Uplink-Downlink Tradeoff in Distributed Secure Matrix Multiplication

***Abstract*—**

***Index Terms*—Matrix Multiplication, Security, Interference Alignment, Secret Sharing.**

I. Introduction

II. System Model

We consider the problem of distributed secure matrix multiplication. In this problem, the user has two *confidential* matrices $\boldsymbol{A} \in \mathbb{F}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{F}^{n\times p}$ with elements drawn from a sufficiently large field $\mathbb{F}$. The goal of the user is to retrieve the matrix product $\boldsymbol{AB}$ by using N servers without revealing the identity of both $\boldsymbol{A}$ and $\boldsymbol{B}$ to the *curious* servers. We assume that any set $\mathcal{L} \subseteq [N]$ of $|\mathcal{L}| = \ell \le N$ are colluding, i.e., they collaborate.

To ensure secure matrix multiplication, the user applies encoding functions $\boldsymbol{f} = (f_1, f_2, \ldots, f_N)$ and $\boldsymbol{g} = (g_1, g_2, \ldots, g_N)$ to encode matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. Hereby, f_n and g_n denote the functions that encode matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ for the n-th server. $\tilde{\boldsymbol{A}}_n$ and $\tilde{\boldsymbol{B}}_n$ are the encoded versions of $\boldsymbol{A}$ and $\boldsymbol{B}$ provided to the n-th server; in other words, they are the outputs of encoding functions f_n and g_n, i.e.,

$$\tilde{\boldsymbol{A}}_n = f_n(\boldsymbol{A}),\ \tilde{\boldsymbol{B}}_n = g_n(\boldsymbol{B}).$$

We assume that every server is *honest*, thus the server response $\boldsymbol{Z}_n$ is a deterministic function of $\tilde{\boldsymbol{A}}_n$ and $\tilde{\boldsymbol{B}}_n$, i.e., $H(\boldsymbol{Z}_n|\tilde{\boldsymbol{A}}_n, \tilde{\boldsymbol{B}}_n) = 0$. Upon receiving all server answers $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_N$, the user is able to determine $\boldsymbol{AB}$ by invoking the decoding function $d(\cdot)$, such that $\boldsymbol{AB} = d(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_N)$, or satisfy the *decodability constraint*

$$H(\boldsymbol{AB}|\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_N) = 0.$$

Since servers $n \in \mathcal{L}$ collude and the information-theoretic security has to be preserved, the collection of encoding matrices $\tilde{\boldsymbol{A}}_n$ and $\tilde{\boldsymbol{B}}_n, \forall n \in \mathcal{L}$, denoted by $\tilde{\boldsymbol{A}}_{\mathcal{L}}$ and $\tilde{\boldsymbol{B}}_{\mathcal{L}}$, do not reveal any information on private matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, such that

$$I(\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}; \boldsymbol{A}, \boldsymbol{B}) = 0, \qquad \forall \mathcal{L} \subseteq [N], |\mathcal{L}| = \ell.$$

In this paper, we consider the tradeoff between uplink (UL) and the downlink (DL) rate defined according to:

$$R_{\text{UL}} = \frac{H(\boldsymbol{A}, \boldsymbol{B})}{\sum_{n=1}^{N} H(\tilde{\boldsymbol{A}}_n) + H(\tilde{\boldsymbol{B}}_n)} \qquad \text{and} \tag{1}$$

$$R_{\text{DL}} = \frac{H(\boldsymbol{AB})}{\sum_{n=1}^{N} H(\boldsymbol{Z}_n)}. \tag{2}$$

Obviously, we know that $R_{\text{UL}}, R_{\text{DL}} \ge 0$. In the next two sections, we elaborate on both converse and achievability results on these two metrics.

III. Converse

In the following, we construct two upper bounds on the uplink rate R_{UL}. These bounds are given in the following lemma.

