Rate-Efficiency and Straggler-Robustness through Partition in Distributed Two-Sided Secure Matrix Computation

Abstract

Computationally efficient matrix multiplication is a fundamental requirement in various fields, including and particularly in data analytics. To do so, the computation task of a large-scale matrix multiplication is typically outsourced to multiple servers. However, due to data misusage at the servers, security is typically of concern. In this paper, we study the two-sided secure matrix multiplication problem, where a user is interested in the matrix product $\boldsymbol{AB}$ of two finite field private matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ from an information-theoretic perspective. In this problem, the user exploits the computational resources of N servers to compute the matrix product, but simultaneously tries to conceal the private matrices from the servers. Our goal is twofold: (i) to *maximize* the communication rate, and, (ii) to *minimize* the effective number of server observations needed to determine $\boldsymbol{AB}$, while preserving security, where we allow for up to $\ell \leq N$ servers to collude. To this end, we propose a general aligned secret sharing scheme for which we optimize the matrix partition of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in order to either optimize objective (i) or (ii) as a function of the system parameters (e.g., N and ℓ). A proposed *inductive* approach gives us *analytical, close-to-optimal* solutions for both (i) and (ii). With respect to (i), our scheme significantly outperforms the existing scheme of *Chang and Tandon* in terms of (a) communication rate, (b) maximum tolerable number of colluding servers and (c) computational complexity.

Index Terms

Matrix Multiplication, Security, Interference Alignment, Secret Sharing, Straggler Mitigation.

I. Introduction

In machine learning and scientific computation, matrix multiplication plays an important role. However, in many cases high memory requirements and computational effort is required. Distributed approaches have been used to circumvent computational and memory related

barriers of matrix multiplication [1]–[4]. Although distributed matrix multiplication can resolve computational and memory related difficulties, it causes new security problems. In the cryptography literature, different schemes have been proposed that balance security and efficiency of distributed matrix multiplication. *Bultel et al.* [5] suggest partially homomorphic encryption approaches in the framework of MapReduce matrix multiplication. In other related works, cryptographic techniques are applied to the problem of distributed matrix multiplication in cloud computing [6], [7].

As opposed to cryptographic techniques, *information-theoretic* techniques have been hardly applied to the problem of secure matrix multiplication. In [8], *Nodehi and Maddah-Ali* apply information theory to the framework of *limited-sharing multi-party computation*. In limited-sharing multi-party computation, a set of sources offload the computation task, i.e., computing a polynomial function of input matrices available at the sources, to a set of servers. The result of the computation has to be delivered to a master node. The authors propose an efficient *polynomial sharing* scheme that minimizes the number of required servers (which is known as *recovery threshold*) while preserving the privacy of colluding servers and the master. Similar schemes have been applied to the context of non-uniform computation delays at the servers [9].

In [10], *Chang and Tandon* study the communication rate of a secure matrix multiplication problem consisting of a single user and N curious servers which are responsible for the computation of two matrices available at the user. The communication rate is seeked to be maximized when ℓ servers collude. The authors divide the security problem in two models.

(i) **One-sided:** Only one of the two matrices is private. The other matrix is publicly available at all servers.

(ii) **Two-sided:** Both matrices stored at the user are private and not available at the servers.

While for the *one-sided model*, they characterize the capacity with respect to the communication rate, the capacity for the *two-sided model* remains unknown. By comparing with the converse, their proposed scheme for the second model seems to be loose in terms of communication rate and the maximum number of tolerable colluding servers supporting a non-zero rate.

In this paper, we propose a novel *aligned secret sharing* scheme under arbitrary matrix partition for the two-sided model to optimize the two conflicting metrics – (i) rate and (ii)

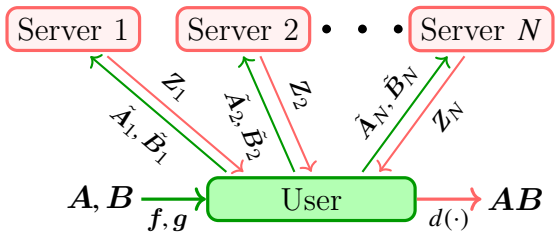

Fig. 1: System model of the two-sided distributed matrix multiplication problem.

recovery threshold. To this end, we formulate two optimization problems, (i) one which *maximizes* the rate and (ii) the other which *minimizes* the number of effective server needed when computing $\boldsymbol{AB}$ subject to a minimum rate constraint. Both optimization problems find the optimal matrix partition of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. Through an inductive approach, we find *analytical, close-to-optimal* solutions of the optimization problems. These solutions identify the optimal matrix partition as a function of N and ℓ and a minimum rate requirement R_{th}. With respect to objective (i), our scheme significantly improves upon the scheme of Chang and Tandon in terms of rate, computational complexity on the servers and the maximum number of tolerable colluding servers. While the maximum number of tolerable colluding servers of the scheme proposed by Chang and Tandon is equal to $\lfloor \sqrt{N} - 1 \rfloor$, our scheme attains a non-zero rate for up to $\lfloor (N-1)/2 \rfloor$ colluding servers. Despite of the higher communication rate in comparison to [10], our scheme attains a lower computational complexity at the servers.

Notations: Throughout this paper, boldface lower-case and capital letters represent vectors and matrices, respectively. Further, for any two integers a, b with $a \leq b$, we define $[a : b] \triangleq \{a, a+1, \ldots, b\}$ and we denote $[1 : b]$ simply as $[b]$. Next, we use the short hand r^+ for $\max\{0, r\}$. $\mathbb{Z}$ refers to the set of all integers, while $\mathbb{Z}^+$ to the subset of positive integers. Random variables X, Y and Z are said for form a Markov chain denoted by $X \to Y \to Z$ if $p(z|x, y) = p(z|y)$.

II. System Model

In a fully, or two-sided, secure matrix multiplication problem, a user is interested in computing the matrix product $\boldsymbol{AB}$ of two finite field *private* matrices[1] $\boldsymbol{A} \in \mathbb{F}^{m \times n}$ and

[1]Each matrix element is from a sufficiently large field $\mathbb{F}$.

$\boldsymbol{B} \in \mathbb{F}^{n\times p}$ securely (see Fig. 1). Hereby, the user has access to a distributed computation system consisting of N honest, but curious computation servers connected to the user by private, error-free links. The user seeks the support of these servers but aims at concealing $\boldsymbol{A}$ and $\boldsymbol{B}$ from the servers.

To this end, the user deploys encoding functions f_i and g_i to generate securely encoded matrices $\tilde{\boldsymbol{A}}_i = f_i(\boldsymbol{A})$ and $\tilde{\boldsymbol{B}}_i = g_i(\boldsymbol{B})$ which are sent to the i-th server. The set of all N encoding functions with respect to matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are denoted by $\boldsymbol{f} = (f_1, \ldots, f_N)$ and $\boldsymbol{g} = (g_1, \ldots, g_N)$, respectively.

Since every server i is by assumption honest, the answer of the i-th server denoted by $\boldsymbol{Z}_i$ is a *deterministic* function[2] of $\tilde{\boldsymbol{A}}_i$ and $\tilde{\boldsymbol{B}}_i$, i.e.,

$$H(\boldsymbol{Z}_i|\tilde{\boldsymbol{A}}_i, \tilde{\boldsymbol{B}}_i) = 0.$$

The user has to be able to determine $\boldsymbol{AB}$ after applying the decoding function $d(\cdot)$ on the collection of all N answers $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N$. i.e., $\boldsymbol{AB} = d(\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N)$, or information-theoretically satisfy the *decodability constraint*

$$H(\boldsymbol{AB}|\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N) = 0. \tag{1}$$

In this paper, we study the (N, ℓ) fully secure matrix multiplication problem. In this setting, security has to be preserved when $\ell \leq N$ servers may collude. In other words, despite having access to the collection of encoded matrices $\tilde{\boldsymbol{A}}_{\mathcal{L}}$ and $\tilde{\boldsymbol{B}}_{\mathcal{L}}$, $\mathcal{L} \subseteq [N], |\mathcal{L}| = \ell$, secrecy has to be maintained. Thus, $\tilde{\boldsymbol{A}}_{\mathcal{L}}$ and $\tilde{\boldsymbol{B}}_{\mathcal{L}}$ do not reveal any information on the private matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. This is expressed information-theoretically by the *security constraint*

$$I(\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}; \boldsymbol{A}, \boldsymbol{B}) = 0, \quad \forall \mathcal{L} \subseteq [N], |\mathcal{L}| = \ell. \tag{2}$$

Next, we define two conflicting metrics – (i) *rate* and (ii) *recovery threshold* – which we seek to optimize in subsequent sections.

First, we say that the rate $R_{N,\ell}$ is *achievable* if there exists $\boldsymbol{f}, \boldsymbol{g}$ and $d(\cdot)$ satisfying the decodability and security constraints. The rate $R_{N,\ell}$ is the ratio between the number of desired bits vs. the number of downloaded bits and is thus given by

$$R_{N,\ell} = \frac{H(\boldsymbol{AB})}{\sum_{i=1}^{N} H(\boldsymbol{Z}_i)}. \tag{3}$$

[2]This function is known by the user.

The *capacity* $C_{N,\ell}$ is the supremum of $R_{N,\ell}$ over all achievable schemes.

Second, we call a secure matrix multiplication strategy to be $\omega_{N,\ell}$-*securely recoverable* if the user can recover the matrix product $\boldsymbol{AB}$ from results of $\Omega \subseteq [N], |\Omega| = \omega_{N,\ell}$ servers while complying with the security constraint when any combination of $\ell \leq N$ servers collude. The *recovery threshold* is the *minimum* integer $\omega_{N,\ell}$ such that the multiplication scheme composed of encoders $\boldsymbol{f}, \boldsymbol{g}$ and decoder $d(\cdot)$ is $\omega_{N,\ell}$-securely recoverable.

III. Aligned secret sharing scheme with Matrix Partition

In an (N, ℓ) fully secure matrix multiplication problem, a user is interested in computing $\boldsymbol{AB}$ using N servers without revealing ℓ colluding servers information about $\boldsymbol{A}$ and $\boldsymbol{B}$. In our proposed aligned secret sharing scheme, the user breaks $\boldsymbol{A}$ vertically into $r_{\boldsymbol{A}}$ sub-matrices and $\boldsymbol{B}$ horizontally into $r_{\boldsymbol{B}}$ sub-matrices, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_{r_{\boldsymbol{A}}} \end{bmatrix} \text{ and } \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \dots & \boldsymbol{B}_{r_{\boldsymbol{B}}} \end{bmatrix}.$$

Thus, we get $\boldsymbol{A}$ and $\boldsymbol{B}$ by concatenating the sub-matrices $\boldsymbol{A}_i \in \mathbb{F}^{(m/r_{\boldsymbol{A}}) \times n},\ i \in [r_{\boldsymbol{A}}]$ and $\boldsymbol{B}_j \in \mathbb{F}^{n \times (p/r_{\boldsymbol{B}})},\ j \in [r_{\boldsymbol{B}}]$. The number of rows m and n are multiple of $r_{\boldsymbol{A}}$ and $r_{\boldsymbol{B}}$, respectively. Under the proposed matrix partition, the matrix product is given by

$$\begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B}_1 & \boldsymbol{A}_1\boldsymbol{B}_2 & \dots & \boldsymbol{A}_1\boldsymbol{B}_{r_{\boldsymbol{B}}} \\ \boldsymbol{A}_2\boldsymbol{B}_1 & \boldsymbol{A}_2\boldsymbol{B}_2 & \dots & \boldsymbol{A}_2\boldsymbol{B}_{r_{\boldsymbol{B}}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{r_{\boldsymbol{A}}}\boldsymbol{B}_1 & \boldsymbol{A}_{r_{\boldsymbol{A}}}\boldsymbol{B}_2 & \dots & \boldsymbol{A}_{r_{\boldsymbol{A}}}\boldsymbol{B}_{r_{\boldsymbol{B}}} \end{bmatrix}.$$

The user encodes the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ according to

$$\tilde{\boldsymbol{A}}_i = \sum_{j=1}^{r_{\boldsymbol{A}}} \boldsymbol{A}_j x_i^{(j-1)} + \sum_{k=1}^{\ell} \boldsymbol{K}_{\boldsymbol{A}_k} x_i^{(k+r_{\boldsymbol{A}}-1)},$$

$$\tilde{\boldsymbol{B}}_i = \sum_{j=1}^{r_{\boldsymbol{B}}} \boldsymbol{B}_j x_i^{(j-1)(r_{\boldsymbol{A}}+\ell)} + \sum_{k=1}^{\ell} \boldsymbol{K}_{\boldsymbol{B}_k} x_i^{(k+r_{\boldsymbol{A}}-1)+(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)},$$

where all entries of matrices $\boldsymbol{K}_{\boldsymbol{A}_1}, \dots, \boldsymbol{K}_{\boldsymbol{A}_\ell} \in \mathbb{F}^{(m/r_{\boldsymbol{A}}) \times n}$ and $\boldsymbol{K}_{\boldsymbol{B}_1}, \dots, \boldsymbol{K}_{\boldsymbol{B}_\ell} \in \mathbb{F}^{n \times (p/r_{\boldsymbol{B}})}$ are i.i.d. uniform random variables. We may represent the collection of observations of all colluding servers $\mathcal{L} = \{i_1, i_2, \dots, i_\ell\}$ as follows.

