On the Capacity of Secure Distributed Matrix Multiplication

Wei-Ting Chang   Ravi Tandon
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ, USA
E-mail: {*wchang, tandonr*}@email.arizona.edu

***Abstract*—Matrix multiplication is one of the key operations in various engineering applications. Outsourcing large-scale matrix multiplication tasks to multiple distributed servers or cloud is desirable to speed up computation. However, security becomes an issue when these servers are untrustworthy. In this paper, we study the problem of secure distributed matrix multiplication from distributed untrustworthy servers. This problem falls in the category of secure function computation and has received significant attention in the cryptography community. However, characterizing the fundamental limits of information-theoretically secure matrix multiplication remain an open problem. We focus on information-theoretically secure distributed matrix multiplication with the goal of characterizing the minimum communication overhead. The capacity of secure matrix multiplication is defined as the maximum possible ratio of the desired information and the total communication received from N distributed servers. In particular, we study the following two models where we want to multiply two matrices $A \in \mathbb{F}^{m\times n}$ and $B \in \mathbb{F}^{n\times p}$: (a) one-sided secure matrix multiplication with ℓ colluding servers, in which B is a public matrix available at all servers and A is a private matrix. (b) fully secure matrix multiplication with ℓ colluding servers, in which both A and B are private matrices. The goal is to securely multiply A and B when any ℓ servers can collude. For model (a), we characterize the capacity as $C^{(\ell)}_{\text{one-sided}} = (N-\ell)/N$ by providing a secure matrix multiplication scheme and a matching converse. For model (b), we propose a novel scheme that lower bounds the capacity, i.e., $C^{(\ell)}_{\text{fully}} \geq (\lceil\sqrt{N}-\ell\rceil)^2/(\lceil\sqrt{N}-\ell\rceil+\ell)^2$.**

Keywords – Matrix Multiplication, Security, Secret Sharing.

I. Introduction

In the era of Big Data, performing computationally intensive operations on a local machine becomes challenging and inefficient. Relying on powerful distributed servers is desirable for improving efficiency. As clients, users can upload their data onto servers, and let servers perform computationally expensive tasks for them. However, if the servers are untrustworthy and the data contain sensitive information, it raises security concerns. Therefore, designing algorithms to take advantage of the powerful untrusted servers while keeping them from learning anything about input data is of significant interest.

Cryptography community has looked at this problem under the secure multi-party computation framework, also known as secure function computation. In a secure function computation problem, parties want to jointly compute a function without revealing their respective input to other parties. For example, Alice, who has input x, wants to compute $f(x, y)$ without leaking x to Bob, who has input y, where f is some function they want to compute jointly. Similarly, Bob does not want to reveal y to Alice. Alice and Bob should not learn anything about each other's input from the result of the computation, either. Some previous works include secure two-party computation [1] which proposed using one-way functions to achieve security, and secure multi-party computation [2], [3] to name a few. A class of encryption schemes called Fully Homomorphic Encryption guarantees that any unencrypted items, including the inputs, any intermediate values and the outputs will not be leaked to unintended party. Naturally, it is often used as a solution to secure function computation problems and other types of security problems [4], [5].

Matrix multiplication is a fundamental building block of many science and engineering fields, such as machine learning, image and signal processing, wireless communication, and optimization. In this paper, we focus on the problem of secure distributed matrix multiplication. Secure matrix multiplication is particularly important in scenarios where one wants to train a machine learning model distributedly using gradient descent based algorithms while keeping the training data and the model parameters private. Secure matrix multiplication has been studied in cryptography community, and different approaches have been proposed, including a weaker version of fully homomorphic encryption, namely partially homomorphic encryption [6]–[8].

In contrast to the focus of cryptography community, there are not many works on distributed matrix multiplication with information-theoretic secrecy constraints. There has been significant recent work [9]–[11] in speeding up computation and reducing communication overhead using codes when it comes to distributed matrix multiplication in information theory community. These works speed up matrix multiplication and reduce communication overhead by adding redundancy to the computation using codes. This controlled redundancy introduced by codes allows the distributed system to efficiently tolerate servers who do not respond in a timely manner and mitigate stragglers. Several other recent works that studied secure distributed computing problems that are similar to ours include [12]–[14].