Lemma 1. The uplink rate R_{UL} is bounded from above by

$$R_{\text{UL}} \le \min\left\{R_{\text{DL}}^{\text{one-sided}}, \frac{C_{\text{DL}}}{\gamma}\right\}$$

with $R_{\text{DL}}^{\text{one-sided}} = \frac{N-\ell}{N}$, $\gamma = H(\boldsymbol{AB})/H(\boldsymbol{A},\boldsymbol{B})$ with $\gamma \le 1$.

Proof. We start with the proof of the first bound. Secure computing of $\boldsymbol{AB}$ requires that (i) $H(\boldsymbol{A}, \boldsymbol{B}|\tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}) = 0$ and (ii) $H(\boldsymbol{A}, \boldsymbol{B}|\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}) = H(\boldsymbol{A}, \boldsymbol{B})$ for any $\mathcal{L} \subseteq [N], |\mathcal{L}| = \ell$ (cf. security constraint). Thus, we infer that

$$H(\boldsymbol{A}, \boldsymbol{B}) = I(\tilde{\boldsymbol{A}}_{\mathcal{L}^C}, \tilde{\boldsymbol{B}}_{\mathcal{L}^C}; \boldsymbol{A}, \boldsymbol{B}|\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}})$$

Ignoring the second term of above mutual information term gives the upper bound

$$H(\boldsymbol{A}, \boldsymbol{B}) \le H(\tilde{\boldsymbol{A}}_{\mathcal{L}^C}, \tilde{\boldsymbol{B}}_{\mathcal{L}^C}|\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}). \tag{3}$$

Since there are $\binom{N}{N-\ell}$ possible subsets $\mathcal{L}^C$ of non-colluding servers of size $N - \ell$, we can sum up (3) and obtain

$$\binom{N}{N-\ell} H(\boldsymbol{A}, \boldsymbol{B}) \le \sum_{\substack{\mathcal{L}^C \subseteq [1:N] \\ |\mathcal{L}^C| = N-\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}^C}, \tilde{\boldsymbol{B}}_{\mathcal{L}^C}|\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}})$$

$$\iff H(\boldsymbol{A}, \boldsymbol{B}) \le \frac{N-\ell}{\binom{N}{N-\ell}} \sum_{\substack{\mathcal{L}^C \subseteq [1:N] \\ |\mathcal{L}^C| = N-\ell}} \frac{H(\tilde{\boldsymbol{A}}_{\mathcal{L}^C}, \tilde{\boldsymbol{B}}_{\mathcal{L}^C}|\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}})}{N-\ell}$$

Now, we can apply Han's concentration inequality on conditional entropies to get

$$\begin{aligned} H(\boldsymbol{A}, \boldsymbol{B}) &\le \frac{(N-\ell)}{N} H(\tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}) \\ &\le \frac{(N-\ell)}{N} \sum_{n=1}^{N} H(\tilde{\boldsymbol{A}}_n, \tilde{\boldsymbol{B}}_n) \\ &\le \frac{(N-\ell)}{N} \left(\sum_{n=1}^{N} H(\tilde{\boldsymbol{A}}_n) + H(\tilde{\boldsymbol{B}}_n)\right). \end{aligned}$$

Rearranging above inequality gives the first bound. Next, we establish the second bound. To this end, recall from the decodability constraint that

$$H(\boldsymbol{AB}) = I(\boldsymbol{AB}; \tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}).$$

In the sequel, we upper bound $I(\boldsymbol{AB}; \tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]})$ for $N - 2\ell \ge 0$. To this end, we construct the following three sets:

- $\mathcal{L}_1 \subseteq [1:N]$, $|\mathcal{L}_1| = \ell$
- $\mathcal{L}_2 \subseteq [1:N] \setminus \mathcal{L}_1$, $|\mathcal{L}_2| = \ell$