$$\underbrace{\begin{bmatrix}\tilde{\boldsymbol{A}}_{i_1}\\ \tilde{\boldsymbol{A}}_{i_2}\\ \vdots\\ \tilde{\boldsymbol{A}}_{i_\ell}\end{bmatrix}}_{\triangleq \tilde{\boldsymbol{A}}_{\mathcal{L}}} = \underbrace{\begin{bmatrix} x_{i_1}^0 & x_{i_1}^1 & \dots & x_{i_1}^{r_{\boldsymbol{A}}-1}\\ x_{i_2}^0 & x_{i_2}^1 & \dots & x_{i_2}^{r_{\boldsymbol{A}}-1}\\ \vdots & \vdots & \vdots & \vdots\\ x_{i_\ell}^0 & x_{i_\ell}^1 & \dots & x_{i_\ell}^{r_{\boldsymbol{A}}-1}\end{bmatrix}}_{\triangleq \boldsymbol{R}_{\boldsymbol{A}}} \underbrace{\begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\\ \vdots\\ \boldsymbol{A}_{r_{\boldsymbol{A}}}\end{bmatrix}}_{\triangleq \boldsymbol{A}} + \underbrace{\underbrace{\begin{bmatrix} x_{i_1}^{r_{\boldsymbol{A}}} & 0 & \dots & 0\\ 0 & x_{i_2}^{r_{\boldsymbol{A}}} & \dots & 0\\ \vdots & & \ddots & \\ 0 & 0 & \dots & x_{i_\ell}^{r_{\boldsymbol{A}}}\end{bmatrix}}_{\triangleq \boldsymbol{S}_{\boldsymbol{A}}} \underbrace{\begin{bmatrix} 1 & x_{i_1}^1 & \dots & x_{i_1}^{\ell-1}\\ 1 & x_{i_2}^1 & \dots & x_{i_2}^{\ell-1}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_{i_\ell}^1 & \dots & x_{i_\ell}^{\ell-1}\end{bmatrix}}_{\triangleq \boldsymbol{P}}}_{\triangleq \boldsymbol{P}_{\boldsymbol{A}}} \underbrace{\begin{bmatrix}\boldsymbol{K}_{\boldsymbol{A}_1}\\ \boldsymbol{K}_{\boldsymbol{A}_2}\\ \vdots\\ \boldsymbol{K}_{\boldsymbol{A}_\ell}\end{bmatrix}}_{\triangleq \boldsymbol{K}_{\boldsymbol{A}_{\mathcal{L}}}}$$

$$\underbrace{\begin{bmatrix}\tilde{\boldsymbol{B}}_{i_1}\\ \tilde{\boldsymbol{B}}_{i_2}\\ \vdots\\ \tilde{\boldsymbol{B}}_{i_\ell}\end{bmatrix}}_{\triangleq \tilde{\boldsymbol{B}}_{\mathcal{L}}} = \underbrace{\begin{bmatrix} x_{i_1}^0 & x_{i_1}^{r_{\boldsymbol{A}}+\ell} & \dots & x_{i_1}^{(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)}\\ x_{i_2}^0 & x_{i_2}^{r_{\boldsymbol{A}}+\ell} & \dots & x_{i_2}^{(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)}\\ \vdots & \vdots & \vdots & \vdots\\ x_{i_\ell}^0 & x_{i_\ell}^{r_{\boldsymbol{A}}+\ell} & \dots & x_{i_\ell}^{(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)}\end{bmatrix}}_{\triangleq \boldsymbol{R}_{\boldsymbol{B}}} \underbrace{\begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{B}_2\\ \vdots\\ \boldsymbol{B}_{r_{\boldsymbol{A}}}\end{bmatrix}}_{\triangleq \boldsymbol{B}'}$$

$$+ \underbrace{\underbrace{\begin{bmatrix} x_{i_1}^{r_{\boldsymbol{A}}+(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)} & 0 & \dots & 0\\ 0 & x_{i_2}^{r_{\boldsymbol{A}}+(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)} & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & x_{i_\ell}^{r_{\boldsymbol{A}}+(r_{\boldsymbol{B}}-1)(r_{\boldsymbol{A}}+\ell)}\end{bmatrix}}_{\triangleq \boldsymbol{S}_{\boldsymbol{B}}} \underbrace{\begin{bmatrix} 1 & x_{i_1}^1 & \dots & x_{i_1}^{\ell-1}\\ 1 & x_{i_2}^1 & \dots & x_{i_2}^{\ell-1}\\ \vdots & \vdots & \vdots & \vdots\\ 1 & x_{i_\ell}^1 & \dots & x_{i_\ell}^{\ell-1}\end{bmatrix}}_{\triangleq \boldsymbol{P}}}_{\triangleq \boldsymbol{P}_{\boldsymbol{B}}} \underbrace{\begin{bmatrix}\boldsymbol{K}_{\boldsymbol{B}_1}\\ \boldsymbol{K}_{\boldsymbol{B}_2}\\ \vdots\\ \boldsymbol{K}_{\boldsymbol{B}_\ell}\end{bmatrix}}_{\triangleq \boldsymbol{K}_{\boldsymbol{B}_{\mathcal{L}}}}$$

Thus, the security constraint (2) is satisfied since

$$\begin{aligned} I(\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}; \boldsymbol{A}, \boldsymbol{B}) &= I(\tilde{\boldsymbol{A}}_{\mathcal{L}}, \tilde{\boldsymbol{B}}_{\mathcal{L}}; \boldsymbol{A}, \boldsymbol{B}') = I(\boldsymbol{R}_{\boldsymbol{A}}\boldsymbol{A} + \boldsymbol{P}_{\boldsymbol{A}}\boldsymbol{K}_{\boldsymbol{A}_{\mathcal{L}}}, \boldsymbol{R}_{\boldsymbol{B}}\boldsymbol{B}' + \boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{K}_{\boldsymbol{B}_{\mathcal{L}}}; \boldsymbol{A}, \boldsymbol{B}')\\ &\overset{(I)}{=} I(\boldsymbol{P}_{\boldsymbol{A}}^{-1}\boldsymbol{R}_{\boldsymbol{A}}\boldsymbol{A} + \boldsymbol{K}_{\boldsymbol{A}_{\mathcal{L}}}, \boldsymbol{P}_{\boldsymbol{B}}^{-1}\boldsymbol{R}_{\boldsymbol{B}}\boldsymbol{B}' + \boldsymbol{K}_{\boldsymbol{B}_{\mathcal{L}}}; \boldsymbol{A}, \boldsymbol{B}') = I(\boldsymbol{K}_{\boldsymbol{A}_{\mathcal{L}}}, \boldsymbol{K}_{\boldsymbol{B}_{\mathcal{L}}}; \boldsymbol{A}, \boldsymbol{B}') = 0, \end{aligned} \tag{4}$$

where (I) is since the inverses $\boldsymbol{P}_{\boldsymbol{A}}^{-1}$ and $\boldsymbol{P}_{\boldsymbol{B}}^{-1}$ exist. These two inverses exist since $\boldsymbol{P}_{\boldsymbol{A}}$ and $\boldsymbol{P}_{\boldsymbol{B}}$ are the product of diagonal matrices $\boldsymbol{S}_{\boldsymbol{A}}$ and $\boldsymbol{S}_{\boldsymbol{B}}$ and the $\ell \times \ell$ Vandermonde matrix $\boldsymbol{P}$. The diagonal matrices are non-singular when their diagonal elements are all non-zero. Further, the Vandermonde matrix is invertible when the x_{i_j} are distinct. A field size $|\mathbb{F}| \geq N + 1$ is sufficient to ensure the invertibility of $\boldsymbol{P}_{\boldsymbol{A}}$ and $\boldsymbol{P}_{\boldsymbol{B}}$ (see decodability discussions of $\boldsymbol{AB}$ below).

The exponents of the x_i are carefully chosen to facilitate the alignment of undesired components [11]. Details are discussed in the next paragraph. The user sends the pair $(\tilde{\boldsymbol{A}}_i, \tilde{\boldsymbol{B}}_i)$ to the server where server i in return computes $\boldsymbol{Z}_i = \tilde{\boldsymbol{A}}_i\tilde{\boldsymbol{B}}_i$ and sends its *answer* $\boldsymbol{Z}_i$ back to the user. The user seeks to retrieve $\boldsymbol{AB}$ by observing up to N polynomials $p(x_i)$,

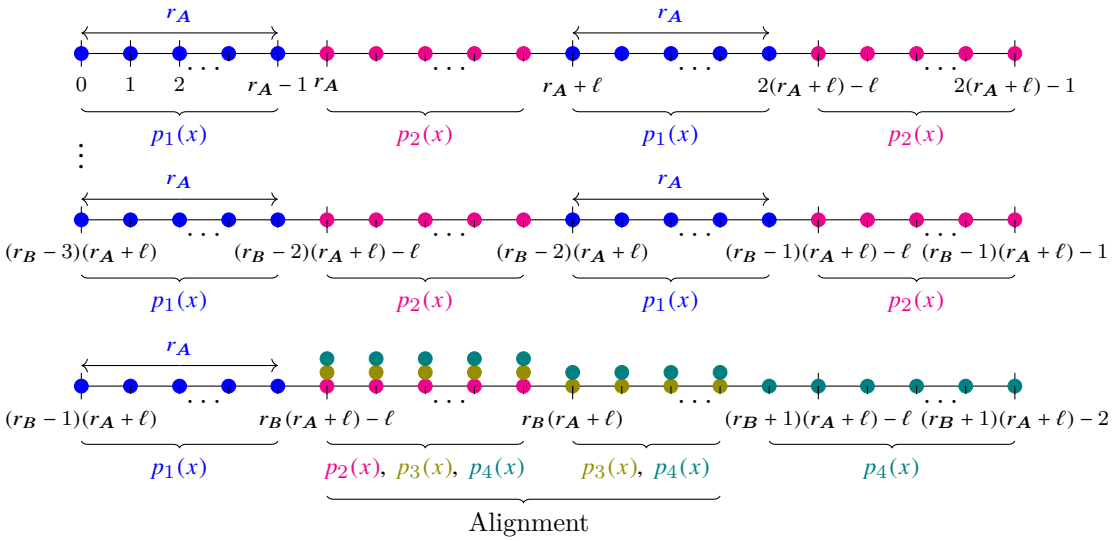

Fig. 2: Number line of the exponent of the polynomial $p(x)$ and its association to the terms $p_i(x), i = 1, \ldots, 4$.

$i = 1, \ldots, N$ of degree $Q_{N,\ell} - 1$, where $Q_{N,\ell} \triangleq (r_A + \ell)(r_B + 1) - 1$[3]. Each polynomial $p(x_i)$, $i = 1, \ldots, N$ has the form $p(x) = \sum_{i=1}^{4} p_i(x)$ (for $x = x_i$). Explicitly $p(x)$ is given by

$$p(x) = \underbrace{\sum_{j=1}^{r_A}\sum_{j'=1}^{r_B} \boldsymbol{A}_j \boldsymbol{B}_{j'} x^{j+(j'-1)(r_A+\ell)-1}}_{\triangleq p_1(x)} + \underbrace{\sum_{k=1}^{\ell}\sum_{j'=1}^{r_B} \boldsymbol{K}_{\boldsymbol{A}_k} \boldsymbol{B}_{j'} x^{k+r_A+(j'-1)(r_A+\ell)-1}}_{\triangleq p_2(x)}$$
$$+ \underbrace{\sum_{j=1}^{r_A}\sum_{k'=1}^{\ell} \boldsymbol{A}_j \boldsymbol{K}_{\boldsymbol{B}_{k'}} x^{j+k'+(r_A-1)+(r_B-1)(r_A+\ell)-1}}_{\triangleq p_3(x)} + \underbrace{\sum_{k=1}^{\ell}\sum_{k'=1}^{\ell} \boldsymbol{K}_{\boldsymbol{A}_k} \boldsymbol{K}_{\boldsymbol{B}_{k'}} x^{k+k'+2(r_A-1)+(r_B-1)(r_A+\ell)}}_{\triangleq p_4(x)}. \tag{5}$$

Alternatively, we may present all N polynomials compactly by

$$\underbrace{\begin{bmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_N) \end{bmatrix}}_{\triangleq \boldsymbol{p}} = \underbrace{\begin{bmatrix} x_1^0 & x_1^1 & \ldots & x_1^{Q_{N,\ell}-1} \\ x_2^0 & x_2^1 & \ldots & x_2^{Q_{N,\ell}-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_N^0 & x_N^1 & \ldots & x_2^{Q_{N,\ell}-1} \end{bmatrix}}_{\triangleq \boldsymbol{D}_{\text{align}}} \underbrace{\begin{bmatrix} \boldsymbol{A}_1\boldsymbol{B}_1 \\ \boldsymbol{A}_2\boldsymbol{B}_1 \\ \vdots \\ \boldsymbol{K}_{\boldsymbol{A}_\ell}\boldsymbol{K}_{\boldsymbol{B}_\ell} \end{bmatrix}}_{\triangleq \boldsymbol{s}}. \tag{6}$$

[3]We frequently omit using the first or the second subscript when N or ℓ remain constant, e.g., we simply write Q to denote $Q_{N,\ell}$ for constant (N, ℓ). In almost all cases where N is of no concern, we omit the first index and write Q_ℓ.

To reconstruct $\boldsymbol{AB}$, the user is interested in $p_1(x)$. The remaining terms $p_i(x), i = 2, \ldots, 4$, can be thought of *interference*. Thus, with respect to $p_1(x)$ each exponent in x needs to have only one attributable item $\boldsymbol{A}_j\boldsymbol{B}_{j'}$ to distinguish desired components from each other and also from undesired components $\boldsymbol{K}_{\boldsymbol{A}_k}\boldsymbol{B}_{j'}$, $\boldsymbol{A}_j\boldsymbol{K}_{\boldsymbol{B}_{k'}}$ and $\boldsymbol{K}_{\boldsymbol{A}_k}\boldsymbol{K}_{\boldsymbol{B}_{k'}}$. One can verify that each exponent of $p_1(x)$ does not occur in the remaining undesired terms $p_i(x), i = 2, \ldots, 4$. In contrast, there are multiple items assigned to the remaining exponents not being included in $p_1(x)$. In other words, we *align* multiple undesired items to single exponents. Thus, this scheme is called an *aligned secret sharing* scheme. A pictorial representation of the association of components to exponents is provided in Fig. 2. The exponents of x_i in $\tilde{\boldsymbol{A}}_i$ and $\tilde{\boldsymbol{B}}_i$ are chosen in the encoding process to avoid overlaps of desired terms (i) with each other and (ii) with undesired terms while simultaneously create as many alignment opportunities as possible when computing $\boldsymbol{Z}_i = \tilde{\boldsymbol{A}}_i\tilde{\boldsymbol{B}}_i$. The desired terms consume $r_{\boldsymbol{A}}r_{\boldsymbol{B}}$ exponents while the interference occupies $\ell(r_{\boldsymbol{B}} + 1) + r_{\boldsymbol{A}} - 1$ exponents. More specifically, the first $\ell(r_{\boldsymbol{B}} - 1)$ components of $p_2(x)$ are not aligned with other interference components of $p_3(x)$ and $p_4(x)$. In contrast, the remaining $r_{\boldsymbol{A}} + 2\ell - 1$ exponents of $p_2(x)$, $p_3(x)$ or $p_4(x)$ are subject to (subspace) alignment of at least two components.

