identification.

Notation. For an integer $n \geq 1$, we define $[n] \triangleq \{1, 2, \ldots, n\}$. For a matrix M, we use $\mathsf{M}_{i,:}$ and $\mathsf{M}_{:,j}$ to represent its ith row and jth column, respectively. Moreover, for sets $\mathcal{A}$ and $\mathcal{B}$, $\mathsf{M}_{\mathcal{A},\mathcal{B}}$ is the submatrix of M where only the rows in $\mathcal{A}$ and the columns in $\mathcal{B}$ are retained. For $x, y \in \{0, 1\}$, we define $x \succ y$ if $(x, y) = (1, 0)$ and $x \preceq y$, otherwise. For a vector $\boldsymbol{z}$, we define $\mathsf{supp}(\boldsymbol{z}) = \{j : \boldsymbol{z}_j \neq 0\}$.

II. System Model

The server node aims at computing T matrix-vector products $\mathbf{B} \cdot \mathbf{v}_t$ for $t \in [T]$, where $\mathbf{B} \in \mathbb{F}^{r\times c}$ and $\mathbf{v}_t \in \mathbb{F}^{c\times 1}$. In other words, we aim at computing the tth matrix-vector product $\mathbf{B} \cdot \mathbf{v}_t$ in time-slot $t \in [T]$. To speed up the computation, the server can distribute the task among n workers, denoted by the set $[n]$. However, a set $\mathcal{L} \subseteq [n]$ of $|\mathcal{L}| = L$ workers are unreliable and may return incorrect/noisy results for the task

This research was supported in part by the U.S. National Science Foundation under Grants CCF-1907785 and CCF-2045237.

assigned to them. We model the random behavior of these L unreliable workers as follows: at time-slot $t \in [T]$, each unreliable worker is *attacked* with probability α, and returns an incorrect result, independent of other workers and other time-slots. If an unreliable worker remains unattacked in a time-slot t, it behaves reliably and returns the correct result. We denote by $\mathcal{L}_t \subseteq \mathcal{L}$ the subset of unreliable workers, which are attacked in time-slot t. For any $w \in \mathcal{L}$, we have

$$\mathbb{P}\big(w \in \mathcal{L}_t \,\big|\, w \in \mathcal{L}\big) = \alpha. \tag{1}$$

For the distributed computing scheme, the vectors $\mathbf{v}_t$'s are first communicated to all the n workers. Moreover, each worker $w \in [n]$ receives a matrix $\mathbf{W}^{(w)} \in \mathbb{F}^{s\times c}$ with $s = r/k \ll r$ rows (assuming that $k \in \mathbb{N}$ divides r) obtained by using a set of encoding functions $f^{(w)} : \mathbb{F}^{r\times c} \to \mathbb{F}^{s\times c}$ as

$$\mathbf{W}^{(w)} = f^{(w)}(\mathbf{B}), \qquad w \in [n]. \tag{2}$$

For each $t \in [T]$, the wth worker then computes the matrix-vector product $\mathbf{a}_t^{(w)} = \mathbf{W}^{(w)} \cdot \mathbf{v}_t$, and sends a message back to the server. In particular, the result $\tilde{\mathbf{a}}_t^{(w)}$ returned by the wth worker is given by

$$\tilde{\mathbf{a}}_t^{(w)} = \begin{cases} \mathbf{W}^{(w)} \cdot \mathbf{v}_t & \text{if } w \notin \mathcal{L}_t, \\ \mathbf{W}^{(w)} \cdot \mathbf{v}_t + \mathbf{z}_t^{(w)} & \text{if } w \in \mathcal{L}_t, \end{cases} \tag{3}$$

where $\mathbf{z}_t^{(w)} \in \mathbb{F}^{s\times 1}$ is the noise vector corresponding to worker w and is independent of the noise vectors of other attacked workers (since we assume the unreliable workers to be non-colluding). The server is then required to identify the unreliable workers and subsequently reconstruct/decode the correct product $\mathbf{B} \cdot \mathbf{v}_t$ by applying a decoding function $g(\cdot)$ on the results received from (a subset of) the reliable workers.

We propose a distributed computing scheme where the encoding functions $f^{(i)}$ in (2) and the decoding function g are such that the overall computational complexity at the server is low because: (i) the set of unreliable workers can be identified with a low computational cost of $O\Big(\frac{L\log(n)}{\alpha}\Big) \times O\big(\frac{rn}{Lk}\big)$; and (ii) the decoding can be performed with a cost of $O\big(\frac{rn}{k}\big) \times T$ operations for T matrix-vector products.

III. Group Testing for Identifying Unreliable Workers

We are interested in identifying the set of all unreliable workers $\mathcal{L}$. Here, we face two main challenges: (1) workers cannot be tested individually, and (2) an unreliable worker is not always attacked and hence, we cannot identify it from its result in a single time-slot.

Instead, we can test a group of workers, and a test result will be negative if all the unreliable workers in the selected group are unattacked during the time-slot of interest. In other words, a test performed based on the results of time-slot t will be positive if and only if there is at least one attacked unreliable worker $w \in \mathcal{L}_t$ in the tested group.