$$\begin{aligned}
H(\boldsymbol{AB}) &\leq H(\tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}) - \frac{1}{\binom{N}{N-\ell}\binom{N-\ell}{\ell}} \sum_{\substack{\mathcal{L}_1 \subseteq [1:N] \\ |\mathcal{L}_1|=\ell}} \sum_{\substack{\mathcal{L}_2 \subseteq [1:N]\setminus\mathcal{L}_1 \\ |\mathcal{L}_2|=\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}_1}, \tilde{\boldsymbol{B}}_{\mathcal{L}_1} | \boldsymbol{AB}) \\
&\quad - \frac{1}{\binom{N}{N-\ell}\binom{N-\ell}{\ell}} \sum_{\substack{\mathcal{L}_1 \subseteq [1:N] \\ |\mathcal{L}_1|=\ell}} \sum_{\substack{\mathcal{L}_2 \subseteq [1:N]\setminus\mathcal{L}_1 \\ |\mathcal{L}_2|=\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}_2}, \tilde{\boldsymbol{B}}_{\mathcal{L}_2} | \boldsymbol{AB}, \tilde{\boldsymbol{A}}_{\mathcal{L}_1}, \tilde{\boldsymbol{B}}_{\mathcal{L}_1}) \\
&= H(\tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}) - \frac{1}{\binom{N}{N-\ell}\binom{N-\ell}{\ell}} \sum_{\substack{\mathcal{L}_1 \subseteq [1:N] \\ |\mathcal{L}_1|=\ell}} \sum_{\substack{\mathcal{L}_2 \subseteq [1:N]\setminus\mathcal{L}_1 \\ |\mathcal{L}_2|=\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}_1}, \tilde{\boldsymbol{B}}_{\mathcal{L}_1}) \\
&\quad - \frac{1}{\binom{N}{N-\ell}\binom{N-\ell}{\ell}} \sum_{\substack{\mathcal{L}_1 \subseteq [1:N] \\ |\mathcal{L}_1|=\ell}} \sum_{\substack{\mathcal{L}_2 \subseteq [1:N]\setminus\mathcal{L}_1 \\ |\mathcal{L}_2|=\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}_1}, \tilde{\boldsymbol{B}}_{\mathcal{L}_1} | \boldsymbol{AB}, \tilde{\boldsymbol{A}}_{\mathcal{L}_2}, \tilde{\boldsymbol{B}}_{\mathcal{L}_2}) \\
&\overset{(a)}{\leq} H(\tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]}) - \frac{2}{\binom{N}{N-\ell}\binom{N-\ell}{\ell}} \sum_{\substack{\mathcal{L}_1 \subseteq [1:N] \\ |\mathcal{L}_1|=\ell}} \sum_{\substack{\mathcal{L}_2 \subseteq [1:N]\setminus\mathcal{L}_1 \\ |\mathcal{L}_2|=\ell}} H(\tilde{\boldsymbol{A}}_{\mathcal{L}_1}, \tilde{\boldsymbol{B}}_{\mathcal{L}_1} | \boldsymbol{AB}, \tilde{\boldsymbol{A}}_{\mathcal{L}_2}, \tilde{\boldsymbol{B}}_{\mathcal{L}_2}) \\
&\overset{(b)}{\leq} \frac{(N-2\ell)}{N} \sum_{n=1}^{N} H(\tilde{\boldsymbol{A}}_n, \tilde{\boldsymbol{B}}_n) \leq \frac{(N-2\ell)}{N} \left(\sum_{n=1}^{N} H(\tilde{\boldsymbol{A}}_n) + H(\tilde{\boldsymbol{B}}_n) \right)
\end{aligned}$$

- $\mathcal{L}_r = [1:N] \setminus \cup_{i=1}^{2} \mathcal{L}_i$

We sum $I(\boldsymbol{AB}; \tilde{\boldsymbol{A}}_{[1:N]}, \tilde{\boldsymbol{B}}_{[1:N]})$ over all possible sets $\mathcal{L}_1$ and $\mathcal{L}_2$ such that $H(\boldsymbol{AB}) / \sum_{n=1}^{N}(H(\tilde{\boldsymbol{A}}_n) + H(\tilde{\boldsymbol{B}}_n)) \leq C_{\text{DL}} \triangleq \frac{(N-2\ell)}{N}$. The details are provided at the top of this page.

The multiplication of this inequality with the non-negative constant $1/\gamma$ gives the second bound.