Recall that the polynomial $p(x)$ has a degree of $Q-1$ and the user has access to N observations. In order to enable decoding, we have to ensure that the degree of the polynomial does not exceed the total number of available servers, or observations, N, i.e.,

$$Q_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) \triangleq (r_{\boldsymbol{A}} + \ell)\,(r_{\boldsymbol{B}} + 1) - 1 \leq N. \tag{7}$$

Importantly, we need to ensure that any combination $\boldsymbol{p}(x_{\boldsymbol{\rho}}) \triangleq [p(x_{\rho_1}), p(x_{\rho_2}), \ldots, p(x_{\rho_Q})]$ ($\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_Q]$ with $\rho_i < \rho_j$, $i < j, \forall i, j$) of Q polynomials out of N polynomials (cf. $\boldsymbol{p}$ in (6)) has to enable decodability of $\boldsymbol{s}$. In other words, every $Q \times Q$ sub-matrix (denoted by $\boldsymbol{D}'_{\text{align}}$) of $\boldsymbol{D}_{\text{align}}$ has to be of full rank, or equivalently its determinant $\det(\boldsymbol{D}'_{\text{align}})$ being non-zero. Hereby, $\det(\boldsymbol{D}'_{\text{align}})$ is the classical Vandermonde determinant

$$\prod_{1 \leq i < j \leq Q} (x_{\rho_j} - x_{\rho_i}),$$

which is non-zero in $\mathbb{F}$ whenever $x_{\rho_j} \neq x_{\rho_i}$ [12]. This is feasible as long as $|\mathbb{F}| \geq N + 1$.

The user can retrieve its desired items in $\boldsymbol{s}$ by polynomial interpolation. Since the user recovers $r_{\boldsymbol{A}}r_{\boldsymbol{B}}$ desired items out of $Q_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ calculated items, the aligned secret sharing

scheme achieves a rate of

$$R_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) \triangleq \frac{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}{Q_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})} = \frac{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}{(r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1}.$$

In order to maximize the rate $R_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$[4], we need to solve the optimization problem

$$\max_{r_{\boldsymbol{A}}, r_{\boldsymbol{B}}} \quad \frac{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}{(r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1} \tag{8}$$

$$\text{subject to} \quad (r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1 \leq N \tag{8a}$$

$$r_{\boldsymbol{A}}, r_{\boldsymbol{B}} \in \mathbb{Z}^+. \tag{8b}$$

We denote the optimal decision variables and objective value of 8 by $(r^\star_{\boldsymbol{A}}, r^\star_{\boldsymbol{B}})$ and $R^\star$.

Further, we note that the effective number of server observations the user needs to determine the matrix product $\boldsymbol{AB}$ is Q. Thus, the aligned secret sharing strategy is Q-*securely recoverable*. In order to make the aligned secret sharing scheme less prone to slower computing servers, or *stragglers*, one has to solve the optimization problem

$$\min_{r_{\boldsymbol{A}}, r_{\boldsymbol{B}}} \quad (r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1 \tag{9}$$

$$\text{subject to} \quad \frac{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}{(r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1} \geq R_{\text{th}} \tag{9a}$$

$$(r_{\boldsymbol{A}} + \ell)(r_{\boldsymbol{B}} + 1) - 1 \leq N \tag{9b}$$

$$r_{\boldsymbol{A}}, r_{\boldsymbol{B}} \in \mathbb{Z}^+. \tag{9c}$$

The optimization problems (8) and (9) find the best choice of how to partition the left and right matrices at the user for given N and ℓ (and a minimum rate requirement $R_{\text{th}} \leq R^\star$ in (9)). For both problems, we propose close-to-optimal analytical solutions. The solutions to the rate maximization problem (8) and Q-secure recoverability problem are stated in Theorem 1 and 3, respectively. To differentiate the (optimal) solution of 9 from 8, we use the breve mark ($\breve{}$) instead of the star symbol (*), e.g., $\breve{Q}$ instead of $Q^\star$.

[4]For the sake of simplicity, the notation of $R_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ is aligned with that of $Q_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$.

IV. Secure MatDot Scheme

As opposed to the aligned secret sharing scheme of Section III, we do not divide matrices $\boldsymbol{A}$ vertically and $\boldsymbol{B}$ horizontally, but rather in an opposite manner, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \dots \boldsymbol{A}_r \end{bmatrix} \text{ and } \boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \vdots \\ \boldsymbol{B}_r \end{bmatrix}.$$

The sub-matrices $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$, $i, j \in [r]$, are of dimension $m \times n/r$ and $n/r \times p$, respectively. The desired matrix resembles a (matrix) dot product ('*MatDot*') and is given by $\boldsymbol{AB} = \sum_{t=1}^{r} \boldsymbol{A}_t \boldsymbol{B}_t$. This scheme is based on the recently proposed MatDot scheme of *Dutta et al.* [13]. We adapt this scheme to the context of secure matrix multiplication. To this end, we encode $\boldsymbol{A}$ and $\boldsymbol{B}$ for the i-th server as follows

$$\tilde{\boldsymbol{A}}_i = \sum_{j=1}^{r} \boldsymbol{A}_j x_i^{(j-1)} + \sum_{k=1}^{\ell} \boldsymbol{K}_{\boldsymbol{A}_k} x_i^{(k+r-1)},$$

$$\tilde{\boldsymbol{B}}_i = \sum_{j=1}^{r} \boldsymbol{B}_j x_i^{(r-j)} + \sum_{k=1}^{\ell} \boldsymbol{K}_{\boldsymbol{B}_k} x_i^{(k+r-1)},$$

where the matrices $\boldsymbol{K}_{\boldsymbol{A}_k}$ and $\boldsymbol{K}_{\boldsymbol{B}_k}$, $k \in [\ell]$, are random matrices with elements drawn from an i.i.d. uniform distribution over the field $\mathbb{F}$. Thus, the answer $\boldsymbol{Z}_i = \tilde{\boldsymbol{A}}_i \tilde{\boldsymbol{B}}_i$ of server i for $x = x_i$ becomes

$$p(x) = \underbrace{\sum_{t=1}^{r} \boldsymbol{A}_t \boldsymbol{B}_t x_i^{(r-1)}}_{\text{desired}} + \sum_{j=1}^{r} \sum_{j'=1, j' \neq j}^{r} \boldsymbol{A}_j \boldsymbol{B}_{j'} x^{(r+j-j'-1)} + \sum_{k=1}^{\ell} \sum_{j'=1}^{r} \boldsymbol{K}_{\boldsymbol{A}_k} \boldsymbol{B}_{j'} x^{(k+2r-j'-1)}$$
$$+ \sum_{j=1}^{r} \sum_{k'=1}^{\ell} \boldsymbol{A}_j \boldsymbol{K}_{\boldsymbol{B}_{k'}} x^{(r+k'+j-2)} + \sum_{k=1}^{\ell} \sum_{k'=1}^{\ell} \boldsymbol{K}_{\boldsymbol{A}_k} \boldsymbol{K}_{\boldsymbol{B}_{k'}} x^{(2r+k+k'-2)}.$$

We observe that the coefficient of $x^{(r-1)}$ in $p(x)$ is in fact the desired matrix product $\boldsymbol{AB} = \sum_{t=1}^{r} \boldsymbol{A}_t \boldsymbol{B}_t$. Further, it is easy to see that the polynom $p(x)$ is of degree $Q_{N,\ell}(r) - 1$ where $Q_{N,\ell}(r) = 2r + 2\ell - 1$. Similarly to the discussion of the aligned secret sharing scheme, we can show that as long as $Q_{N,\ell}(r) \leq N$ a field size $|\mathbb{F}| \geq N + 1$ is sufficient to satisfy the decodability and security constraints. To *maximize* the rate, we choose $r = 1$ such that the achievable rate of the *secure MatDot* scheme becomes

$$R_{N,\ell} = \frac{1}{Q_{N,\ell}(r)} = \frac{1}{2\ell + 1} \tag{10}$$

if $2\ell + 1 \leq N$ and 0 otherwise. As far as the recovery threshold is concerned, one can adjust $r \in \mathbb{Z}^+$ satisfying $R_{\text{th}} \leq \frac{1}{2(r+\ell)-1} \leq \frac{1}{2\ell+1}$ such that

$$\omega_{N,\ell} = \min\{2(r+\ell)-1, N\}. \tag{11}$$

V. Secure Cross subspace alignment-based Matrix Multiplication

In addition to the aligned secret sharing scheme, we propose another scheme which aligns undesired components to *multiple* signaling dimensions. This scheme was recently proposed by *Jia et al.* in the context of private information retrieval and termed *cross subspace alignment* [14]. In the following, we apply this scheme to the context of distributed matrix multiplication. Similarly to the aligned secret sharing scheme, the user applies matrix partitioning with $r_{\boldsymbol{A}} = 1$ and $r \triangleq r_{\boldsymbol{B}}$:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_2 & \dots & \boldsymbol{B}_r \end{bmatrix},$$

such that

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

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

$$\tilde{\boldsymbol{A}}_n^{(i)} = \frac{\Delta_n}{(i+\alpha_n)}\Big(\boldsymbol{A} + \sum_{k=1}^{\ell} (i+\alpha_n)^k \boldsymbol{Z}_{ik}\Big), \tag{12}$$

$$\tilde{\boldsymbol{B}}_{in} = \boldsymbol{B}_i + \sum_{k=1}^{\ell} (i+\alpha_n)^k \boldsymbol{Z}'_{ik}, \tag{13}$$

where $\Delta_n = \prod_{u=1}^{r}(u+\alpha_n)$. The collection $\tilde{\boldsymbol{A}}_n = \{\tilde{\boldsymbol{A}}_n^{(1)}, \dots, \tilde{\boldsymbol{A}}_n^{(r)}\}$ and $\tilde{\boldsymbol{B}}_n = \{\tilde{\boldsymbol{B}}_{1n}, \dots, \tilde{\boldsymbol{B}}_{rn}\}$ is then conveyed to the n-th server. This allows server n to compute the answer

$$\boldsymbol{Z}_n = \sum_{i=1}^{r} \tilde{\boldsymbol{A}}_n^{(i)} \tilde{\boldsymbol{B}}_{in} = \sum_{i=1}^{r} \boldsymbol{C}_{in} \tag{14}$$

with

$$\boldsymbol{C}_{in} = \frac{\Delta_n}{(i+\alpha_n)}\Big(\boldsymbol{AB}_i + \sum_{j=1}^{\ell} (i+\alpha_n)^j \big(\boldsymbol{AZ}'_{ij} + \boldsymbol{Z}_{ij}\boldsymbol{B}_i\big) + \sum_{k=1}^{\ell}\sum_{m=1}^{\ell} (i+\alpha_n)^k (i+\alpha_n)^m \boldsymbol{Z}_{ik}\boldsymbol{Z}'_{im}\Big).$$

This expression can be further simplified, namely

$$\boldsymbol{C}_{in} = \underbrace{\frac{\Delta_n}{(i+\alpha_n)}\boldsymbol{A}\boldsymbol{B}_i}_{\text{Desired}} + \underbrace{\Delta_n\Bigg(\sum_{j=1}^{\ell}(i+\alpha_n)^{j-1}\Big(\boldsymbol{A}\boldsymbol{Z}'_{ij}+\boldsymbol{Z}_{ij}\boldsymbol{B}_i\Big)+\sum_{k=1}^{\ell}\sum_{m=1}^{\ell}(i+\alpha_n)^{k+m-1}\boldsymbol{Z}_{ik}\boldsymbol{Z}'_{im}\Bigg)}_{\text{Interference}}. \tag{15}$$

Binomial expansion in the form $(i+\alpha_n)^j = \sum_{t=0}^{j}\binom{j}{t}\alpha_n^t i^{(j-t)}$ enables the *dispersion of interference terms* $\boldsymbol{A}\boldsymbol{Z}'_{ij}+\boldsymbol{Z}_{ij}\boldsymbol{B}_i$ and $\boldsymbol{Z}_{ik}\boldsymbol{Z}'_{im}$ to *multiple* effective coefficients $\Delta_n\alpha_n^u$ for $u \in [0:2\ell-1]$. Thus, we may reformulate $\boldsymbol{C}_{in}$ compactly as follows

$$\boldsymbol{C}_{in} = \frac{\Delta_n}{(i+\alpha_n)}\boldsymbol{A}\boldsymbol{B}_i + \sum_{j=0}^{2\ell-1}\Delta_n\alpha_n^j\boldsymbol{I}_{ij},$$

where $\boldsymbol{I}_{ij}$ denotes the sum of interference terms attributed to $\boldsymbol{A}\boldsymbol{Z}'_{ij}+\boldsymbol{Z}_{ij}\boldsymbol{B}_i$ and $\boldsymbol{Z}_{ik}\boldsymbol{Z}'_{im}$. (The exact form of $\boldsymbol{I}_{ij}$ is of negligible importance in the construction of the achievable scheme.) Thus, the n-th answer the user receives, becomes

$$\boldsymbol{Z}_n = \sum_{i=1}^{r}\frac{\Delta_n}{(i+\alpha_n)}\boldsymbol{A}\boldsymbol{B}_i + \sum_{i=1}^{r}\sum_{j=0}^{2\ell-1}\Delta_n\alpha_n^j\boldsymbol{I}_{ij}. \tag{16}$$

If the user has all N answers $\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_n$ available, it can determine all r desired sub-matrices $\boldsymbol{A}\boldsymbol{B}_i$, $i=1,\ldots,r$ as long as the decoding matrix

$$\boldsymbol{D}_{\text{cross}} = \begin{bmatrix} \frac{\Delta_1}{(1+\alpha_1)} & \cdots & \frac{\Delta_1}{(r+\alpha_1)} & \Delta_1 & \Delta_1\alpha_1 & \cdots & \Delta_1\alpha_1^{2\ell-1} \\ \frac{\Delta_2}{(1+\alpha_2)} & \cdots & \frac{\Delta_2}{(r+\alpha_2)} & \Delta_2 & \Delta_2\alpha_2 & \cdots & \Delta_2\alpha_2^{2\ell-1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\Delta_N}{(1+\alpha_N)} & \cdots & \frac{\Delta_N}{(r+\alpha_N)} & \Delta_N & \Delta_N\alpha_N & \cdots & \Delta_N\alpha_N^{2\ell-1} \end{bmatrix} \tag{17}$$

of dimension $N\times(r+2\ell)$ is full rank. In [14], the authors show that $\boldsymbol{D}_{\text{cross}}$ is in fact full rank when $|\mathbb{F}| \geq N+r$. Further, it is easy to proof that the security constraint (2) is satisfied. To *maximize the rate*, on the one hand, we choose $r = N-2\ell$ such that

$$R_{N,\ell} = \frac{r}{N} = 1-\frac{2\ell}{N}. \tag{18}$$

On the other hand, we can flexibly choose $r \in \mathbb{Z}^+$ satisfying $R_{\text{th}} \leq r/N \leq 1-2\ell/N$ to adjust the achievable recovery threshold (at the cost of a reduced rate) to

$$\omega_{N,\ell} = \min\{r+2\ell, N\}. \tag{19}$$

Remark 1 (J-sided matrix multiplication). In analogy to the construction of the achievable scheme applicable only to the product of two matrices, the scheme can be extended to the

computation of $\prod_{j=1}^{J} \boldsymbol{M}_j$, where $\boldsymbol{M}_j$ is the j-th matrix from the left. Choosing the first $J-1$ matrices similarly to (12) and $\boldsymbol{M}_J$ according to (13), we can attain a rate of $R_{N,\ell} = 1 - \frac{J\ell}{N}$. The achievable recovery threshold, on the other hand, becomes $\omega_{N,\ell} = \min\{r + J\ell, N\}$ with r representing the partition level of $\boldsymbol{M}_J$.