We let m_t be the number of tests that are performed in time-slot $t \in [T]$. Then the total number of tests is $M := \sum_{t=1}^{T} m_t$. These tests can be represented using a *contact* matrix $\mathsf{M}^{(c)} \in \{0,1\}^{M\times n}$, where $\mathsf{M}^{(c)}_{j,w} = 1$ if and only if the wth worker is included in the jth test. The tests conducted in time-slot t correspond to the rows $\mathcal{T}_t = [1 + \sum_{j=1}^{t-1} m_t : \sum_{j=1}^{t} m_t]$ of $\mathsf{M}^{(c)}$. We can use a vector $\boldsymbol{x} \in \{0,1\}^{n\times 1}$ to indicate whether or not each worker is unreliable, that is, $\boldsymbol{x}_w = 1$ if and only if $w \in \mathcal{L}$. Ideally, if all the unreliable workers were always attacked, i.e., $\alpha = 1$, the result of the tests can be represented by a vector $\boldsymbol{y}^{(c)} \in \{0,1\}^{M}$ as

$$\boldsymbol{y}^{(c)} = \mathsf{M}^{(c)} \odot \boldsymbol{x}, \tag{4}$$

where the multiplication and addition are logical `and` and `or`, respectively. More precisely, we have $\boldsymbol{y}_i^{(c)} = \bigvee_{j=1}^{n} (\mathsf{M}_{i,j}^{(c)} \wedge \boldsymbol{x}_j)$.

To capture the unreliable behavior of the compromised workers, we adopt the notation in [17], and define a *sampling* matrix $\mathsf{M}^{(s)}$ which is obtained from the contact matrix $\mathsf{M}^{(c)}$ as

$$\mathsf{M}^{(s)}_{\mathcal{T}_t,w} = \begin{cases} \mathsf{M}^{(c)}_{\mathcal{T}_t,w} & \text{if } w \in ([n] \setminus \mathcal{L}) \cup \mathcal{L}_t, \\ \mathbf{0} & \text{if } w \in \mathcal{L} \setminus \mathcal{L}_t. \end{cases} \tag{5}$$

Moreover, the result of the actual tests in the presence of unreliable workers is given by

$$\boldsymbol{y} = \mathsf{M}^{(s)} \odot \boldsymbol{x}. \tag{6}$$

To understand how the sampling matrix models the unreliable behavior, let us assume that worker w was selected to be included in a test $j \in \mathcal{T}_t$ in time-slot t. Then, we have $\mathsf{M}^{(c)}_{j,w} = 1$. If $w \in \mathcal{L}_t$, then in time-slot t the unreliable worker is attacked and exposes its true identity and therefore, we should have $\boldsymbol{y}_j^{(c)} = \boldsymbol{y}_j = 1$, which is achieved by setting $\mathsf{M}^{(s)}_{j,w} = \mathsf{M}^{(c)}_{j,w}$ in such a scenario. Now, assume that worker w is unreliable (i.e., $\boldsymbol{x}_w = 1$), but unattacked in time-slot t, that is, $w \in \mathcal{L} \setminus \mathcal{L}_t$. In this case, w hides its true identity. Since $\mathsf{M}^{(c)}_{j,w} = 1$, we have $\boldsymbol{y}_j^{(c)} = 1$. But the actual test result $\boldsymbol{y}_j$ should not be influenced by w. This unattacked behavior of worker w in test $j \in \mathcal{T}_t$ is captured by setting $\mathsf{M}^{(s)}_{j,w} = 0$.

Note that if $\alpha = 1$, then $\mathcal{L}_t = \mathcal{L}$ for all $t \in [T]$, and therefore $\mathsf{M}^{(c)}_{j,w} = \mathsf{M}^{(s)}_{j,w}$ for all $j \in [M]$ and $w \in [n]$, which reduces the problem to the classical group testing problem [8].

Examples of a contact matrix $\mathsf{M}^{(c)}$ and a sampling matrix $\mathsf{M}^{(s)}$ for $n = 5$ workers, $T = 2$, $m_1 = 2$, and $m_2 = 1$ is given in (7). Here, $\mathcal{L} = \{3, 4\}$, and $\mathcal{L}_1 = \{4\}$. Since worker 3 is unattacked at $t = 1$, the entries of $\mathsf{M}^{(c)}$ at column 3 and $\mathcal{T}_1 = [1:2]$ are replaced by 0 in $\mathsf{M}^{(s)}$:

$$\mathsf{M}^{(c)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}, \ \mathsf{M}^{(s)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

Remark 1. *Note that the tests are always conducted according to the contact matrix $\mathsf{M}^{(c)}$, but due to the random behavior of the unreliable workers, the actual result of the tests is determined by $\boldsymbol{y} = \mathsf{M}^{(s)} \odot \boldsymbol{x}$ (rather than $\boldsymbol{y}^{(c)} = \mathsf{M}^{(c)} \odot \boldsymbol{x}$).*

We seek to minimize the number of tests M, i.e., design a contact matrix $\mathsf{M}^{(c)}$ with as few rows as possible, such that, given $\mathsf{M}^{(c)}$ and $\boldsymbol{y}$, we can identify the unreliable set $\mathcal{L}$ (or equivalently the vector $\boldsymbol{x}$), with an arbitrarily small probability

of error (as $n \to \infty$). In particular, we are interested in characterizing how the number of tests M should scale with respect to the underlying parameters, n, L and α, to achieve a vanishing error probability.