□

Remark 1. Note that we can use the proof of the second bound as an alternative to derive the download rate of the distributed secure matrix multiplication problem. This proof differs from the genie-aided bound of our previous work.

Remark 2. The upper bound on R_{UL} suggests a decrease in uplink rates as ℓ increases until at $\ell > \ell_{\max}$, $C_{\text{DL}} = 0$ such that $R_{\text{UL}} = 0$.

IV. Review: Downlink Capacity-Achieving Scheme

Before discussing our scheme that balances uplink rate against downlink rate, we review the downlink capacity-achieving scheme termed *secure cross subspace alignment* (SCSA). We describe its main ingredients being matrix partitioning, user-based encoding, server-based multiplication and user-based decoding.

A. Matrix Partitioning

All schemes do some sort of horizontal and vertical matrix partitioning of both $\boldsymbol{A}$ and $\boldsymbol{B}$. To this end, we define the partitioning operator $\mathsf{PART}\,([v_{\boldsymbol{A}}, h_{\boldsymbol{A}}], [v_{\boldsymbol{B}}, h_{\boldsymbol{B}}])$ (with $h_{\boldsymbol{A}} = v_{\boldsymbol{B}}$), which breaks matrix $\boldsymbol{A}$ into $v_{\boldsymbol{A}} h_{\boldsymbol{A}}$ equal-size sub-matrices $\boldsymbol{A}_{ij} \in \mathbb{F}^{m/v_{\boldsymbol{A}} \times n/h_{\boldsymbol{A}}}, \forall i \in [1:v_{\boldsymbol{A}}], j \in [1:h_{\boldsymbol{A}}]$ and matrix $\boldsymbol{B}$ into $v_{\boldsymbol{B}} h_{\boldsymbol{B}}$ sub-matrices $\boldsymbol{B}_{jk} \in \mathbb{F}^{n/v_{\boldsymbol{B}} \times p/h_{\boldsymbol{B}}}, \forall j \in [1:v_{\boldsymbol{B}}], k \in [1:h_{\boldsymbol{B}}]$ such that

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \dots & \boldsymbol{A}_{1h_{\boldsymbol{A}}} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & \dots & \boldsymbol{A}_{2h_{\boldsymbol{A}}} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{A}_{v_{\boldsymbol{A}}1} & \boldsymbol{A}_{v_{\boldsymbol{A}}2} & \vdots & \boldsymbol{A}_{v_{\boldsymbol{A}}h_{\boldsymbol{A}}} \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} & \dots & \boldsymbol{B}_{1h_{\boldsymbol{B}}} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} & \dots & \boldsymbol{B}_{2h_{\boldsymbol{A}}} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{B}_{v_{\boldsymbol{B}}1} & \boldsymbol{B}_{v_{\boldsymbol{B}}2} & \vdots & \boldsymbol{B}_{v_{\boldsymbol{B}}h_{\boldsymbol{B}}} \end{bmatrix}.$$

Note that the PART-operator works under the assumption that m, n and p are multiple of $v_{\boldsymbol{A}}, h_{\boldsymbol{A}}$ and $h_{\boldsymbol{B}}$, respectively. In SCSA, the user applies $\mathsf{PART}([1,1],[1,r])$ with $r = N - 2\ell$ so that

$$\boldsymbol{AB} = \begin{bmatrix} \boldsymbol{AB}_1 & \boldsymbol{AB}_2 & \dots & \boldsymbol{AB}_r \end{bmatrix}.$$

B. User-Based Encoding

Based on the above partition, the user encodes matrix $\boldsymbol{A}$ and each sub-matrix $\boldsymbol{B}_i$ (destined to the n-th server) individually according to:

$$\tilde{\boldsymbol{A}}_n^{(i)} = \frac{\Delta_n}{(i + \alpha_n)} \Big(\boldsymbol{A} + \sum_{k=1}^{\ell} (i + \alpha_n)^k \boldsymbol{Z}_{ik} \Big), \tag{4}$$

$$\tilde{\boldsymbol{B}}_{in} = \boldsymbol{B}_i + \sum_{k=1}^{\ell} (i + \alpha_n)^k \boldsymbol{Z}'_{ik}, \tag{5}$$