Remark 2 (Analogy to X-secure T-private information retrieval). In the X-secure T-private information retrieval (XSTPIR) problem [14], [15], a user wants to obtain a file W_θ (included in a library of K files, $W_{[K]} \triangleq \{W_1, \ldots, W_K\}$) from N servers under two security constraints in terms of (i) file storage at the servers and (ii) private information retrieval of W_θ. Specifically, (i) ensures that any set of X colluding servers learn nothing about $W_{[K]}$ and (ii) that the identity of the requested file W_θ is protected from any group of T colluding servers. With respect to (ii), the user sends queries $Q_n^{[\theta]}$ to each server $n \in [N]$. *Jia et al.* develop a secure cross subspace alignment scheme which is *asymptotically* ($K \to \infty$) rate optimal. In our work, we adapt this scheme to the problem of distributed matrix multiplication. Specifically, we use the query design $Q_n^{[\theta]}$ for the encoding design (cf. (12)) of the *unpartitioned* left matrix $\boldsymbol{A}$.

VI. Rate Maximization and Discussion

Theorem 1. The solution $(\hat{r}_{\boldsymbol{A}}, \hat{r}_{\boldsymbol{B}})$ is a close-to-optimal analytical solution to the optimization problem (8) for given parameters N and ℓ. Hereby,

$$\hat{r}_{\boldsymbol{B}} = \left\lceil -\frac{3}{2} + \sqrt{\frac{1}{4} + \frac{N}{\ell}} \right\rceil \tag{20}$$

and $\hat{r}_{\boldsymbol{A}}$ is the largest possible integer $r_{\boldsymbol{A}} \geq 1$ that satisfies the inequality

$$(r_{\boldsymbol{A}} + \ell)(\hat{r}_{\boldsymbol{B}} + 1) - 1 \leq N. \tag{21}$$

Proof. The proof is based on the inductive approach of deriving the relationship between consecutive optimal solution pairs $(r^\star_{\ell-1,\boldsymbol{A}}, r^\star_{\ell-1,\boldsymbol{B}})$ and $(r^\star_{\ell,\boldsymbol{A}}, r^\star_{\ell,\boldsymbol{B}})$. Ultimately, under some additional approximations, this helps us in deriving (20) and (21). For further details, we refer the reader to Section A. □

Remark 3 (Feasibility). For $\ell \in [\ell_{\max}]$ (where $\ell_{\max} = \lfloor \frac{N-1}{2} \rfloor$) Eq. (20) returns a positive $\hat{r}_{\boldsymbol{B}} \in \mathbb{Z}^+$. In fact, the overall solution $(\hat{r}_{\boldsymbol{A}}, \hat{r}_{\boldsymbol{B}})$ is feasible with respect to the optimization problem (8). However, for all $\ell \in [\ell_{\max} + 1 : N]$ Eq. (20) gives $\hat{r}_{\boldsymbol{B}} = 0$ which violates the constraint (**??**). Therefore, Eq. (20) implicitly accounts for the (in)feasibility of (8) for given

ℓ and N. An infeasible solution translates to a zero rate, or mathematically $R^{\star}_{N,\ell} = \hat{R}_{N,\ell} = 0$. Hereby, $R^{\star}_{N,\ell}$ denotes the optimal rate of the optimization problem (8) and $\hat{R}_{N,\ell}$ our proposed estimate.

Remark 4 (Upper bound). The best information-theoretic upper bound known of the two-sided matrix multiplication problem on the rate is derived in [10]. The best known upper bound of the two-sided model is in fact the one-sided model for which the capacity is known to be $C^{\text{one-sided}}_{N,\ell} = \frac{N-\ell}{N}$.

In the following theorem, we improve upon the rate in achievability sense and on the upper bound of the two-sided matrix multiplication problem. This allows us to make information-theoretic optimality claims.

Theorem 2 (Capacity). For the (N, ℓ) two-sided secure matrix multiplication problem (computing $\boldsymbol{AB}$), where left and right matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are information-theoretically secured from any $\ell \leq N$ colluding servers, the capacity is given by

$$C^{\text{two-sided}}_{N,\ell} = \left(1 - \frac{2\ell}{N}\right)^{+}. \tag{22}$$

Proof. The capacity-achieving strategy (lower bound) is the cross subspace alignment-based scheme for $r = N - 2\ell$ (cf. Section V). The upper bound (converse), on the other hand uses a genie-aided approach. Specifically, by providing both user and the servers with appropriate side information, we can transform the (N, ℓ) two-sided matrix multiplication problem into an one-sided $(N - \ell, \ell)$ model for which the capacity is already known (cf. Remark 4). For further details, we relegate the reader to Section VIII. □

Remark 5 (Capacity J-sided matrix multiplication). From Section V and Corollary 1, we deduce that the rate *capacity* of the J-sided (N, ℓ) distributed matrix multiplication problem computing $\prod_{j=1}^{J} \boldsymbol{M}_j$ securely corresponds to $C^{J-\text{sided}}_{N,\ell} = \frac{N-J\ell}{N}$.

Remark 6. Compared to the scheme proposed by Chang and Tandon (CT) [10, Theorem 2], our aligned secret sharing and cross subspace alignment schemes significantly improve on the communication rate (see Fig. 3). This is illustrated in Fig. 3 when comparing the achievable communication rate of 'Secure Cross Alignment', 'Unequal Partition' and 'Equal Partition' with 'Chang and Tandon' (CT). Specifically, while our schemes ensure a *non-zero* rate for at most $\lfloor (N-1)/2 \rfloor$ colluding servers, CTs scheme support only $\lfloor \sqrt{N} - 1 \rfloor$ colluding servers. Further, with respect to the aligned secret sharing scheme, appropriate matrix partition is

of importance when comparing the achievable rates of optimized (or unequal) and equal ($r_A = r_B$) partitions in Fig. 3. The unequal partitions use the partitioning proposed in Theorem 1.

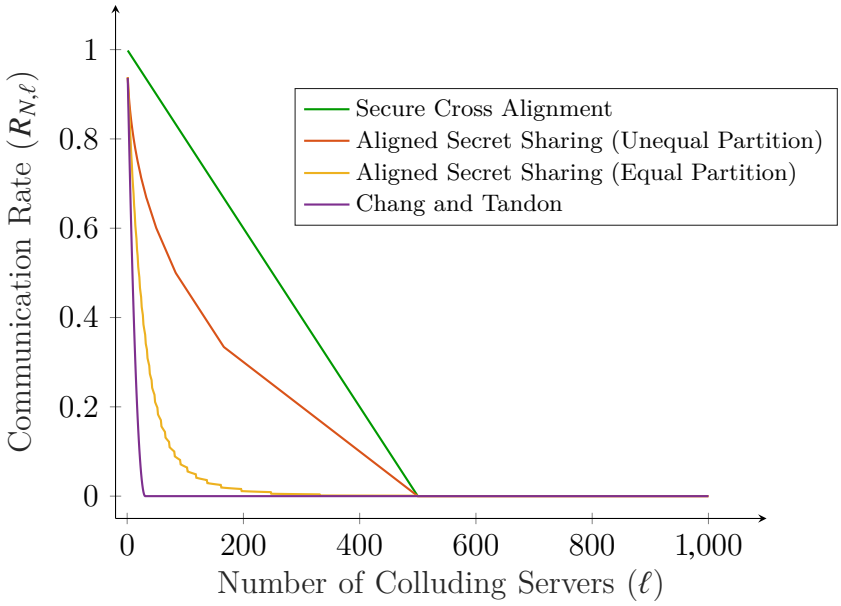

Fig. 3: Comparison between the achievable communication rates for (i) cross subspace alignment-based scheme (which is in fact rate-optimal), (ii) unequally and (iii) equally partitioned aligned secret sharing scheme and (iv) the scheme proposed by Chang and Tandon for $N = 1000$ as a function of the number of colluding servers ℓ.

Remark 7 (Additive gap). To evaluate the quality of our proposed analytical solution to the optimization problem (8), we evaluate the maximum additive gap $\max_{\ell\in[N]} |R_\ell^\star - \hat{R}_\ell|$ (see Fig. **??**). The optimal solution is determined in a brute-force fashion by exhaustive search which is costly in computation. *Numerical* results show that the proposed solution is at most $3 \cdot 10^{-2}$ additively off from the optimal solution.

Remark 8. Our proposed solution to the optimization problem (8) is frequently the optimal solution. Fig. **??** shows the ratio of the number of sub-optimal solutions of the provided estimation to the number of optimal solutions in the optimization problem (8) for constant N and variable ℓ. Fig. **??** suggests that our solution solves (8) for almost all $\ell \in [N]$ optimally except of very few cases where an almost negligible additive gap is attained.

Remark 9 (Server computational complexity). We define the *per-server* computational complexity as the number of necessary multiply-accumulate (MAC) operations to determine

$\boldsymbol{Z}_i = \tilde{\boldsymbol{A}}_i \tilde{\boldsymbol{B}}_i$. Clearly, the MAC complexity under a general $r_{\boldsymbol{A}}$ and $r_{\boldsymbol{B}}$-partition of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ becomes $\Theta\left(\frac{mnp}{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}\right)$. Recall that the denominator $r_{\boldsymbol{A}} r_{\boldsymbol{B}} < N$ represents the dimension reserved for desired sub-block matrix products. For constant N and ℓ, our aligned secret sharing scheme achieves (in comparison to CTs scheme) the better alignment efficiency and thus a larger product $r_{\boldsymbol{A}} r_{\boldsymbol{B}}$. On the other hand, when neglecting the $\Theta(mp)$ additions in the cross subspace alignment scheme (cf. Eq. (14)), this scheme typically produces the largest product $r_{\boldsymbol{A}} r_{\boldsymbol{B}} = N - 2\ell$ ($r_{\boldsymbol{A}} = 1$ and $r_{\boldsymbol{B}} = N - 2\ell$). This in return, results in improved per-server complexities of our proposed schemes (in comparison to CTs scheme) when m, n and p remain constant.

Remark 10 (User decoding complexity). In the aligned secret sharing scheme, the decoding at the user can be interpreted as an interpolation of a $Q-1$-degree polynomial for $\frac{mp}{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}$ times. Hereby, the complexity of a t-degree polynomial interpolation is $O(t \log^2 t \log\log t)$ [16]. Thus, the decoding complexity at the user is of order $O(mp \log^2 \eta \log\log \eta)$ with $\eta = \max\{r_{\boldsymbol{A}}, \ell\} r_{\boldsymbol{B}}$.

Remark 11 (Recovery threshold). In the aligned secret sharing scheme, the effective number of server observations the user needs to determine the matrix product $\boldsymbol{AB}$ is (after rate maximization) $Q^\star$. In the problem (8), it is desirable to choose $r_{\boldsymbol{A}}$ and $r_{\boldsymbol{B}}$ as large as possible without violating the inequality constraint $Q \leq N$. Typically, after rate maximization we obtain highly straggler-dependent solutions for which $Q^\star \approx N$. Similarly, in the cross subspace alignment scheme the achievable recovery maximization is exactly N when maximizing the rate.

Remark 12 (Input matrix dimension). Recall that in the aligned secret sharing scheme the user splits the input matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ into $r_{\boldsymbol{A}}$ sub-matrices $\boldsymbol{A}_i \in \mathbb{F}^{(m/r_{\boldsymbol{A}}) \times n}$ and $r_{\boldsymbol{B}}$ sub-matrices $\boldsymbol{B}_j \in \mathbb{F}^{n \times (p/r_{\boldsymbol{B}})}$. According to Theorem 1, we can easily show from (20)

$$\max\left\{1, -\frac{3}{2} + \sqrt{\frac{N}{\ell}}\right\} \leq r_{\boldsymbol{B}} \leq \sqrt{\frac{N}{\ell}}$$

and based on that from (21)

$$\max\left\{1, \sqrt{N\ell} - \ell - 1\right\} \leq r_{\boldsymbol{A}} \leq 2\sqrt{N\ell} - \ell + 2.$$

This suggests that for feasibility in the matrix partitioning, p and m shall (at least) scale according to $\Theta(\sqrt{N/\ell})$ and $\Theta(\sqrt{N\ell})$, respectively.

VII. Solution of the Matrix Partitioning Problem (9)

Theorem 3. The solution $(\mathring{r}_{\boldsymbol{A}}, \mathring{r}_{\boldsymbol{B}})$ is a close-to-optimal analytical solution to the optimization problem (9) for a given parameter R_{th} which is feasible with respect to the given parameters N und ℓ. Hereby,

$$\mathring{r}_{\boldsymbol{B}} = \left\lceil \frac{2}{1 - R_{\text{th}}} - 2 \right\rceil \tag{23}$$

and $\mathring{r}_{\boldsymbol{A}}$ is the smallest possible integer $r_{\boldsymbol{A}} \geq 1$ that satisfies the inequality

$$\frac{r_{\boldsymbol{A}} \mathring{r}_{\boldsymbol{B}}}{(r_{\boldsymbol{A}} + \ell)\,(\mathring{r}_{\boldsymbol{B}} + 1) - 1} \geq R_{\text{th}}. \tag{24}$$

Proof. The proof is based on comparision with the optimization problem (1). For more details we refer the reader to Section B. □

Remark 13. We measure the accuracy of our solution (9) by considering the maximum relative gap $\max_{R_{\text{th}}, \ell \in [\bar{\ell}]} \frac{|\mathring{Q}_{R_{\text{th}},\ell} - \breve{Q}_{R_{\text{th}},\ell}|}{\mathring{Q}_{R_{\text{th}},\ell}}$ (see Fig. **??**). Hereby, $\breve{Q}_{R_{\text{th}},\ell}$ and $\mathring{Q}_{R_{\text{th}},\ell}$ are, respectively, the optimal and according to Theorem 3 the proposed achievable recovery threshold. *Numerical* results show that the proposed solution relative gap is at most 0.14. With respect to R_{th}, we use in the numerical simulation 100 equidistant values inside the interval $[R^{\star}_{N,\ell}/100, R^{\star}_{N,\ell}]$ for every given pair (N, ℓ). The gap of 0.14 is typically of no concern since the ratio of sub-optimal solutions is less than 0.025 for arbitrary N (see Fig. **??**).