To construct $\mathsf{M}^{(c)}$, we consider parameters (Z, m), where we assume $m_t = m$ tests are performed only for $t \in [Z]$ (i.e., $m_t = 0$ for $t > Z$). Therefore, the total number of tests performed is $M = mZ$. The parameters (m, Z) will be determined later to guarantee the identification of the unreliable workers with high probability.

Probabilistic construction for $\mathsf{M}^{(c)}$ and $\mathsf{M}^{(s)}$: *We generate $\mathsf{M}^{(c)} \in \mathbb{F}^{Zm\times n}$ randomly, where each entry is drawn from a Bernoulli distribution with parameter $q := \frac{\theta}{L}$ (where the parameter θ will be determined later), independent of all other entries. For this scheme, the corresponding sampling matrix $\mathsf{M}^{(s)}$ is generated from $\mathsf{M}^{(c)}$ as follows,*

$$\mathsf{M}^{(s)}_{\mathcal{T}_z,w} = \begin{cases} \mathsf{M}^{(c)}_{\mathcal{T}_z,w} & \text{if } w \in ([n] \setminus \mathcal{L}) \cup \mathcal{L}_z, \\ \mathbf{0} & \text{if } w \in \mathcal{L} \setminus \mathcal{L}_z, \end{cases} \tag{8}$$

where, $\mathcal{T}_z = [(z-1)m + 1 : zm]$ is the set of rows which correspond to time-slot $z \in Z$.

The (d, ϵ)-threshold decoder: *For $z \in [Z]$, we let $\boldsymbol{y}_{\mathcal{T}_z} \in \{0, 1\}^m$ be the test result vector of time-slot z, which is given by $\boldsymbol{y}_{\mathcal{T}_z} = \mathsf{M}^{(s)}_{\mathcal{T}_z,:} \odot \boldsymbol{x}$, where $\mathcal{T}_z = [(z-1)m+1 : zm]$ is the set of tests in time-slot z. Moreover, we define the score of a worker $w \in [n]$ in time-slot z by $I_{w,z}$, which is given by*

$$I_{w,z} = \begin{cases} 1 & \text{if } \mathsf{M}^{(c)}_{\mathcal{T}_z,w} \preceq \boldsymbol{y}_{\mathcal{T}_z} \text{ and } \mathsf{M}^{(c)}_{\mathcal{T}_z,w} \neq \mathbf{0}, \\ \epsilon & \text{if } \mathsf{M}^{(c)}_{\mathcal{T}_z,w} = \mathbf{0}, \\ 0 & \text{if } \mathsf{M}^{(c)}_{\mathcal{T}_z,w} \not\preceq \boldsymbol{y}_{\mathcal{T}_z}. \end{cases} \tag{9}$$

Then, a worker w will be marked as unreliable (i.e., $\hat{\boldsymbol{x}}_w = 1$) if and only if $I_w := \sum_{z=1}^{Z} I_{w,z} \geq d$. Note that an unreliable worker's score is expected to be higher than a reliable worker's score. We leverage this fact to obtain the following result that guarantees the success of the proposed decoder (with high probability) for a proper choice of parameters.

Theorem 1. *For any arbitrary, but a-priori fixed, L-sparse vector $\boldsymbol{x}$ and any error exponent $\beta > 0$, there exists a choice of parameters (q, m, Z, d, ϵ) such that the (d, ϵ)-threshold decoder can decode $\boldsymbol{x}$ with error probability $P_e = \mathbb{P}[\hat{\boldsymbol{X}} \neq \boldsymbol{x} | \boldsymbol{X} = \boldsymbol{x}] \leq n^{-\beta}$ with at most $450(1+\beta)\frac{L\log(n)}{\alpha}$ tests.*

Remark 2. *In this work, we focus on noise-level-independent Bernoulli designs of the contact matrix $\mathsf{M}^{(c)}$ with parameter $q = \frac{\theta}{L}$ (where q does not depend on the noise parameter α). Advantages of such a design include: (1) increased robustness against erroneous estimates of α, and (2) no requirement of redesigning $\mathsf{M}^{(c)}$ every time that α fluctuates. Under such a construction of $\mathsf{M}^{(c)}$, it is not difficult to show an information theoretic converse of $O\Big(\frac{L\log(n)}{\alpha\log(1/\alpha)}\Big)$ tests for the group testing noise model considered in this work. The (d, ϵ)-threshold decoder almost achieves this converse, except for a small factor of $\log(1/\alpha)$. Moreover, noise-level-dependent designs could lead to better achievable bounds, as it has been observed in other noise models [20], [21] and we intend to explore such designs in the extended version of this paper.*

In the proof of Theorem 1, we will show that under a proper choice of (q, m, Z, d, ϵ), the error probability vanishes as $n^{-\beta}$. In particular, we set $(q, m, Z, \epsilon) = \Big(\frac{\theta}{L}, \frac{L}{\theta}, \frac{\lambda\log n}{\alpha}, \theta\alpha\Big)$, where λ and $\theta \in [0, 1]$ are design parameters, to be determined later. Moreover, the value of d is given in (14).

We will need the following two propositions, which play an essential role in the proof. The proofs of the propositions are presented in [22, Appendix A] and [22, Appendix B].