where $\Delta_n = \prod_{u=1}^{r} (u + \alpha_n)$ and $\boldsymbol{Z}_{ik}$, $\boldsymbol{Z}'_{ik}$ represent i.i.d. noise terms to ensure privacy of $\boldsymbol{A}$ and $\boldsymbol{B}$. $\alpha_n, n \in [1:N]$ are distinct elements of

$$\mathbb{G} = \{\alpha_n \in \mathbb{F} \mid \alpha_n + j \neq 0, \forall j \in [1:r]\}. \tag{6}$$

The user then sends the pairs

$$(\tilde{\boldsymbol{A}}_n^{(1)}, \tilde{\boldsymbol{B}}_{1n}), \dots, (\tilde{\boldsymbol{A}}_n^{(r)}, \tilde{\boldsymbol{B}}_{rn})$$

to the n-th server.

C. Server-Based Multiplication

Upon receiving all pairs $\{\tilde{\boldsymbol{A}}_n^{(i)}, \tilde{\boldsymbol{B}}_{in})\}_{i=1}^r$, the n-th server computes

$$\begin{aligned}\boldsymbol{Z}_n &= \sum_{i=1}^{r} \tilde{\boldsymbol{A}}_n^{(i)} \tilde{\boldsymbol{B}}_{in} \\ &= \sum_{i=1}^{r} \frac{\Delta_n}{(i+\alpha_n)} \boldsymbol{A}\boldsymbol{B}_i + \sum_{i=1}^{r} \sum_{j=0}^{2\ell-1} \Delta_n \alpha_n^j \boldsymbol{I}_{ij}, \end{aligned} \tag{7}$$

where $\boldsymbol{I}_{ij}$ denotes the effective interference terms. The n-th server output $\boldsymbol{Z}_n$ is then transferred to the user.

D. User-Based Decoding

The user receives the server responses $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_N$. In SCSA, *all* undesired terms (e.g., $\boldsymbol{A}\boldsymbol{Z}'_{ij}$) disperse to multiple powers $\Delta_n \alpha_n^u$ in the range $u \in [0 : 2\ell-1]$ (The superposition of these terms gives $\boldsymbol{I}_{ij}$). Simultaneously, all desired terms $\boldsymbol{A}\boldsymbol{B}_i$ are distinguishable from another and from the interference by their unique powers $\frac{\Delta_n}{(i+\alpha_n)}$. In other words, the user is able to decode the desired items since it can construct a *full rank* decoding matrix

$$\begin{bmatrix} \frac{\Delta_1}{(1+\alpha_1)} & \dots & \frac{\Delta_1}{(r+\alpha_1)} & \Delta_1 & \Delta_1\alpha_1 & \dots & \Delta_1\alpha_1^{2\ell-1} \\ \frac{\Delta_2}{(1+\alpha_2)} & \dots & \frac{\Delta_2}{(r+\alpha_2)} & \Delta_2 & \Delta_2\alpha_2 & \dots & \Delta_2\alpha_2^{2\ell-1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\Delta_N}{1+\alpha_N} & \dots & \frac{\Delta_N}{(r+\alpha_N)} & \Delta_N & \Delta_N\alpha_N & \dots & \Delta_N\alpha_N^{2\ell-1} \end{bmatrix}$$

from server observations $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_N$. Since r signal dimensions out of $N = r + 2\ell$ total dimensions are occupied by desired sub-matrix products $\boldsymbol{A}\boldsymbol{B}_i, \forall i \in [1 : r]$, the achievable downlink rate becomes

$$R_{\text{DL}}^{\text{SCSA}} = \frac{N - 2\ell}{N}.$$

V. Uplink-Rate Adjustable Schemes

Recall that in SCSA, we apply $\mathsf{PART}([11], [1, r])$, i.e., the matrix $\boldsymbol{A}$ is left without partitioning while matrix $\boldsymbol{B}$ is horizontally partitioned into r sub-matrices. The user conveys to the n-th server r encoded pairs $\{\tilde{\boldsymbol{A}}_n^{(i)}, \tilde{\boldsymbol{B}}_{in}\}$, $i \in [1 : r]$. The transmission of multiple pairs with a low partitioning level to single servers, results in an excessive use of uplink resources.