VIII. Converse (Upper Bound on the Rate)

In the sequel, we apply a *genie-aided* upper bound by providing both the user and servers with additional *side information*. Through this technique, we *transform a two-sided* (N, ℓ) *matrix multiplication problem into an one-sided* $(N - \ell, \ell)$ *model*. To distinguish the encoding matrices and answers of the one-sided model from the two-sided, we reserve the '°' notation solely for the one-sided model (e.g., $\mathring{\boldsymbol{B}}_i$ vs. $\boldsymbol{B}_i$). *Without loss of generality*, we assume that the expected number of downloaded bits from each server is of the same value, i.e., $H(\boldsymbol{Z}) = H(\boldsymbol{Z}_i)$ and $H(\mathring{\boldsymbol{Z}}) = H(\mathring{\boldsymbol{Z}}_i), \forall i \in [N]$. Further, consider the case where the set of non-colluding servers $\mathcal{L}^C$ has a *smaller cardinality* than the set of colluding servers $\mathcal{L}$, $\left(|\mathcal{L}^C| = N - \ell \geq \ell = |\mathcal{L}|\right)$.

Let us now elaborate on the construction of the genie. We provide (i) the user with the collection of answers $\boldsymbol{Z}_{\mathcal{L}'}$ from all servers $n \in \mathcal{L}'$ with $\mathcal{L}'$ satisfying $\mathcal{L}' \subseteq \mathcal{L}^C$, $|\mathcal{L}'| = \ell$.

Further, we give all servers (ii) $\boldsymbol{B}_{\mathcal{L}'}$ and (iii) the matrix $\boldsymbol{B}$. Since $(\boldsymbol{A},\boldsymbol{B}) \to (\tilde{\boldsymbol{A}}_{\mathcal{L}'}, \tilde{\boldsymbol{B}}_{\mathcal{L}'}) \to \boldsymbol{Z}_{\mathcal{L}'}$, we know by the data processing inequality that

$$I(\boldsymbol{A},\boldsymbol{B};\tilde{\boldsymbol{A}}_{\mathcal{L}'},\tilde{\boldsymbol{B}}_{\mathcal{L}'}) \geq I(\boldsymbol{A},\boldsymbol{B};\boldsymbol{Z}_{\mathcal{L}'}).$$

Next, due to $H(\boldsymbol{Z}_{\mathcal{L}'}|\boldsymbol{A},\boldsymbol{B}) \leq H(\boldsymbol{Z}_{\mathcal{L}'}|\boldsymbol{AB})$ and the non-negativity of mutual information, one car infer that

$$I(\boldsymbol{A},\boldsymbol{B};\boldsymbol{Z}_{\mathcal{L}'}) \geq I(\boldsymbol{AB};\boldsymbol{Z}_{\mathcal{L}'}) \geq 0,$$

such that

$$I(\boldsymbol{A},\boldsymbol{B};\tilde{\boldsymbol{A}}_{\mathcal{L}'},\tilde{\boldsymbol{B}}_{\mathcal{L}'}) \geq I(\boldsymbol{A},\boldsymbol{B};\boldsymbol{Z}_{\mathcal{L}'}) \geq I(\boldsymbol{AB};\boldsymbol{Z}_{\mathcal{L}'}) \geq 0. \tag{25}$$

From the security constraint $I(\boldsymbol{A},\boldsymbol{B};\tilde{\boldsymbol{A}}_{\mathcal{L}'},\tilde{\boldsymbol{B}}_{\mathcal{L}'}) = 0$ (cf. (2)) and (25), we conclude that $I(\boldsymbol{AB};\boldsymbol{Z}_{\mathcal{L}'}) = 0$ or equivalently $H(\boldsymbol{AB}) = H(\boldsymbol{AB}|\boldsymbol{Z}_{\mathcal{L}'})$. Thus, the user can only decode $\boldsymbol{AB}$ if it obtains the answers $\mathring{\boldsymbol{Z}}_{\tilde{\mathcal{L}}}$ with $\tilde{\mathcal{L}} = \mathcal{L} \cup \{\mathcal{L}^C \setminus \mathcal{L}'\}$.[5] Consequently, from the perspective of the servers $n \in \tilde{\mathcal{L}}$, the one-sided capacity-achieving scheme (of rate $C^{\text{one-sided}}_{N-\ell,\ell}$) with $N-\ell$ available servers, out of which ℓ collude, is applicable. This is mainly because the matrix $\boldsymbol{B}$ is now a *public* matrix. We therefore provide the user with $\mathring{\boldsymbol{Z}}_{\tilde{\mathcal{L}}}$ according to the capacity-achieving one-sided scheme to allow for the decoding of $\boldsymbol{AB}$. Then, the number of bits downloaded *per server* for the one-sided setting corresponds to

$$H(\mathring{\boldsymbol{Z}}) = \frac{H(\boldsymbol{AB})}{(N-\ell)C^{\text{one-sided}}_{N-\ell,\ell}}.$$

Similarly, for the two-sided model we infer

$$H(\boldsymbol{Z}) = \frac{H(\boldsymbol{AB})}{NC^{\text{two-sided}}_{N,\ell}}.$$

Since $H(\mathring{\boldsymbol{Z}}) \leq H(\boldsymbol{Z})$ by construction of the genie, we conclude that

$$C^{\text{two-sided}}_{N,\ell} \leq \frac{(N-\ell)}{N} C^{\text{one-sided}}_{N-\ell,\ell} \overset{(a)}{=} 1 - \frac{2\ell}{N}, \tag{26}$$

where in (a), we used $C^{\text{one-sided}}_{N,\ell} = 1 - \frac{\ell}{N}$ [10, Theorem 1].

Corollary 1. The rate capacity for the J-sided (N,ℓ) secure distributed matrix multiplication problem computing $\prod_{j=1}^{J} \boldsymbol{M}_j$ is upper bounded by

$$C^{J-\text{sided}}_{N,\ell} \leq 1 - \frac{J\ell}{N}. \tag{27}$$

[5]Note that we have constructed the genie such that $\boldsymbol{A}_{\mathcal{L}'}$ and servers within the set $\mathcal{L}'$ are obsolete. Thus, $I(\boldsymbol{AB};\boldsymbol{Z}_{\mathcal{L}'},\mathring{\boldsymbol{Z}}_{\tilde{\mathcal{L}}}) = I(\boldsymbol{AB};\mathring{\boldsymbol{Z}}_{\tilde{\mathcal{L}}})$.

Proof. Proof by induction. The base cases for $J \in \{1, 2\}$ readily follow from [10, Theorem 1] and (26), respectively. Constructing the genie similarly to the two-sided case ($J = 2$), we can easily show that

$$C_{N,\ell}^{J-\text{sided}} \leq \frac{(N-\ell)}{N} C_{N-\ell,\ell}^{(J-1)-\text{sided}}.$$

Now we apply this bound successively for $N \geq J\ell$, i.e.,

$$\begin{aligned} C_{N,\ell}^{J-\text{sided}} &\leq \frac{(N-\ell)}{N} C_{N-\ell,\ell}^{(J-1)-\text{sided}} \leq \frac{(N-\ell)}{N}\frac{(N-2\ell)}{(N-\ell)} C_{N-2\ell,\ell}^{(J-2)-\text{sided}} \leq \dots \\ &\leq \prod_{j=1}^{J-1} \frac{(N-j\ell)}{(N-(j-1)\ell)} C_{N-(J-1)\ell,\ell}^{\text{one-sided}} = \frac{(N-(J-1)\ell)}{N}\left(1 - \frac{\ell}{(N-(J-1)\ell)}\right) = 1 - \frac{J\ell}{N}. \end{aligned}$$

This finalizes the proof. □

IX. Concluding Remarks

In this paper, we studied the two-sided secure matrix multiplication problem, where a user is interested in the matrix product $\boldsymbol{AB}$ of two private matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The user tries to conceal the private matrices from N servers (where we allow for up to ℓ servers to collude), but uses them to compute the matrix product. We propose a partition-based aligned secret sharing scheme. Next, we formulate and solve two optimization problems that determine the optimal matrix partition of input matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to (i) maximize the communication rate of this scheme and (ii) to maximize the recovery threshold. With respect to objective (i), numerical results show that this scheme significantly outperforms the state-of-the-art scheme of Chang and Tandon presented in [10]. In summary, our work shows that appropriate matrix partition is of importance in enabling rate-efficient, straggler-robust and secure two-sided distributed matrix computation.

Appendix A
Close-to-optimal Solution of Optimization Problem (8)

Next, we propose a close-to-optimal solution to the optimization problem (8). To establish this solution, we need the following lemmas.

Lemma 1. For every optimal solution of the optimization problem (8), there is at least one maximizing pair denoted by $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ which satisfies $r_{\boldsymbol{A}}^{\star} \geq r_{\boldsymbol{B}}^{\star}$.

Proof. Proof by contradiction. Suppose that the maximizing pair $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ satisfies $r_{\boldsymbol{A}}^{\star} < r_{\boldsymbol{B}}^{\star}$. The associated number of exploited servers is then given by

$$Q^{\star}(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star}) = (r_{\boldsymbol{A}}^{\star} + \ell)(r_{\boldsymbol{B}}^{\star} + 1) - 1.$$

On the other hand, the associated number of exploited servers for the *inverted* pair $(r_{\boldsymbol{B}}^{\star}, r_{\boldsymbol{A}}^{\star})$ corresponds to

$$Q'(r_{\boldsymbol{B}}^{\star}, r_{\boldsymbol{A}}^{\star}) = (r_{\boldsymbol{B}}^{\star} + \ell)(r_{\boldsymbol{A}}^{\star} + 1) - 1.$$

Subtracting Q' from $Q^{\star}$ gives

$$Q' - Q^{\star} = (r_{\boldsymbol{A}}^{\star} - r_{\boldsymbol{B}}^{\star})(\ell - 1) \leq 0.$$

When $Q' - Q^{\star} \leq 0$, we have

$$\begin{aligned} Q' - Q^{\star} \leq 0 \Leftrightarrow Q' \leq Q^{\star} \\ \Leftrightarrow R(r_{\boldsymbol{B}}^{\star}, r_{\boldsymbol{A}}^{\star}) = \frac{r_{\boldsymbol{A}}^{\star} r_{\boldsymbol{B}}^{\star}}{Q'} \geq \frac{r_{\boldsymbol{A}}^{\star} r_{\boldsymbol{B}}^{\star}}{Q^{\star}} = R^{\star}(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star}). \end{aligned} \tag{28}$$

We infer from inequality (28) that the *inverted* pair $(r_{\boldsymbol{B}}^{\star}, r_{\boldsymbol{A}}^{\star})$ attains a higher rate than $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$. This is in contradiction with the assumption that $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ is a maximizing pair.

□

Lemma 2. When $\ell_{\max} = \lfloor \frac{N-1}{2} \rfloor$ and $N \geq 3$, $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star}) = (1, 1)$ is an *unique* maximizing pair of the optimization problem (8).

Proof. Define

$$\ell_{\max} = \left\lfloor \frac{N-1}{2} \right\rfloor = \begin{cases} \frac{N}{2} - 1 & N \text{ even} \\ \frac{N-1}{2} & N \text{ odd} \end{cases}.$$

We set $(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) = (1 + a, 1 + b)$, where $a, b \in \mathbb{Z}^{+} \cup \{0\}$, such that the number of exploited servers equals

$$\begin{aligned} Q_{N,\ell_{\max}} =& (r_{\boldsymbol{A}} + \ell_{\max})(r_{\boldsymbol{B}} + 1) - 1 \\ =& (a + \ell_{\max} + 1)(b + 2) - 1 \\ =& \begin{cases} N - 1 + 2a + b + ab + \ell_{\max} b & N \text{ even} \\ N + 2a + b + ab + \ell_{\max} b & N \text{ odd} \end{cases}. \end{aligned}$$

The only feasible pair (a, b) satisfying the inequality constraint of the optimization problem (8) is $(a, b) = (0, 0)$. Therefore the only maximizing pair of the optimization problem (8) is $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star}) = (1, 1)$. □

Definition 1. For a given ℓ and N, $(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ is a *strongly feasible* pair of the optimization problem (8) if and only if

(i) it satisfies the inequality constraint $Q_{N,\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) \leq N$,

(ii) and there exists no *feasible* pair $(r'_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ or $(r_{\boldsymbol{A}}, r'_{\boldsymbol{B}})$ with $r'_{\boldsymbol{A}} \geq r_{\boldsymbol{A}}$ or $r'_{\boldsymbol{B}} \geq r_{\boldsymbol{B}}$.

Lemma 3. Every maximizing pair $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ of the optimization problem 8 satisfies the strong feasibility condition.

Proof. The rate for any pair $(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) \in \mathbb{Z}_2^+$ is given by

$$R(r_{\boldsymbol{A}}, r_{\boldsymbol{B}}) = \frac{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}{r_{\boldsymbol{A}} r_{\boldsymbol{B}} + \ell r_{\boldsymbol{B}} + r_{\boldsymbol{A}} + \ell - 1} = \frac{1}{1 + \frac{\ell}{r_{\boldsymbol{A}}} + \frac{1}{r_{\boldsymbol{B}}} + \frac{\ell - 1}{r_{\boldsymbol{A}} r_{\boldsymbol{B}}}}. \tag{29}$$

Suppose by contradiction that the maximizing pair $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ is *not* strongly feasible, i.e., it does not satisfy condition (ii) of Definition 3. Thus, $r_{\boldsymbol{A}}$ or $r_{\boldsymbol{B}}$ can be increased to values above $r_{\boldsymbol{A}}^{\star}$ or $r_{\boldsymbol{B}}^{\star}$ without violating the inequality constraint of the optimization problem. An increase of $r_{\boldsymbol{A}}$ or $r_{\boldsymbol{B}}$ leads to an increase in the rate (cf. Eq. 29). This contradicts that $(r_{\boldsymbol{A}}^{\star}, r_{\boldsymbol{B}}^{\star})$ is a maximizing pair of the optimization problem 8. □

Lemma 4. Let $(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$ be a strongly feasible pair. When ℓ decreases by one $(\ell \leftarrow \ell - 1)$ and we simultaneously increase $r_{\boldsymbol{A}}$ by one $(r_{\boldsymbol{A}} \leftarrow r_{\boldsymbol{A}} + 1)$ while keeping $r_{\boldsymbol{B}}$ constant (i) has no effect on the number of exploited servers and keeps it at $\tilde{Q} \triangleq Q_{\ell}(r_{\boldsymbol{A}}, r_{\boldsymbol{B}})$, (ii) generates a new strongly feasible pair $(r_{\boldsymbol{A}} + 1, r_{\boldsymbol{B}})$ at $\ell - 1$ and (iii) increases the rate additively by $\frac{r_{\boldsymbol{B}}}{\tilde{Q}}$.