Proposition 1. *The total expected scores for reliable and unreliable workers are given by,*

$$\mu_f := \mathbb{E}\big[I_w \,\big|\, w \notin \mathcal{L}\big] = Z(h_L - (1-\epsilon)(1-q)^m), \tag{10}$$
$$\mu_m := \mathbb{E}\big[I_w \,\big|\, w \in \mathcal{L}\big] = Z(\alpha + (1-\alpha)h_{L-1} - (1-\epsilon)(1-q)^m),$$

where for every integer $x \in [0 : L]$, we have that

$$h_x := \sum\nolimits_{\ell=0}^{x} \binom{x}{\ell} \alpha^\ell (1-\alpha)^{x-\ell} \big(1 - q(1-q)^\ell\big)^m. \tag{11}$$

Proposition 2. *Assume $qm \leq 1$. Then, h_x in (11) satisfies*

$$(1-q)^m \leq h_L \leq (1-q)^m + mLq^2\alpha, \tag{12}$$

and, $$h_L - (1-\alpha)h_{L-1} \leq \alpha(1 - mq(1-qL)/2). \tag{13}$$

Proof of Theorem 1. There are two types of error associated with the (d, ϵ)-threshold decoder: (i) a *false alarm*, where a worker $w \in [n] \setminus \mathcal{L}$ is identified as unreliable, and (ii) a *mis-detection* error, where an unreliable worker $w \in \mathcal{L}$ is identified as reliable. Correspondingly, we define $\mathsf{P}_+(w) := \mathbb{P}(I_w \geq d | w \notin \mathcal{L})$ and $\mathsf{P}_-(w) = \mathbb{P}(I_w < d | w \in \mathcal{L})$, where d is the decoder parameter which is set to

$$d := (1+\eta)\mu_f = (1+\eta)Z(h_L - (1-\epsilon)(1-q)^m), \tag{14}$$

where $\eta > 0$ will be determined later. Since there are L unreliable and $n - L$ reliable workers, using the union bound, the total error probability is upper bounded by

$$\mathsf{P}_\mathrm{e} \leq \sum_{w\in[n]\setminus\mathcal{L}} \mathsf{P}_+(w) + \sum_{w\in\mathcal{L}} \mathsf{P}_-(w). \tag{15}$$

In the following, we will bound both probabilities of error for the regime of parameters of interest. We start with

$$\begin{aligned} \mathsf{P}_+(w) &= \mathbb{P}\big(I_w \geq d \;\big|\; w \notin \mathcal{L}\big) = \mathbb{P}\big(I_w \geq (1+\eta)\mu_f \;\big|\; w \notin \mathcal{L}\big) \\ &\overset{(a)}{\leq} \exp\left(-\frac{\eta^2}{2+\eta}\mu_f\right) = \exp\left(-\frac{\eta^2}{2+\eta}Z(h_L - (1-\epsilon)(1-q)^m)\right) \\ &\overset{(b)}{\leq} \exp\left(-\frac{\eta^2}{2+\eta}Z\epsilon(1-q)^m\right) \\ &\overset{(c)}{\leq} \exp\left(-\frac{\eta^2}{2+\eta}\theta\exp\left(-\frac{qm}{1-q}\right)Z\alpha\right) \overset{(d)}{=} \exp(-\zeta_1 Z\alpha), \end{aligned} \tag{16}$$

where: (a) is due to the Chernoff bound; in (b) we used Proposition 2; (c) follows from the facts that $\epsilon = \theta\alpha$, and

$1-x \geq \exp\Big(-\frac{x}{1-x}\Big)$ which holds for $x<1$; and in (d), we used the choice of parameters $(q,m)=\big(\frac{\theta}{L},\frac{L}{\theta}\big)$ which yields $1-q \geq 1-\theta$ and

$$\zeta_1 = \frac{\eta^2}{2+\eta}\theta\exp\bigg(-\frac{1}{1-q}\bigg) \geq \frac{\eta^2}{2+\eta}\theta\exp\bigg(-\frac{1}{1-\theta}\bigg).$$

Similarly, for $\mathsf{P}_-(w)$ we can write

$$\begin{aligned}\mathsf{P}_-(w) &= \mathbb{P}(I_w < d | w\in\mathcal{L}) = \mathbb{P}(I_w<(1-\delta)\mu_m | w\in\mathcal{L})\\ &\leq \exp\bigg(-\frac{1}{2}\delta^2\mu_m\bigg),\end{aligned} \tag{17}$$

where $\delta := \frac{\mu_m - d}{\mu_m}$. We note that

$$\begin{aligned}&\mu_m - d = \mu_m - (1+\eta)\mu_f\\ &= Z[\alpha - \eta(h_L - (1-\epsilon)(1-q)^m) - (h_L - (1-\alpha)h_{L-1})]\\ &\overset{(e)}{\geq} Z\bigg[\alpha - \eta(\epsilon(1-q)^m + mLq^2\alpha) - \alpha\Big(1-\frac{mq(1-qL)}{2}\Big)\bigg]\\ &\overset{(f)}{=} Z\alpha[-\eta\theta(1-q)^m - \eta mLq^2 + mq/2 - mLq^2/2]\\ &= \zeta_2 Z\alpha,\end{aligned} \tag{18}$$

where (e) follows from Proposition 2, and in (f) we plugged in $\epsilon=\theta\alpha$. Note that

$$\begin{aligned}\zeta_2 &= \frac{mq}{2} - \eta\theta(1-q)^m - \Big(\eta+\frac{1}{2}\Big)mLq^2\\ &\overset{(g)}{\geq} \frac{1}{2} - \theta\Big(\eta\exp(-1) + \Big(\eta+\frac{1}{2}\Big)\Big),\end{aligned}$$

where in (g) we used the facts that $(q,m)=\big(\frac{\theta}{L},\frac{L}{\theta}\big)$ and $(1-q)^m \leq \exp(-mq) = \exp(-1)$.