To make better use of uplink resources, we propose two uplink-rate adjustable SCSA schemes – uplink-adjustable SCSA (USCSA) and group-based, uplink-adjustable SCSA (GSCSA) – which guarantee better uplink performance in exchange for a loss in the downlink rate. As opposed to the classical SCSA of section IV, both schemes use a more general partitioning and a modified encoding strategy. We discuss the details in the next sub-sections.

TABLE I: Applied Matrix Partitioning Strategies Based on the Proposed Schemes

Scheme	Partitioning	Comments
SCSA	$\mathsf{PART}([1, 1], [1, r])$	$r = N - 2\ell$
USCSA	$\mathsf{PART}([v_{\boldsymbol{A}}^{\text{USCSA}}, 1], [1, h_{\boldsymbol{B}}^{\text{USCSA}}])$	
GSCSA	$\mathsf{PART}([v_{\boldsymbol{A}}^{\text{GSCSA}}, 1], [1, 1])$	$v_{\boldsymbol{A}}^{\text{GSCSA}} = v_{\boldsymbol{A}}^{\text{USCSA}} h_{\boldsymbol{B}}^{\text{USCSA}}$

A. Matrix Partitioning

Table I specifies the different partitioning strategies applied by the user for the different schemes. In the sequel, we use $f = v_{\boldsymbol{A}}^{\text{USCSA}}$ and $q = h_{\boldsymbol{B}}^{\text{USCSA}}$ for ease of notation. It is easy to see that the partitioning applied by USCSA is the most general one and subsumes both the partitioning applied in SCSA and GSCSA, e.g., when $f = 1$ and $q = r$, the SCSA matrix partitioning is established. In GSCSA we assign sub-matrices $\boldsymbol{A}_i, i \in [1 : fq]$ into q groups comprised of f sub-matrices per group. Group $j, \forall j \in [1 : q]$ includes sub-matrices $\{\boldsymbol{A}_{(j-1)f+1}, \boldsymbol{A}_{(j-1)f+2}, \dots, \boldsymbol{A}_{jf}\}$. In the sequel, we use the indexing set $\mathcal{I}_j \triangleq \{(j-1)f + 1, (j-1)f + 2, \dots, jf\}$ to refer to the j-th partitioning group.

B. User-Based Encoding

Now, we describe the encoding of both USCSA and GSCSA. We start with USCSA. Under the described partitioning, the encoded matrices destined to the n-th server are

- in case of USCSA for $i \in [1 : q]$

$$\begin{aligned}\tilde{\boldsymbol{A}}_n^{(i)} &= \sum_{j=1}^{f} \frac{\Delta_n}{j + (i-1)f + \alpha_n} \boldsymbol{A}_j + \Delta_n \sum_{k=1}^{\ell} (i+\alpha_n)^{k-1} \boldsymbol{Z}_{ik}, \\ \tilde{\boldsymbol{B}}_{in} &= \boldsymbol{B}_i + \prod_{j=1}^{f} (j + (i-1)f + \alpha_n) \Big(\sum_{k=1}^{\ell} (i+\alpha_n)^{k-1} \boldsymbol{Z}'_{ik} \Big), \end{aligned}$$

- and in case of GSCSA for $i \in [1 : q]$

$$\begin{aligned}\tilde{\boldsymbol{A}}_n^{(i)} &= \sum_{j \in \mathcal{I}_i} \frac{\Delta_n}{j + \alpha_n} \boldsymbol{A}_j + \Delta_n \sum_{k=1}^{\ell} (i+\alpha_n)^{k-1} \boldsymbol{Z}_{ik}, \\ \tilde{\boldsymbol{B}}_n^{(i)} &= \boldsymbol{B} + \prod_{j \in \mathcal{I}_i} (j + \alpha_n) \sum_{k=1}^{\ell} (i+\alpha_n)^{k-1} \boldsymbol{Z}'_{ik}, \end{aligned}$$