Proof. Consider the pairs $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ and $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$, where $r_{2,\boldsymbol{A}} = r_{1,\boldsymbol{A}} + 1$ and $r_{2,\boldsymbol{B}} = r_{1,\boldsymbol{B}}$. The number of exploited servers for the pair $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ with $\ell_2 = \ell_1 - 1$ colluding servers is given by

$$\tilde{Q} \triangleq Q_{\ell_2}(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}}) = (r_{1,\boldsymbol{A}} + \ell_1 + 1 - 1)\,(r_{1,\boldsymbol{B}} + 1) - 1 = Q_{\ell_1}(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}). \tag{30}$$

From $Q_{\ell_2} = Q_{\ell_1}$, (i) and (ii) of Lemma 4 readily follow. The rate associated with the pair $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ then becomes

$$\begin{aligned} R_{\ell_2}(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}}) =& \frac{r_{2,\boldsymbol{A}} r_{2,\boldsymbol{B}}}{Q_{\ell_2}(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})} = \frac{(r_{1,\boldsymbol{A}} + 1) r_{1,\boldsymbol{B}}}{Q_{\ell_1}(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})} \\ =& \frac{r_{1,\boldsymbol{A}} r_{1,\boldsymbol{B}}}{\tilde{Q}} + \frac{r_{1,\boldsymbol{B}}}{\tilde{Q}}. \end{aligned}$$

□

Lemma 5. Suppose that $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ and $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ are two strongly feasible pairs, where $r_{2,\boldsymbol{B}} \geq r_{1,\boldsymbol{B}}$ and $r_{1,\boldsymbol{A}}+1 \geq r_{1,\boldsymbol{B}}$. Decreasing ℓ by one ($\ell \leftarrow \ell-1$) and simultaneously increasing $r_{1,\boldsymbol{A}}$ ($r_{1,\boldsymbol{A}} \leftarrow r_{1,\boldsymbol{A}}+1$) and $r_{2,\boldsymbol{A}}$ ($r_{2,\boldsymbol{A}} \leftarrow r_{2,\boldsymbol{A}}+1$) by one while not changing $r_{1,\boldsymbol{B}}$ and $r_{2,\boldsymbol{B}}$ results in an increase of the rate for both of the pairs $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ and $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$. The additive increase in the rate for the pair $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ is larger than for the pair $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$.

Proof. We have $r_{1,\boldsymbol{A}}+1 \geq r_{1,\boldsymbol{B}}$ so that

$$\begin{aligned} & r_{1,\boldsymbol{B}}^2 \leq (r_{1,\boldsymbol{A}}+1) r_{1,\boldsymbol{B}} \leq (r_{1,\boldsymbol{A}}+\ell) r_{1,\boldsymbol{B}} \leq (r_{1,\boldsymbol{A}}+\ell)(r_{1,\boldsymbol{B}}+1) - 1 = Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) \\ \Leftrightarrow \quad & r_{1,\boldsymbol{B}}^2 + r_{1,\boldsymbol{B}} Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) \leq r_{1,\boldsymbol{B}} Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) + Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) \\ \Leftrightarrow \quad & \frac{r_{1,\boldsymbol{B}}}{Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})} \leq \frac{r_{1,\boldsymbol{B}}+1}{r_{1,\boldsymbol{B}} + Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})} \end{aligned} \tag{31}$$

follows. Since $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ is a strongly feasible pair (cf. Definition 1), neither $r_{1,\boldsymbol{A}}$ nor $r_{1,\boldsymbol{B}}$ can increase while the other element of the pair $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ remains constant. Recall that

$$Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) = (r_{1,\boldsymbol{A}}+\ell)\,(r_{1,\boldsymbol{B}}+1) - 1 \leq N. \tag{32}$$

Incrementing $r_{1,\boldsymbol{A}}$ by one increases Q_ℓ by $r_{1,\boldsymbol{B}}+1$. Similarly, increasing $r_{1,\boldsymbol{B}}$ by one enlarges Q_ℓ by $r_{1,\boldsymbol{A}}+\ell$. Moreover, $r_{1,\boldsymbol{A}} \geq r_{1,\boldsymbol{B}}$ and $\ell \geq 1$ implies

$$r_{1,\boldsymbol{A}} + \ell \geq r_{1,\boldsymbol{B}} + 1.$$

Therefore, strong feasibility along with above observation suggests that Q_ℓ is at least $N - r_{1,\boldsymbol{B}}$. We remind the reader that as long as $Q_\ell < N - r_{1,\boldsymbol{B}}$, the strong feasibility assumption is violated. As a result, the number of exploited servers for the strongly feasible pair $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ is lower bounded according to

$$N - r_{1,\boldsymbol{B}} \leq Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}).$$

On the other hand, the number of exploited servers for the strongly feasible pair $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ is bounded from above by

$$Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}}) \leq N.$$

Therefore

$$\begin{aligned} & Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}}) - Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) \leq r_{1,\boldsymbol{B}} \\ \Leftrightarrow \quad & Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}}) \leq Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}}) + r_{1,\boldsymbol{B}}. \end{aligned} \tag{33}$$

Combining inequalities (31), (33) and $r_{1,\boldsymbol{B}} \leq r_{2,\boldsymbol{B}}$, we get

$$\frac{r_{1,\boldsymbol{B}}}{Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})} \overset{(31)}{\leq} \frac{r_{1,\boldsymbol{B}}+1}{r_{1,\boldsymbol{B}}+Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})} \overset{(33)}{\leq} \frac{r_{1,\boldsymbol{B}}+1}{Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})} \leq \frac{r_{2,\boldsymbol{B}}}{Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})}.$$

Hereby, $\frac{r_{1,\boldsymbol{B}}}{Q_\ell(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})}$ and $\frac{r_{2,\boldsymbol{B}}}{Q_\ell(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})}$ are the corresponding terms by which the rate increases after subjecting the pairs $(r_{1,\boldsymbol{A}}, r_{1,\boldsymbol{B}})$ and $(r_{2,\boldsymbol{A}}, r_{2,\boldsymbol{B}})$ to the mapping of Lemma 4. □

Lemma 6. Consider the optimization problem (8) for a constant N and two consecutive values of ℓ denoted by ℓ_1 and $\ell_2 = \ell_1 - 1$, respectively. The optimal variables for these two problems, represented by $\boldsymbol{r}^\star_{\ell_1} \triangleq (r^\star_{1,\boldsymbol{A}}, r^\star_{1,\boldsymbol{B}})$ if $\ell = \ell_1$ and $\boldsymbol{r}^\star_{\ell_2} \triangleq (r^\star_{2,\boldsymbol{A}}, r^\star_{2,\boldsymbol{B}})$ if $\ell = \ell_2$, have a specific relation. That is, if $\boldsymbol{r}^\star_{\ell_1}$ is known, there are just two possibilities for $\boldsymbol{r}^\star_{\ell_2}$:

(*i*) $\boldsymbol{r}^\star_{\ell_2} = \left(r^\star_{1,\boldsymbol{A}} + 1, r^\star_{1,\boldsymbol{B}}\right)$,

(*ii*) $\boldsymbol{r}^\star_{\ell_2}$ satisfies $r^\star_{2,\boldsymbol{A}} \leq r^\star_{1,\boldsymbol{A}}$ and $r^\star_{2,\boldsymbol{B}} > r^\star_{1,\boldsymbol{B}}$.

Proof. In the following, we go through three possibilities in the choice of $r^\star_{2,\boldsymbol{B}}$ in comparison to $r^\star_{1,\boldsymbol{B}}$: (*i*) $r^\star_{2,\boldsymbol{B}} = r^\star_{1,\boldsymbol{B}}$, (*ii*) $r^\star_{2,\boldsymbol{B}} < r^\star_{1,\boldsymbol{B}}$ and (*iii*) $r^\star_{2,\boldsymbol{B}} > r^\star_{1,\boldsymbol{B}}$.

(*i*) If $r^\star_{2,\boldsymbol{B}} = r^\star_{1,\boldsymbol{B}}$, choose $r^\star_{2,\boldsymbol{A}} = r^\star_{1,\boldsymbol{A}} + a, a \in \mathbb{Z}$. Then, the number of exploited servers becomes:

$$\begin{aligned} Q^\star_{\ell_2}(\boldsymbol{r}^\star_{\ell_2}) &= \left(r^\star_{2,\boldsymbol{A}} + \ell_2\right)\left(r^\star_{2,\boldsymbol{B}} + 1\right) - 1 \\ &= \left(r^\star_{1,\boldsymbol{A}} + a + \ell_1 - 1\right)\left(r^\star_{1,\boldsymbol{B}} + 1\right) - 1 \\ &= \left(r^\star_{1,\boldsymbol{A}} + a' + \ell_1\right)\left(r^\star_{1,\boldsymbol{B}} + 1\right) - 1 \leq N, \end{aligned} \tag{34}$$

where $a' = a - 1, a' \in \mathbb{Z}$. Since the pair $\boldsymbol{r}^\star_{\ell_1}$ is optimal, according to Lemma 3, it must be strongly feasible. Due to the strong feasibility of the pairs $\boldsymbol{r}^\star_{\ell_1}$ and $\boldsymbol{r}^\star_{\ell_2}$, we have $a' = 0$. The other possibilities $a' \in \mathbb{Z}^+$ or $a' \in \mathbb{Z}^-$ are sub-optimal. First, if (34) is satisfied for $a' \in \mathbb{Z}^+$ contradicts with the strong feasibility assumption of the pair $\boldsymbol{r}^\star_{\ell_1}$. Second, if $a' \in \mathbb{Z}^-$, there exists another strongly feasible pair $(r^\star_{2,\boldsymbol{A}} - a', r^\star_{2,\boldsymbol{B}})$ for $\ell = \ell_1$. Then, this strongly feasible pair fulfills the inequality

$$\begin{aligned} Q^\star_{\ell_1}(\boldsymbol{r}^\star_{\ell_1}) &= \left(r^\star_{1,\boldsymbol{A}} + \ell_1\right)\left(r^\star_{1,\boldsymbol{B}} + 1\right) - 1 \\ &= \left(r^\star_{2,\boldsymbol{A}} - a + \ell_2 + 1\right)\left(r^\star_{2,\boldsymbol{B}} + 1\right) - 1 \\ &= \left(r^\star_{2,\boldsymbol{A}} - a' + \ell_2\right)\left(r^\star_{2,\boldsymbol{B}} + 1\right) - 1 \leq N, \end{aligned}$$

where $a' = a - 1$ and $a' \in \mathbb{Z}^-$. However, this is in conflict with the pair $\boldsymbol{r}^\star_{\ell_2}$ being strongly feasible. In summary, this establishes possibility (*i*) of Lemma 6.

(*ii*) If $r^\star_{2,\boldsymbol{B}} < r^\star_{1,\boldsymbol{B}}$, we choose $r^\star_{2,\boldsymbol{A}}$ so that the pair $\tilde{\boldsymbol{r}}_{\ell_1} \triangleq (r^\star_{2,\boldsymbol{A}} - 1, r^\star_{2,\boldsymbol{B}})$ is strongly feasible for ℓ_1. Recall from Lemma 4 that the pair $\boldsymbol{r}^\star_{\ell_2}$ is also strongly feasible if $\ell_2 = \ell_1 - 1$ and N being constant. Simultaneously, the pair $\boldsymbol{r}^\star_{\ell_1}$ maximizes the rate R_{N,ℓ_1}. Thus, we have

$$R_{N,\ell_1}(\tilde{\boldsymbol{r}}^\star_{\ell_1}) \leq R_{N,\ell_1}(\boldsymbol{r}^\star_{\ell_1}). \tag{35}$$

We denote the rate increase from the rate pair $\tilde{\boldsymbol{r}}_{\ell_1}$ to $\boldsymbol{r}^\star_{\ell_2}$ when ℓ decreases by one ($\ell_1 \leftarrow \ell_2$) by $\Delta_{N,\ell_1\leftarrow\ell_2}(\tilde{\boldsymbol{r}}_{\ell_1} \leftarrow \boldsymbol{r}^\star_{\ell_2})$. Similarly, $\Delta_{N,\ell_1\leftarrow\ell_2}(\boldsymbol{r}^\star_{\ell_1} \leftarrow \tilde{\boldsymbol{r}}_{\ell_2})$ refers to the rate increase from $\boldsymbol{r}^\star_{\ell_1}$ to $\tilde{\boldsymbol{r}}_{\ell_2} \triangleq (r^\star_{1,\boldsymbol{A}} + 1, r^\star_{1,\boldsymbol{B}})$. Thus, the overall achievable rates at pairs $\boldsymbol{r}^\star_{\ell_2}$ and $\tilde{\boldsymbol{r}}_{\ell_2}$ correspond to

$$\begin{aligned} R_{N,\ell_2}(\boldsymbol{r}^\star_{\ell_2}) &= R_{N,\ell_1}(\tilde{\boldsymbol{r}}_{\ell_1}) + \Delta_{N,\ell_1\leftarrow\ell_2}(\tilde{\boldsymbol{r}}_{\ell_1} \leftarrow \boldsymbol{r}^\star_{\ell_2}) \\ R_{N,\ell_2}(\tilde{\boldsymbol{r}}_{\ell_2}) &= R_{N,\ell_1}(\boldsymbol{r}^\star_{\ell_1}) + \Delta_{N,\ell_1\leftarrow\ell_2}(\boldsymbol{r}^\star_{\ell_1} \leftarrow \tilde{\boldsymbol{r}}_{\ell_2}) \end{aligned}. \tag{36}$$

Due to Lemma 5, we have $\Delta_{N,\ell_1\leftarrow\ell_2}(\boldsymbol{r}^\star_{\ell_1} \leftarrow \tilde{\boldsymbol{r}}_{\ell_2}) \geq \Delta_{N,\ell_1\leftarrow\ell_2}(\tilde{\boldsymbol{r}}_{\ell_1} \leftarrow \boldsymbol{r}^\star_{\ell_2})$. Consequently, we infer from (35) and (36) that $R_{N,\ell_2}(\tilde{\boldsymbol{r}}_{\ell_2}) \geq R_{N,\ell_2}(\boldsymbol{r}^\star_{\ell_2})$. However, this violates the assumption of optimality at $\boldsymbol{r}^\star_{\ell_2}$. Thus, $\boldsymbol{r}^\star_{\ell_2}$ cannot be a maximizing pair if $r^\star_{2,\boldsymbol{B}} < r^\star_{1,\boldsymbol{B}}$.