Moreover, we have that

$$\begin{aligned}\mu_m &= Z[\alpha + (1-\alpha)h_{L-1} - (1-\epsilon)(1-q)^m]\\ &\overset{(h)}{\leq} Z[\alpha + \epsilon(1-q)^m + mLq^2\alpha] = \zeta_3 Z\alpha,\end{aligned} \tag{19}$$

where the inequality in (h) follows from the chain of inequalities $(1-\alpha)h_{L-1} \leq h_{L-1} \leq h_L \leq (1-q)^m + mLq^2\alpha$, which is implied by Proposition 2. Note that since $\epsilon=\theta\alpha$, we have

$$\zeta_3 = 1 + \theta(1-q)^m + mLq^2 \overset{(i)}{\leq} 1 + \theta\exp(-1) + \theta,$$

where (i) follows since $(q,m)=\big(\frac{\theta}{L},\frac{L}{\theta}\big)$ and from the fact that $(1-q)^m \leq \exp(-mq) = \exp(-1)$. Therefore, plugging (18) and (19) into (17), we get

$$\begin{aligned}\mathsf{P}_-(w) &\leq \exp\bigg(-\frac{1}{2}\frac{(\mu_m-d)^2}{\mu_m}\bigg)\\ &\leq \exp\bigg(-\frac{1}{2}\frac{\zeta_2^2 Z^2\alpha^2}{\zeta_3 Z\alpha}\bigg) = \exp(-\zeta_4 Z\alpha),\end{aligned} \tag{20}$$

where $\zeta_4 = \zeta_2^2/2\zeta_3$. Now, setting $(\theta,\eta)=(0.15,1)$, we have $\zeta_1,\zeta_4 > \zeta := 0.015$. Plugging (16) and (20) in (15), we get

$$\begin{aligned}\mathsf{P}_{\mathrm{e}} &\leq (n-L)\exp(-\zeta_1 Z\alpha) + L\exp(-\zeta_4 Z\alpha)\\ &\overset{(j)}{\leq} n\exp\bigg(-\zeta\frac{\lambda\log n}{\alpha}\alpha\bigg) = \exp(-(\zeta\lambda-1)\log n) \overset{(k)}{=} n^{-\beta},\end{aligned}$$

where (j) holds for $\zeta_1,\zeta_4>\zeta$ and $Z=\frac{\lambda\log n}{\alpha}$, and (k) follows from the fact that $\lambda=(1+\beta)/\zeta$. Note that the total number of tests is $mZ = \frac{L}{\theta}\frac{\lambda\log n}{\alpha} < 450(1+\beta)\frac{L\log n}{\alpha}$. This completes the proof of Theorem 1. □

IV. Distributed Scheme

In this section, we present the distributed computing scheme by dividing the scheme into the following three subsections.

A. Generator Matrix for Encoding

To obtain the generator matrix for encoding, we start with the group testing contact matrix $\mathsf{M}^{(c)} \in \{0,1\}^{M\times n}$ that we obtained in Section III, where $M=mZ$. We design a random parity matrix $\mathsf{M}^{(p)} \in \mathbb{F}^{M\times n}$, given by $\mathsf{M}^{(p)}_{i,j} = \mathsf{M}^{(c)}_{i,j}\cdot\mathsf{Q}_{i,j}$, where the entries of $\mathsf{Q}\in\mathbb{F}^{M\times n}$ are chosen uniformly and independently at random from $\mathbb{F}\setminus\{0\}$. We will use the linear code induced by $\mathsf{M}^{(p)}$ for pre-coding of the matrix $\mathbf{B}$. To this end, let $\mathsf{G}\in\mathbb{F}^{k\times n}$ be the generator matrix of the *systematic* code induced by the parity-check matrix $\mathsf{M}^{(p)}$, that is,

$$\mathsf{G} = \big[\mathbf{I}_{k\times k}\big|\mathbf{R}_{k\times(n-k)}\big],$$

for some matrix $\mathbf{R}\in\mathbb{F}^{k\times(n-k)}$ where[1] $k=n-M$ and G satisfies $\mathsf{M}^{(p)}\cdot\mathsf{G}^T = \mathbf{0}$.