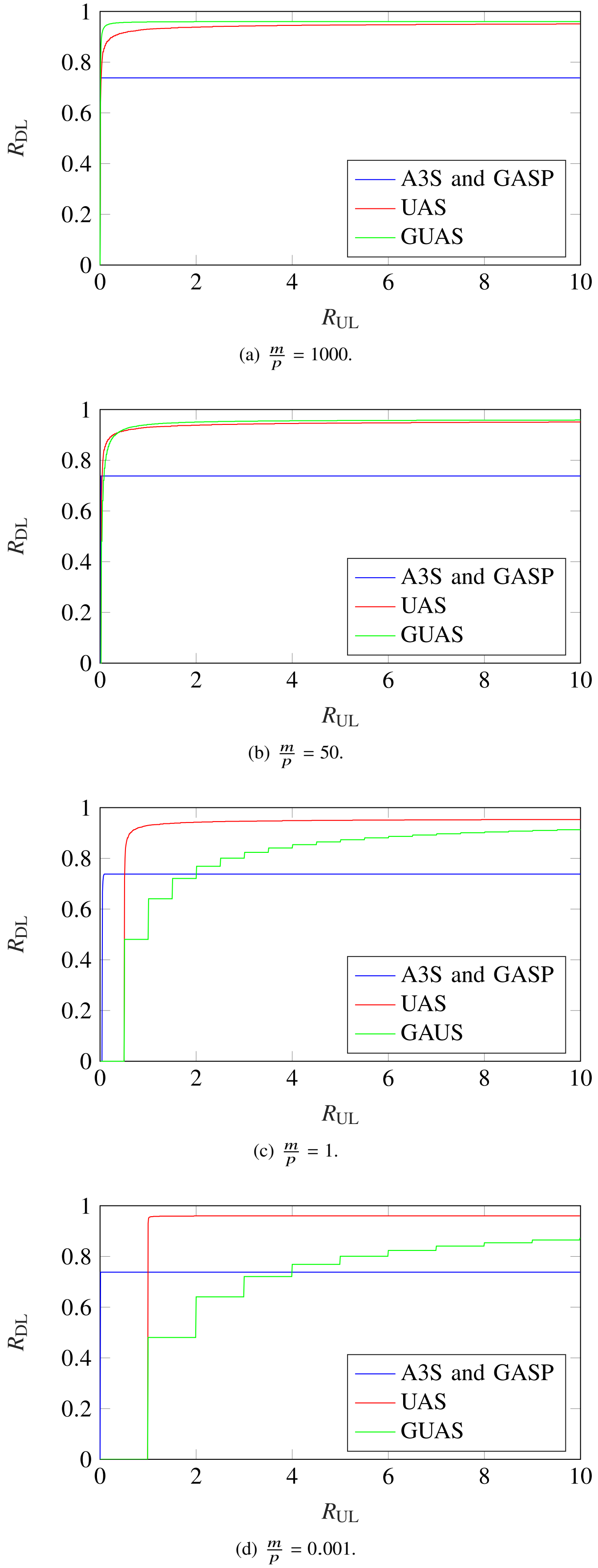
where $\Delta_n = \prod_{u=1}^{fq} (u + \alpha_n)$ and $\boldsymbol{Z}_{ik}$, $\boldsymbol{Z}'_{ik}$ represent i.i.d. noise terms to ensure privacy. The user sends pairs $\{\tilde{\boldsymbol{A}}_n^{(i)}, \tilde{\boldsymbol{B}}_{in}\}_{i=1}^q$ in case of USCSA and $\{\tilde{\boldsymbol{A}}_n^{(i)}, \tilde{\boldsymbol{B}}_n^{(i}\}_{i=1}^q$ in case of GSCSA to the n-th server.