(*iii*) If $r^\star_{2,\boldsymbol{B}} > r^\star_{1,\boldsymbol{B}}$, the number of exploited servers $Q^\star_{\ell_2}$ for the pair $\boldsymbol{r}^\star_{\ell_2}$ can be lower bounded according to

$$\begin{aligned} Q^\star_{\ell_2}(\boldsymbol{r}^\star_{\ell_2}) &= \left(r^\star_{2,\boldsymbol{A}} + \ell_2\right)\left(r^\star_{2,\boldsymbol{B}} + 1\right) - 1 \\ &= \left(r^\star_{2,\boldsymbol{A}} - 1 + \ell_1\right)\left(r^\star_{2,\boldsymbol{B}} + 1\right) - 1 \\ &\geq \left(r^\star_{2,\boldsymbol{A}} - 1 + \ell_1\right)\left(r^\star_{1,\boldsymbol{B}} + 2\right) - 1. \end{aligned} \tag{37}$$

Since $Q^\star_{\ell_2}(\boldsymbol{r}^\star_{\ell_2}) \leq N$, we infer that

$$\left(r^\star_{2,\boldsymbol{A}} - 1 + \ell_1\right)\left(r^\star_{1,\boldsymbol{B}} + 2\right) - 1 \leq N. \tag{38}$$

Since the pair $\boldsymbol{r}^\star_{\ell_1}$ is strongly feasible, we deduce from (38) that

$$r^\star_{2,\boldsymbol{A}} - 1 \leq r^\star_{1,\boldsymbol{A}} - 1 \Leftrightarrow r^\star_{2,\boldsymbol{A}} \leq r^\star_{1,\boldsymbol{A}}.$$

Thus, possibility (*iii*) of Lemma 6 is shown.

□

Beginning from $\ell_{\max} = \lfloor \frac{N-1}{2} \rfloor$, where $\boldsymbol{r}^\star_{\ell_{\max}} = (1,1)$ (cf. Lemma 2), we seek to determine the optimal $\boldsymbol{r}^\star_{\ell-1} \triangleq (r^\star_{\ell-1,\boldsymbol{A}}, r^\star_{\ell-1,\boldsymbol{B}})$ from $\boldsymbol{r}^\star_{\ell} \triangleq (r^\star_{\ell,\boldsymbol{A}}, r^\star_{\ell,\boldsymbol{B}})$. To this end, we exploit Lemma 6, which states that if ℓ decreases by 1, either (*i*) $r^\star_{\ell,\boldsymbol{B}} = r^\star_{\ell-1,\boldsymbol{B}}$ or (*ii*) $r^\star_{\ell,\boldsymbol{B}} > r^\star_{\ell-1,\boldsymbol{B}}$. Specifically,

beginning from $\ell_{\max}$, we iteratively move backwards towards $\ell_{\min} = 1$ and *estimate* the values of ℓ at which $r^\star_{\ell,\boldsymbol{B}}$ increases in comparison to previous iterates. This helps us to determine close-to-optimal estimates $\hat{r}_{\ell,\boldsymbol{B}}$ for a given ℓ at constant N. Using these estimates $\hat{r}_{\ell,\boldsymbol{B}}$, one can solve for $\hat{r}_{\ell,\boldsymbol{A}}$ using the inequality of the optimization problem (8).

Let us start with the process of finding ℓ, where $r^\star_{\ell,\boldsymbol{B}}$ must increase compared to $r^\star_{\ell-1,\boldsymbol{B}}$. In other words, we restrict ourself to track at which values of $\ell \in [\ell_{\min} : \ell_{\max}]$, $r^\star_{\ell,\boldsymbol{B}}$ changes. For notational simplicity, we rewrite $r^\star_{\ell_m,\boldsymbol{B}}$ by m and denote its corresponding pair by $r^\star_{\ell_m,\boldsymbol{A}}$. Here, ℓ_m denotes the first iterate[6], or largest ℓ, for which the optimal $r^\star_{\boldsymbol{B}}$ changes from m' to m, or mathematically,

$$\ell_m = \max\{\ell \in [\ell_{\min} : \ell_{\max}] \mid r^\star_{\ell,\boldsymbol{B}} = m\}.$$

According to (*ii*) of Lemma 6, the difference of $r_{\boldsymbol{B}}$-values at neighboring ℓ-values – ℓ_m and $\ell_m - 1$ – i.e.,

$$r^\star_{\ell_m,\boldsymbol{B}} - r^\star_{\ell_m-1,\boldsymbol{B}} = m - m'$$

is lower-bounded by 1. Recall that this difference does not have to be necessarily one. However, if we *relax the integer assumption* on ℓ_m and ℓ_{m-1}, we can find an ℓ_m such that $m = m' + 1$. This is shown in Fig. 4 (by the step functions in the interval $\ell_m \le \ell \le \ell_{m'}$). We will see at the end of this appendix that the values of m which are not associated to (positive) integer-valued ℓ_m are excluded from the results by using the integer assumption. Furthermore, the non-integer values of $\hat{r}_{\ell,\boldsymbol{B}}$ are avoided by applying the ceiling function $\lceil\cdot\rceil$. We define

$$d_m = |\ell_{m'} - \ell_m|$$

as the required number of steps in ℓ needed to change $r^\star_{\boldsymbol{B}}$ from m' to m. In the sequel, we neglect that ℓ, $r_{\boldsymbol{A}}$ and d_m are integer numbers. In the notation this is accounted by using $\hat{\ell}$, $\hat{r}_{\boldsymbol{A}}$ and $\hat{d}_m$, respectively.

Due to Lemma 4, the rate of the optimal pair $(r^\star_{\ell_m,\boldsymbol{A}}, m)$ is larger than the rate of the sub-optimal pair $(r^\star_{\ell_m+1,\boldsymbol{A}} + 1, m')$. This translates to the inequality

$$\frac{m'(r^\star_{\ell_m+1,\boldsymbol{A}} + 1)}{Q^\star_{\ell_m+1}} \le \frac{m r^\star_{\ell_m,\boldsymbol{A}}}{Q^\star_{\ell_m}}. \tag{39}$$

[6]Recall that we move backwards from $\ell_{\max}$ to $\ell_{\min}$.

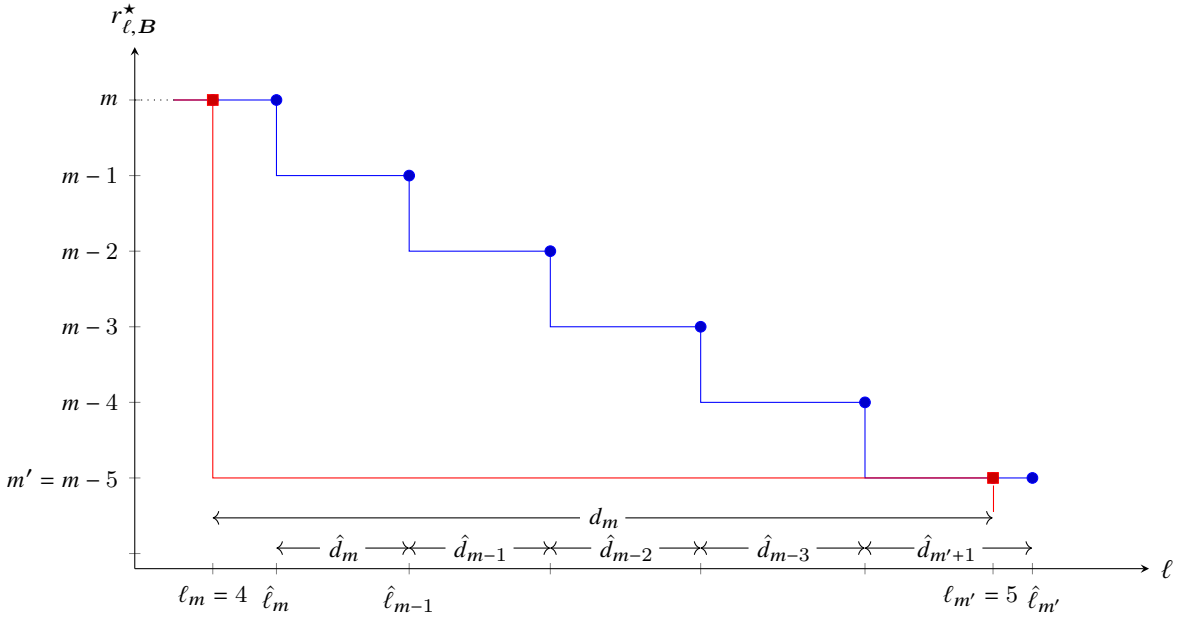

Fig. 4: Illustrative behavior of $r_{\ell,\boldsymbol{B}}$ with respect to $\hat{\ell}_m$ for $N = 10000$, $\ell_m = 4$, $m = r^{\star}_{\ell_m,\boldsymbol{B}} = 49$ and $m' = r^{\star}_{\ell_{m'},\boldsymbol{B}} = 44$. In our approximation we allow for a non-integer relaxation in ℓ_m which we denote by $\hat{\ell}_m$. This allows us to associate values $\hat{\ell}_j$ to $r_{\ell,\boldsymbol{B}} \in [m' : m]$. According to the figure, $\hat{\ell}_m$ is given by $\hat{\ell}_m = \hat{\ell}_{m'} - \sum_{i=m'+1}^{m} \hat{d}_i$ or $\hat{\ell}_m = \ell_{\max} - \sum_{i=2}^{m} \hat{d}_i$.

Applying (i) of Lemma 6 on the pairs $(r^{\star}_{\ell_{m'},\boldsymbol{A}}, m')$ and $(r^{\star}_{\ell_m+1,\boldsymbol{A}}, m')$ leads to

$$r^{\star}_{\ell_m+1,\boldsymbol{A}} = r^{\star}_{\ell_{m'},\boldsymbol{A}} + d_m - 1.$$

We can now reformulate (39) as

$$\frac{m'(r^{\star}_{\ell_{m'},\boldsymbol{A}} + d_m)}{Q^{\star}_{\ell_m+1}} \leq \frac{m r^{\star}_{\ell_m,\boldsymbol{A}}}{Q^{\star}_{\ell_m}}. \tag{40}$$

Assuming that the number of exploited servers is almost constant, i.e., $Q^{\star}_{\ell_m+1} \approx Q^{\star}_{\ell_m}$ allows us to ignore the denominators of the fractions in (40). Further, we apply the *integer relaxation* such that $m' = m - 1$. These approximations allow us to transform (40) to an equality given by[7]

$$\begin{aligned} &(m-1)\,(\hat{r}_{\ell_{m-1},\boldsymbol{A}} + \hat{d}_m) = m\hat{r}_{\ell_m,\boldsymbol{A}} \\ \Leftrightarrow \quad &\hat{d}_m = \frac{m}{m-1}\hat{r}_{\ell_m,\boldsymbol{A}} - \hat{r}_{\ell_{m-1},\boldsymbol{A}}, \end{aligned} \tag{41}$$

[7]Due to these approximations, we replace the "$\star$" superscript with "$\widehat{\ }$".

where $m \in \mathbb{Z}^+ \setminus \{1\}$. Next, we calculate $\hat{Q}_{\ell_m}$ for the pair $(\hat{r}_{\ell_m,\boldsymbol{A}}, m)$ as follows

$$\begin{aligned}
\hat{Q}_{\ell_m} &= (\hat{r}_{\ell_m,\boldsymbol{A}} + \ell_m)\,(m+1) - 1 \\
&= \Big(\hat{r}_{\ell_m,\boldsymbol{A}} + \ell_{\max} - \sum_{i=2}^{m} \hat{d}_i\Big)\,(m+1) - 1 \\
&= \hat{r}_{\ell_m,\boldsymbol{A}}\,(m+1) + \ell_{\max}\,(m+1) - \Big(\sum_{i=2}^{m} \hat{d}_i\Big)\,(m+1) - 1 \\
&\overset{(b)}{\approx} \hat{r}_{\ell_m,\boldsymbol{A}}\,(m+1) + \frac{N}{2}\,(m-1+2) - \left(\sum_{i=2}^{m} \hat{d}_i\right)(m+1) - 1 \\
&= N + \hat{r}_{\ell_m,\boldsymbol{A}}\,(m+1) + \frac{N}{2}\,(m-1) - \left(\sum_{i=2}^{m} \hat{d}_i\right)(m+1) - 1 \\
&\overset{(c)}{=} N - 1,
\end{aligned}$$

where (b) and (c) are due to the approximations $\ell_{\max} \approx \frac{N}{2}$ and $\hat{Q}_{\ell_m} \approx N-1$, respectively. It is easy to conclude from above equality that

$$\hat{r}_{\ell_m,\boldsymbol{A}}\,(m+1) + \frac{N}{2}\,(m-1) - \left(\sum_{i=2}^{m} \hat{d}_i\right)(m+1) = 0. \tag{42}$$

From Eq. (42), we get

$$\sum_{i=2}^{m} \hat{d}_i = \frac{(m-1)}{(m+1)}\frac{N}{2} + \hat{r}_{\ell_m,\boldsymbol{A}}. \tag{43}$$

Similarly, we can approximate $\hat{Q}_{\ell_{m-1}}$ by the same approach such that

$$\sum_{i=2}^{m-1} \hat{d}_i = \frac{(m-2)}{m}\frac{N}{2} + \hat{r}_{\ell_{m-1},\boldsymbol{A}}. \tag{44}$$

Subtracting (44) from (43) gives

$$\begin{aligned}
\hat{d}_m &= \left(\frac{m-1}{m+1} - \frac{m-2}{m}\right)\frac{N}{2} + \hat{r}_{\ell_m,\boldsymbol{A}} - \hat{r}_{\ell_{m-1},\boldsymbol{A}} \\
&= \frac{N}{m\,(m+1)} + \hat{r}_{\ell_m,\boldsymbol{A}} - \hat{r}_{\ell_{m-1},\boldsymbol{A}}.
\end{aligned} \tag{45}$$

From the Equations 41 and 45 $\hat{d}_m$ is given by

$$\begin{aligned}
\hat{d}_m &= \frac{1}{m+1}N - \hat{r}_{\ell_{m-1},\boldsymbol{A}} \\
&\overset{(d)}{=} \frac{1}{m+1}N - \frac{(m-2)}{(m-1)m}N \\
&= \frac{2N}{(m-1)m(m+1)}.
\end{aligned} \tag{46}$$

Note that in (d), we used $\hat{r}_{\ell_m,\boldsymbol{A}} = \frac{(m-1)}{m(m+1)}N$. Considering Eq. (46) and the approximation $\ell_{\max} \approx \frac{N}{2}$, $\hat{\ell}_m$ corresponds to (cf. Fig. 4)

$$\begin{aligned}
\hat{\ell}_m =& \frac{N}{2} - \sum_{i=2}^{m} \hat{d}_i \\
=& \frac{N}{2} - N \sum_{i=2}^{m} \frac{2}{(i-1)i(i+1)} \\
=& \frac{N}{2} - N \left(\sum_{i=2}^{m} \frac{1}{i-1} - \frac{2}{i} + \frac{1}{i+1} \right) \\
=& \frac{N}{2} - N \left[\left(\sum_{i=2}^{m} \frac{1}{i-1} - \frac{1}{i} \right) + \left(\sum_{i=2}^{m} -\frac{1}{i} + \frac{1}{i+1} \right) \right] \\
=& \frac{N}{2} - N \left[1 - \frac{1}{m} - \frac{1}{2} + \frac{1}{m+1} \right] \\
=& \frac{N}{m(m+1)}. \qquad (47)
\end{aligned}$$

Eq. (47) represents an estimate on the number of colluding servers $\hat{\ell}_m$ at which $\hat{r}_{\ell,\boldsymbol{B}}$ increases to m. We use Eq. (47) to determine m as a function of N and ℓ. Specifically, recall that for any $\ell \in (\hat{\ell}_{m+1}, \hat{\ell}_m]$, we know that m is constant and we need to ensure that

$$\ell \geq \hat{\ell}_{m+1},$$

or according to (47) equivalently

$$\begin{aligned}
\ell &\geq \frac{N}{(m+2)(m+1)} \\
\Leftrightarrow m^2 + 3m - \frac{N}{\ell} + 2 &\geq 0. \qquad (48)
\end{aligned}$$

Due to Lemma 1, we choose the smallest m that satisfies (48). This gives us

$$\hat{r}_{\ell,\boldsymbol{B}} = \left\lceil -\frac{3}{2} + \sqrt{\frac{1}{4} + \frac{N}{\ell}} \right\rceil \qquad (49)$$

for $\ell \in (\hat{\ell}_{m+1}, \hat{\ell}_m]$ with $\ell \in \mathbb{Z}^+$. By using the ceiling function, the values of $\hat{r}_{\ell,\boldsymbol{B}}$ which do not correspond to integer numbers are removed for integers ℓ. This concludes the proof of Theorem 1.