Next, we use the matrix G for encoding the original matrix $\mathbf{B}\in\mathbb{F}^{r\times c}$. To this end, $\mathbf{B}$ is first divided horizontally into k equal parts $\mathbf{B}_1,\mathbf{B}_2,\cdots,\mathbf{B}_k$. Then, for $w\in[n]$ we generate the matrix $\mathbf{W}^{(w)}\in\mathbb{F}^{s\times c}$ (where $s=r/k$), given by

$$\mathbf{W}^{(w)} = \sum\nolimits_{j=1}^{k}\mathsf{G}_{j,w}\mathbf{B}_j, \tag{21}$$

and send it to worker w, who is responsible for computing the product $\mathbf{a}_t^{(w)} = \mathbf{W}^{(w)}\cdot\mathbf{v}_t$.

B. Group Testing for Identifying the Unreliable Workers

Worker w will return the result $\tilde{\mathbf{a}}_t^{(w)}$ in (3), which may or may not be equal to the expected result $\mathbf{a}_t^{(w)}$, depending on whether $w\in\mathcal{L}_t$ or not (i.e., worker w is attacked in time-slot t or not; see (3)). The server can not determine if $\tilde{\mathbf{a}}_t^{(w)}$ is correct or incorrect only from the information provided by worker w. However, the set of correct results $\{\mathbf{a}_t^{(w)} : w\in[n]\}$ satisfy M parity equations, corresponding to the M rows of the parity-check matrix $\mathsf{M}^{(p)}$. To formalize this, we define a parity check function as follows.

Parity Check Function: Consider some $i\in[M]$ and a set of workers $\mathcal{U}_i = \mathsf{supp}\Big(\mathsf{M}^{(p)}_{i,:}\Big)$ with the corresponding results $\{\tilde{\mathbf{a}}_t^{(j)} : j\in\mathcal{U}_i\}$. We define the parity check function $\Gamma_t(\mathcal{U}_i)$ as

$$\Gamma_t(\mathcal{U}_i) := \sum_{j\in\mathcal{U}_i}\mathsf{M}^{(p)}_{i,j}\tilde{\mathbf{a}}_t^{(j)}, \tag{22}$$

which can be used to check if there is any attacked (and hence unreliable) worker in the set $\mathcal{U}_i$ that returned an incorrect result in time-slot $t\in[T]$. The following lemma, the proof of which is in [22, Appendix C], states important properties of $\Gamma_t(\cdot)$.

[1]We note that in general it is rather unlikely, but possible, that $\mathsf{M}^{(p)}$ is not full-rank. In spite of that, we can always find $k=n-M$ linearly independent vectors in $\mathbb{F}^n$ which are orthogonal to the rows of $\mathsf{M}^{(c)}$.

Lemma 1. *In the finite field $\mathbb{F}$, for every $i \in [M]$ and set $\mathcal{U}_i = \mathsf{supp}(\mathsf{M}^{(p)}_{i,:})$, we have*

$$\begin{aligned}\mathbb{P}[\Gamma_t(\mathcal{U}_i) = \mathbf{0} | \mathcal{U}_i \cap \mathcal{L}_t = \varnothing] = 1,\\ \mathbb{P}[\Gamma_t(\mathcal{U}_i) = \mathbf{0} | \mathcal{U}_i \cap \mathcal{L}_t \neq \varnothing] = \frac{1}{|\mathbb{F}|}.\end{aligned} \tag{23}$$

We now discuss the use of the parity check function to identify the set of unreliable workers.

Probabilistic Group Testing: In time-slot $z \in [Z]$, the server computes $\hat{\boldsymbol{y}}_j$ for $j \in \mathcal{T}_z = [(m-1)z+1 : mz]$, (where the parameters Z and m are specified in Section III) as follows,

$$\hat{\boldsymbol{y}}_j = \begin{cases} 0 & \text{if } \Gamma_z(\mathcal{U}_j) = \mathbf{0}, \\ 1 & \text{otherwise}, \end{cases} \tag{24}$$

where $\mathcal{U}_j = \mathsf{supp}\Big(\mathsf{M}^{(p)}_{j,:}\Big) = \mathsf{supp}\Big(\mathsf{M}^{(c)}_{j,:}\Big)$, since $\mathsf{M}^{(c)}_{j,w} = 0$ if and only if $\mathsf{M}^{(p)}_{j,w} = 0$. Hence, the parity equation on $\mathcal{U}_j$ is equivalent to a group test on $\mathcal{U}_j$. However, due to (23), we get

$$\mathbb{P}(\hat{\boldsymbol{y}} = \boldsymbol{y}) = \mathbb{P}\Big(\hat{\boldsymbol{y}} = \mathsf{M}^{(s)} \odot \boldsymbol{x}\Big) \geq (1 - 1/|\mathbb{F}|)^M,$$

or equivalently, $\mathbb{P}(\hat{\boldsymbol{y}} \neq \boldsymbol{y}) \leq 1 - (1-1/|\mathbb{F}|)^M \leq \frac{M}{|\mathbb{F}|}$. Therefore, the server can use the (d, ϵ)-threshold decoder to identify all the unreliable workers $\mathcal{L}$ with an error probability less than $n^{-\beta} + \frac{M}{|\mathbb{F}|}$, where the $n^{-\beta}$ and $\frac{M}{|\mathbb{F}|}$ represent the upper bounds on the errors due to probabilistic group testing and the probabilistic nature of the parity-check function, respectively.