C. Server-Based Multiplication

Every server n multiplies its pair elements and accumulates them to retrieve the server output $\boldsymbol{Z}_n$. Mathematically, the server output becomes

- for USCSA

$$\begin{aligned}\boldsymbol{Z}_n &= \sum_{i=1}^{q} \tilde{\boldsymbol{A}}_n^{(i)} \tilde{\boldsymbol{B}}_{in} \\ &= \sum_{i=1}^{q} \sum_{j=1}^{f} \frac{\Delta_n}{j + (i-1)f + \alpha_n} \boldsymbol{A}_j \boldsymbol{B}_i + \sum_{i=1}^{q} \sum_{j=0}^{2(\ell-1)+f} \Delta_n \alpha_n^j \boldsymbol{I}_{ij} \end{aligned}$$



(a) $\frac{m}{p} = 1000$.

(b) $\frac{m}{p} = 50$.

(c) $\frac{m}{p} = 1$.

(d) $\frac{m}{p} = 0.001$.

Fig. 1: Comparison between the achievable downlink rates for (i) A3S scheme and GASP Codes, (ii) UAS scheme, and (iii) GUAG for $N = 1000$ and $\ell = 20$ as a function of the upper bound on the uplink metirc. The ratio of the number of rows of matrix $\boldsymbol{A}$ and the number of columns of matrix $\boldsymbol{B}$ is investigated for three cases: (a) the first two plots on the top $\frac{m}{p} > 1$, (b) the third plot $\frac{m}{p} = 1$, (c) the third plot at the bottom $\frac{m}{p} < 1$.

- and for GSCSA

$$\boldsymbol{Z}_n = \sum_{i=1}^{q} \tilde{\boldsymbol{A}}_n^{(i)} \tilde{\boldsymbol{B}}_n^{(i)} = \sum_{i=1}^{q} \sum_{j=1}^{f} \frac{\Delta_n}{(i-1)f + j + \alpha_n} \boldsymbol{A}_{(i-1)f+j} \boldsymbol{B} + \sum_{i=1}^{q} \sum_{j=0}^{2(\ell-1)+f} \Delta_n \alpha_n^j \boldsymbol{I}_{ij}.$$

D. User-Based Decoding

Similarly to SCSA, we can derive a decoding matrix for both USCSA and GSCSA by finding a linear representation of $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_N$ as a function of desired sub-matrix products $\boldsymbol{A}_i \boldsymbol{B}_j$ and interfering terms $\boldsymbol{I}_{ij}$. The general *full-rank* decoding matrix for both USCSA and GSCSA is given by

$$\begin{bmatrix} \frac{\Delta_1}{(1+\alpha_1)} & \ldots & \frac{\Delta_1}{(fq+\alpha_1)} & \Delta_1 & \Delta_1\alpha_1 & \ldots & \Delta_1\alpha_1^{2(\ell-1)+f} \\ \frac{\Delta_2}{(1+\alpha_2)} & \ldots & \frac{\Delta_2}{(fq+\alpha_2)} & \Delta_2 & \Delta_2\alpha_2 & \ldots & \Delta_2\alpha_2^{2(\ell-1)+f} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\Delta_N}{1+\alpha_N} & \ldots & \frac{\Delta_N}{(fq+\alpha_N)} & \Delta_N & \Delta_N\alpha_N & \ldots & \Delta_N\alpha_N^{2(\ell-1)+f} \end{bmatrix}.$$

This matrix has the dimension $N \times Q$ with

$$Q = fq + 2\ell + f - 1.$$

Since there are at most N server observations accessible, the user selects partitioning levels f and q such that $Q \leq N$. Overall the user recovers fq desired items from Q overall items (including $2\ell + f - 1$ interference terms $\boldsymbol{I}_{ij}$); thus, attaining a rate

$$R_{\mathrm{DL}}^{\mathrm{USCSA}} = R_{\mathrm{DL}}^{\mathrm{GSCSA}} = \frac{fq}{fq + 2\ell + f - 1}.$$

By choosing f and q while maintaining $Q \leq N$, we can flexibly balance the matrix partitioning (and ultimately the uplink rate) against the downlink rate. For the special case, when $f = 1$ and $q = N - 2\ell$, the SCSA downlink rate is obtained.

VI. Numerical Results