Appendix B
Close-to-optimal Solution of Optimization Problem (9)

In the following, we derive Theorem 3. For the sake of clarity, before stating the subsequent lemma on which Theorem 3 is based on, we introduce some notation. Namely, for given

parameters $N = N'$ (Why do you need N'? It is quite confusing and not clear to the reader!), R_{th} and ℓ of (9), we denote the optimal decision variables and their respective objective value by $\breve{\boldsymbol{r}} = (\breve{r}_{\boldsymbol{A}}, \breve{r}_{\boldsymbol{B}})$ and $\breve{Q}$, respectively. Further, $\breve{R}_{R_{\text{th}},\ell}$ represents the rate attained through the partitioning $\breve{\boldsymbol{r}}$.[8] Similarly, for optimization problem (8), $R^{\star}_{\breve{Q},\ell}$ and $\boldsymbol{r}^{\star} = (r^{\star}_{\boldsymbol{A}}, r^{\star}_{\boldsymbol{B}})$ are respectively, the objective value and the optimal decision variables when $N = \breve{Q}$.

The proposed solution in Theorem 3 uses previous results (20)-(21) of the rate maximization problem (8). Specifically, for parameters $N = \breve{Q}$ and ℓ, $R^{\star}_{\breve{Q},\ell} \approx R_{\text{th}}$ holds which allows us to use the solution in Theorem 1 to derive Theorem 3. The following lemma justifies our approximation $R^{\star}_{\breve{Q},\ell} \approx R_{\text{th}}$.

Lemma 7. For given parameters $N = \breve{Q}$ and ℓ, the optimal solutions of the optimization problems (8) and (9) satisfy the following inequalities:

$$R^{\star}_{\breve{Q},\ell}\left(1 - \frac{1}{r^{\star}_{\boldsymbol{A}}}\right) \leq \breve{R}_{R_{\text{th}},\ell} \leq R^{\star}_{\breve{Q},\ell} \tag{50}$$

$$\breve{R}_{R_{\text{th}},\ell}\left(1 - \frac{1}{\breve{r}_{\boldsymbol{A}}}\right) \leq R_{\text{th}} \leq \breve{R}_{R_{\text{th}},\ell} \tag{51}$$

Proof. First, we consider (50). Let us verify that the following two conflicting cases are not true:

(i) $R^{\star}_{\breve{Q},\ell} < \breve{R}_{R_{\text{th}},\ell}$: The constraints (9b) and (9c) hold for $\breve{\boldsymbol{r}}$ and $N = \breve{Q}$. Thus, the constraints (8a) and (8b) also hold for $\breve{\boldsymbol{r}}$ and $N = \breve{Q}$. Consequently, $\breve{R}_{R_{\text{th}},\ell}$ is a feasible solution of the optimization problem (8). However, $R^{\star}_{\breve{Q},\ell} < \breve{R}_{R_{\text{th}},\ell}$ violates the assumption of optimality of $R^{\star}_{\breve{Q},\ell}$.

(ii) $R^{\star}_{\breve{Q},\ell}\left(1 - \frac{1}{r^{\star}_{\boldsymbol{A}}}\right) > \breve{R}_{R_{\text{th}},\ell}$: The constraints (8a) and (8b) hold for $\boldsymbol{r}^{\star}$ and $N = \breve{Q}$. Therefore the constrains (9a), (9b) and (9c) hold for $\boldsymbol{r}^{\star}$ and every $N \geq \breve{Q}$. From the optimization problem (9) with $N = \breve{Q}$, the following sub-cases have to be considered:

- $Q^{\star} < \breve{Q}$: This sub-case violates the optimality of $\breve{Q}$.
- $Q^{\star} = \breve{Q}$: The pair $(r^{\star}_{\boldsymbol{A}} - 1, r^{\star}_{\boldsymbol{B}})$ is also a solution to the optimization problem (9), since the corresponding rate of this pair is given by

$$\frac{(r^{\star}_{\boldsymbol{A}} - 1)r^{\star}_{\boldsymbol{B}}}{(r^{\star}_{\boldsymbol{A}} - 1 + \ell)(r^{\star}_{\boldsymbol{B}} + 1) - 1} > \frac{(r^{\star}_{\boldsymbol{A}} - 1)r^{\star}_{\boldsymbol{B}}}{(r^{\star}_{\boldsymbol{A}} + \ell)(r^{\star}_{\boldsymbol{B}} + 1) - 1} = R^{\star}_{\breve{Q},\ell}\left(1 - \frac{1}{r^{\star}_{\boldsymbol{A}}}\right) > \breve{R}_{R_{\text{th}},\ell} \geq R_{\text{th}}.$$

  However, this violates the optimality of $\breve{Q}$.

[8] Naturally, due to (9a), $\breve{R}_{R_{\text{th}},\ell} \geq R_{\text{th}}$.

Due to the violation of conditions (i) and (ii), (50) readily follows.

In order to show (51), we follow a similar line of argument as for (50) with the slight difference that we now use the solution $\breve{\boldsymbol{r}}$ of the optimization problem (9). Next, we show that the two following cases (violating (51)) are infeasible:

(i) $\breve{R}_{R_{\text{th}},\ell} < R_{\text{th}}$: This case violates (9a).

(ii) $\breve{R}_{R_{\text{th}},\ell}(1 - \frac{1}{\breve{r}_{\boldsymbol{A}}}) > R_{\text{th}}$: This violates $\breve{Q}$ being minimal, since the pair $(\breve{r}_{\boldsymbol{A}} - 1, \breve{r}_{\boldsymbol{B}})$ would under this condition produce a smaller objective value.

This concludes the lemma. □

Combining (50) and (51), we get

$$R^{\star}_{\breve{Q},\ell}\left(1 - \frac{1}{r^{\star}_{\boldsymbol{A}}}\right)\left(1 - \frac{1}{\breve{r}_{\boldsymbol{A}}}\right) \leq R_{\text{th}} \leq R^{\star}_{\breve{Q},\ell} \tag{52}$$

From (52), we infer the approximation $R_{\text{th}} \approx R^{\star}_{\breve{Q},\ell}$. This approximation is sufficiently accurate for both small and large $r^{\star}_{A}$. For instance, small $r^{\star}_{A}$ are usually associated to large $\ell \approx \ell_{\max}$ where the rate $R^{\star}_{\breve{Q},\ell}$ is close to zero.

The approximation $R_{\text{th}} \approx R^{\star}_{\breve{Q},\ell}$, allows us to utilize the optimization problem (8) for solving (9). In other words, instead of *directly* solving the optimization problem (9) for given parameters N, ℓ and R_{th}, we use the '*detour*' of finding the smallest associated Q for ℓ and $R^{\star}_{\breve{Q},\ell} \approx R_{\text{th}}$ in the optimization problem (8). For this detour, the smallest associated Q for ℓ and $R^{\star}_{\breve{Q},\ell} \approx R_{\text{th}}$ becomes the approximation of $\breve{Q}$. We find the smallest Q by finding the respective $r_{\boldsymbol{A}}$ and $r_{\boldsymbol{B}}$ with ℓ colluding servers and a rate $R^{\star}_{\breve{Q},\ell} \approx R_{\text{th}}$ in (8). We start with $r_{\boldsymbol{B}}$. Hereby, we denote $r_{\boldsymbol{B}}$ with relaxed integer assumption on ℓ by m (for more information we refer the reader to Section A). In the sequel of this section, $(\mathring{})$ is used to indicate approximations. From Eq. (46) we have

$$\hat{d}_m = \frac{2N}{(m-1)m(m+1)}. \tag{53}$$

Recall from Section A that as ℓ decreases from $\ell_{\max}$ to $\ell_{\min}$, m remains constant in certain intervals of ℓ (cf. Fig. 3). The extent to which the rate changes as ℓ decreases is given by the ratio

$$\frac{\Delta\mathring{R}_{\text{th},m}}{\mathring{d}_m} = \frac{\mathring{R}_{\text{th},m} - \mathring{R}_{\text{th},m-1}}{\mathring{d}_m} = \Delta\mathring{R}_{\text{th},m}\Big|_{\mathring{d}_m=1} = \frac{\Delta r_A(m-1)}{\mathring{Q}} = \frac{m-1}{\mathring{Q}}. \tag{54}$$

Please ellaborate. It seems that you assume Q to stay constant. Why should $\Delta r_{\boldsymbol{A}} = 1$? From (53) and (54), we obtain

$$\frac{\Delta \mathring{R}_{\mathrm{th},m}}{\hat{d}_m} = \frac{\Delta \mathring{R}_{\mathrm{th},m}}{\frac{2N}{(m-1)m(m+1)}} = \frac{m-1}{\mathring{Q}}. \tag{55}$$

Considering Eq. (55) and the approximation $N = \mathring{Q}$, we have

$$\Delta \mathring{R}_{\mathrm{th},m} = \frac{2}{m(m+1)}. \tag{56}$$

Summing up over $\Delta \mathring{R}_{\mathrm{th},m}$, we get

$$\mathring{R}_{\mathrm{th},m} = \sum_{i=2}^{m} \Delta \mathring{R}_{\mathrm{th},i} = \sum_{i=2}^{m} \frac{2}{i(i+1)} = \frac{m-1}{m+1}. \tag{57}$$

Recall that for any $R_{\mathrm{th},m} \in [\mathring{R}_{\mathrm{th},m}, \mathring{R}_{\mathrm{th},m+1})$, we know that m is constant and we need to ensure that

$$R_{\mathrm{th},m} \leq \mathring{R}_{\mathrm{th},m+1},$$

or according to (57) equivalently

$$R_{\mathrm{th},m} \leq \frac{m}{m+2} \tag{58}$$

$$\Leftrightarrow m \geq \frac{2R_{\mathrm{th},m}}{1 - R_{\mathrm{th},m}}. \tag{59}$$

Due to the Lemma 1, we choose the smallest m that satisfies (59). This gives us

$$\mathring{r}_{\boldsymbol{B}} = \left\lceil \frac{2}{1 - R_{\mathrm{th}}} - 2 \right\rceil \tag{60}$$

for $R_{\mathrm{th},m} \in [\mathring{R}_{\mathrm{th},m}, \mathring{R}_{\mathrm{th},m+1})$.

References

[1] D. Bowler, T. Miyazaki, and M. Gillan, "Parallel sparse matrix multiplication for linear scaling electronic structure calculations," *Computer Physics Communications*, vol. 137, no. 2, pp. 255 – 273, 2001.

[2] J. Li, A. Skjellum, and R. D. Falgout, "A poly-algorithm for parallel dense matrix multiplication on two-dimensional process grid topologies," *Concurrency: Practice and Experience*, vol. 9, no. 5, pp. 345–389.

[3] D. Clarke, A. Lastovetsky, and V. Rychkov, "Column-based matrix partitioning for parallel matrix multiplication on heterogeneous processors based on functional performance models," in *Parallel Processing Workshops*. Springer Berlin Heidelberg, 2012, pp. 450–459.

[4] S. L. Johnsson, T. Harris, and K. K. Mathur, "Matrix multiplication on the connection machine," in *ACM/IEEE Conference on Supercomputing*, Nov 1989, pp. 326–332.

[5] X. Bultel, R. Ciucanu, M. Giraud, and P. Lafourcade, "Secure matrix multiplication with mapreduce," pp. 1–10, 08 2017.

[6] S. Zhang, H. Li, K. Jia, Y. Dai, and L. Zhao, "Efficient secure outsourcing computation of matrix multiplication in cloud computing," in *GLOBECOM*, Dec 2016, pp. 1–6.

[7] K. M. Khan and M. Shaheen, "Secure cloud services: Matrix multiplication revisited," in *International Conference on Computational Science and Engineering*, Dec 2013, pp. 9–14.

[8] H. A. Nodehi and M. A. Maddah-Ali, "Limited-sharing multi-party computation for massive matrix operations," in *ISIT*, June 2018, pp. 1231–1235.

[9] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *ISIT*, pp. 2022–2026, 2018.

[10] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," *CoRR*, vol. abs/1806.00469, 2018.

[11] J. Kakar and A. Sezgin, "A survey on robust interference management in wireless networks," *Entropy*, vol. 19, no. 7, 2017.

[12] H. Althaus and R. Leake, "Inverse of a finite-field vandermonde matrix (corresp.)," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 173–173, January 1969.

[13] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. R. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *CoRR*, vol. abs/1801.10292, 2018.

[14] Z. Jia, H. Sun, and S. A. Jafar, "Cross subspace alignment and the asymptotic capacity of x-secure t-private information retrieval," *CoRR*, vol. abs/1808.07457, 2018.

[15] H. Yang, W. Shin, and J. Lee, "Private information retrieval for secure distributed storage systems," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 12, pp. 2953–2964, Dec 2018.

[16] K. Kedlaya and C. Umans, "Fast polynomial factorization and modular composition," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1767–1802, 2011.