Remark 3. *Here, we need to choose a finite field with $|\mathbb{F}| \gg M$ to guarantee the success of the algorithm with high probability. However, using a more sophisticated analysis for group testing, we can consider the underlying model similar to [23], in which the results of the tests are passed through a Z-channel, i.e., $\mathbb{P}(\hat{\boldsymbol{y}}_j = 0 | \boldsymbol{y}_j = 1) = 1/|\mathbb{F}|$.*

Average Computational Cost of Identifying Unreliable Workers: On average, each row of $\mathsf{M}^{(p)}$ contains $qn = \frac{\theta n}{L}$ non-zero elements and $\tilde{\mathbf{a}}^{(w)}_t$ has $s = \frac{r}{k}$ elements. Therefore, the computational cost of $\Gamma_t(\mathcal{U}_j)$ for $j \in [M]$ is $O\big(\frac{\theta nr}{kL}\big) = O\big(\frac{nr}{kL}\big)$. Moreover, a total of $M = O\Big(\frac{L\log(n)}{\alpha}\Big)$ such tests are performed. Therefore, the total cost of identifying the unreliable workers is $O\Big(\frac{rn\log(n)}{k\alpha}\Big)$. Note that this does not scale with T, the number of vectors to be multiplied by $\mathbf{B}$. Moreover, for the interesting regime where $\frac{L\log n}{\alpha} = o(n)$, we have $M = o(n)$ and hence $k = n - M = \Theta(n)$. Consequently, in this regime, the total cost of identifying the unreliable workers will be $O\Big(\frac{r\log(n)}{\alpha}\Big)$.

C. Reconstruction of Incorrect Results and Decoding

Note that after performing group testing, the server knows the set of unreliable workers $\mathcal{L}$ with a probability of error less than $n^{-\beta} + \frac{M}{|\mathbb{F}|}$. Since the generator matrix is of the form $\mathsf{G} = \big[\mathbf{I}_{k\times k} \big| \mathbf{R}_{k\times(n-k)}\big]$, the first k workers received matrices $\{\mathbf{B}_1, \mathbf{B}_2, \cdots, \mathbf{B}_k\}$ and were expected to compute $\{\mathbf{a}^{(1)}_t = \mathbf{B}_1 \cdot \mathbf{v}_t, \mathbf{a}^{(2)}_t = \mathbf{B}_2 \cdot \mathbf{v}_t, \cdots, \mathbf{a}^{(k)}_t = \mathbf{B}_k \cdot \mathbf{v}_t\}$ in time-slot t. Now, if the server was lucky and all of the first k workers were reliable, that is, $\mathcal{L} \cap [k] = \varnothing$, then $\tilde{\mathbf{a}}^{(w)}_t = \mathbf{a}^{(w)}_t$ for all $w \in [k]$ and therefore the correct matrix-vector product $\mathbf{B} \cdot \mathbf{v}_t$ is directly given by the result of the first k workers, without any extra decoding computation cost at the server. However, if one or more of these k workers were unreliable, that is, $\mathcal{L} \cap [k] \neq \varnothing$, then $\tilde{\mathbf{a}}^{(w)}_t$ may not be equal to $\mathbf{a}^{(w)}_t$ for $w \in \mathcal{L} \cap [k]$. In the following, we show that the server can reconstruct the correct answers $\mathbf{a}^{(w)}_t$ for all the unreliable workers $w \in \mathcal{L}$ with vanishing probability of error. Towards this end, we first define a reconstruction criterion as follows.

Reconstruction Criterion: *For any worker $w \in \mathcal{L}$, the parity matrix $\mathsf{M}^{(p)}$ satisfies the reconstruction criterion if $\mathsf{M}^{(p)}$ has a row i whose support $\mathcal{U}_i = \mathsf{supp}\Big(\mathsf{M}^{(p)}_{i,:}\Big)$ has the following properties: (i) $\mathcal{U}_i \cap \mathcal{L} = w$ and (ii) $\mathcal{U}_i \cap \big([n] \setminus \mathcal{L}\big) \neq \varnothing$.*

The following lemma shows that if $\mathsf{M}^{(p)}$ satisfies the reconstruction criterion for a worker $w \in \mathcal{L}$, then the server can reconstruct the correct result $\mathbf{a}^{(w)}_t$ of w. The proof of the lemma is presented in [22, Appendix D].

Lemma 2. *If the parity matrix $\mathsf{M}^{(p)}$ satisfies the reconstruction criterion for a worker $w \in \mathcal{L}$, then the correct result $\mathbf{a}^{(w)}_t$ of w can be reconstructed as follows,*

$$\mathbf{a}^{(w)}_t = \Big(\mathsf{M}^{(p)}_{i,w}\Big)^{-1}\Big(\mathsf{M}^{(p)}_{i,w}\tilde{\mathbf{a}}^{(w)}_t - \Gamma_t(\mathcal{U}_i)\Big),$$

where $i \in [M]$ is a row for which the two conditions of the reconstruction criterion are satisfied and $\mathcal{U}_i = \mathsf{supp}(\mathsf{M}^{(p)}_{i,:})$.

In the next lemma, we show that $\mathsf{M}^{(p)}$ satisfies the reconstruction criterion for all unreliable workers $w \in \mathcal{L}$ with vanishing probability of error. The proof of the lemma is presented in [22, Appendix E].

Lemma 3. *The parity matrix $\mathsf{M}^{(p)}$ constructed from the group testing contact matrix $\mathsf{M}^{(c)}$ satisfies the reconstruction criterion for all the unreliable workers $w \in \mathcal{L}$, with an error probability less than $n^{-\beta}$.*

Since the reconstruction criterion holds for every $w \in \mathcal{L}$ with a probability of error less than $n^{-\beta}$, the server can reconstruct the results of all the unreliable workers with an error probability less than $n^{-\beta} + \frac{M}{|\mathbb{F}|} + n^{-\beta} = 2n^{-\beta} + \frac{M}{|\mathbb{F}|}$, where $n^{-\beta} + \frac{M}{|\mathbb{F}|}$ and $n^{-\beta}$ are upper bounds on the error probabilities for the identification of the unreliable workers and for the reconstruction of the correct results of the unreliable workers, respectively. Once the correct results are reconstructed, the server can use the first k workers' results to obtain the matrix-vector product $\mathbf{B} \cdot \mathbf{v}_t$.

Average Computational Cost of Decoding: On average, each row of $\mathsf{M}^{(p)}$ contains $qn = \frac{\theta n}{L}$ non-zero elements and $\tilde{\mathbf{a}}^{(j)}_t$ has $s = \frac{r}{k}$ elements. Therefore, the computational cost of reconstructing each unreliable worker's answer by Lemma 2 is $O\big(\frac{\theta nr}{kL}\big) = O\big(\frac{nr}{kL}\big)$. Moreover, the total cost of reconstructing all of the L unreliable workers' results is $O\big(\frac{nr}{k}\big)$. Once the unreliable workers' results are reconstructed, the matrix-vector product $\mathbf{B} \cdot \mathbf{v}_t$ is given by stacking the correct results of first k workers, which has no additional cost. Therefore, the computational cost of decoding is $O\big(\frac{nr}{k}\big)$.

REFERENCES

[1] A. Solanki, M. Cardone, and S. Mohajer, "Non-colluding attacks identification in distributed computing," in *2019 IEEE Information Theory Workshop (ITW)*, 2019, pp. 1–5.

[2] T. Tang, R. E. Ali, H. Hashemi, T. Gangwani, S. Avestimehr, and M. Annavaram, "Adaptive verifiable coded computing: Towards fast, secure and private distributed machine learning," in *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 628–638. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/IPDPS53621.2022.00067

[3] Q. Yu, N. Raviv, J. So, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security and privacy," *arXiv:1806.00939*, 2018.

[4] Q. Yu and A. S. Avestimehr, "Entangled polynomial codes for secure, private, and batch distributed matrix multiplication: Breaking the "cubic" barrier," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 245–250.

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

[6] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Coded Fourier transform," 10 2017.

[7] S. Jain, M. Cardone, and S. Mohajer, "Identifying reliable machines for distributed matrix-vector multiplication," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 820–825.

[8] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Statist.*, vol. 14, no. 4, pp. 436–440, 12 1943.

[9] C. M. Verdun, T. Fuchs, P. Harar, D. Elbrächter, D. S. Fischer, J. Berner, P. Grohs, F. J. Theis, and F. Krahmer, "Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies," *Frontiers in Public Health*, vol. 9, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpubh.2021.583377

[10] D. Z. Du and F. K. Hwang, "Combinatorial group testing and its applications," 1993.

[11] C. Chan, P. H. Che, S. Jaggi, and V. Saligrama, "Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms," *2011 49th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2011*, 07 2011.

[12] A. Mazumdar, "Nonadaptive group testing with random set of defectives," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7522–7531, 2016.

[13] A. Barg and A. Mazumdar, "Group testing schemes from codes and designs," *IEEE Transactions on Information Theory*, vol. PP, pp. 1–1, 08 2017.

[14] H. A. Inan, P. Kairouz, M. Wootters, and A. Ozgur, "On the optimality of the Kautz-Singleton construction in probabilistic group testing," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018, pp. 188–195.

[15] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1880–1901, 2012.

[16] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi, "Grotesque: noisy group testing (quick and efficient)," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2013, pp. 1234–1241.

[17] M. Cheraghchi, A. Hormati, A. Karbasi, and M. Vetterli, "Group testing with probabilistic tests: Theory, design and application," *IEEE Trans. on Inf. Theory*, vol. 57, no. 10, pp. 7057–7067, 2011.

[18] A. Mazumdar and S. Mohajer, "Group testing with unreliable elements," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 1–3.

[19] S. Jain, M. Cardone, and S. Mohajer, "Improved bounds for efficiently decodable probabilistic group testing with unreliable items," in *2023 IEEE Information Theory Workshop (ITW)*.

[20] X. Cheng, S. Jaggi, and Q. Zhou, "Generalized group testing," *IEEE Transactions on Information Theory*, vol. 69, no. 3, pp. 1413–1451, 2023.

[21] G. Arpino, N. Grometto, and A. S. Bandeira, "Group testing in the high dilution regime," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1955–1960.

[22] S. Jain, M. Cardone, and S. Mohajer, "Probabilistic group testing in distributed computing with attacked workers," 2023. [Online]. Available: https://arxiv.org/abs/2305.06526

[23] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Transactions on Information Theory*, vol. 58, pp. 1880–1901, 2009.