Main Contributions: In this work, we wish to combine the desirable features of works in both communities, and devise

This work was supported by NSF Grants CAREER 1651492, and CNS 1715947.

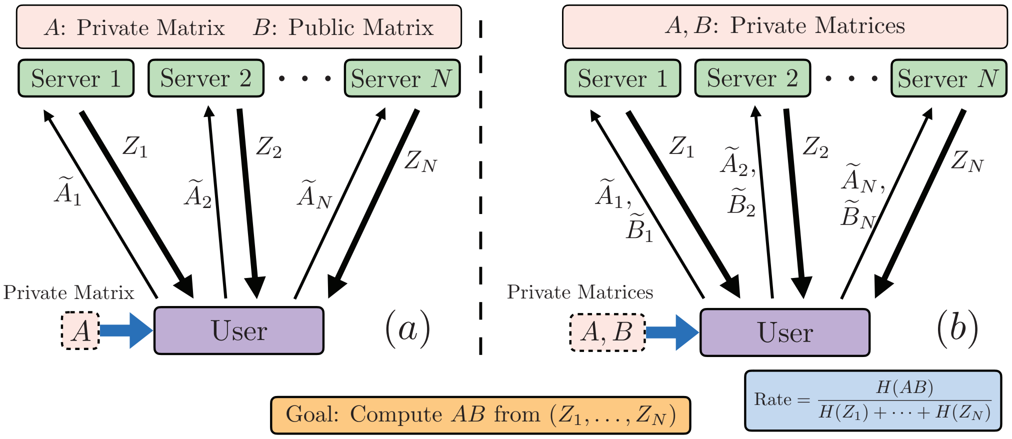
A: Private Matrix B: Public Matrix
Server 1
Server 2
Server N
Z_1
Z_2
Z_N
$\widetilde{A}_1$
$\widetilde{A}_2$
$\widetilde{A}_N$
Private Matrix
A
User
(a)
A, B: Private Matrices
Server 1
Server 2
Server N
Z_1
Z_2
Z_N
$\widetilde{A}_1$, $\widetilde{B}_1$
$\widetilde{A}_2$, $\widetilde{B}_2$
$\widetilde{A}_N$, $\widetilde{B}_N$
Private Matrices
A, B
User
(b)
Goal: Compute AB from $(Z_1, \dots, Z_N)$
Rate $= \frac{H(AB)}{H(Z_1) + \cdots + H(Z_N)}$

Fig. 1: (a) One-sided secure matrix multiplication. (b) Fully secure matrix multiplication.

schemes that are both (a) information-theoretically secure; and (b) have the smallest communication overhead. We consider a system including one user connected to N servers. We assume that servers are honest, but curious. The user wishes to multiply $A \in \mathbb{F}^{m \times n}$ and $B \in \mathbb{F}^{n \times p}$. We consider this problem under two different models.

- We first study the model where B is a public matrix available at all servers, and A is private. The goal is to compute AB securely when any ℓ servers may collude. We devise a capacity achieving scheme based on Shamir's secret sharing scheme [15]. We derive an information-theoretic converse proof and show that the capacity is $(N - \ell)/N$. This result shows that for a scheme to be secured against any ℓ colluding servers, the price we have to pay is ℓ/N.
- We next study the model where both A and B are private matrices with the same goal when any ℓ servers can collude. We devise a novel achievable scheme inspired by the recent works, [9], [10], which show how to leverage codes for distributed matrix multiplication. For this model, our scheme achieves a rate of $(\lceil\sqrt{N} - \ell\rceil)^2/(\lceil\sqrt{N} - \ell\rceil + \ell)^2$. We also show that there is room for improvement and provide an example of the improved scheme using the idea of *aligned secret sharing*.

II. System Model and Problem Formulation

We consider a problem where there are N servers, and a user who wants to compute the product of two input matrices $A \in \mathbb{F}^{m \times n}$ and $B \in \mathbb{F}^{n \times p}$ securely, i.e., AB, using N servers, for some integer m, n and p, and a sufficiently large field $\mathbb{F}$. The user is connected to each server through a private link (Fig. 1) and we assume that the servers are honest, but curious. In order to prevent servers from learning about input matrices, the user sends securely encoded versions of input matrices to servers. We define the encoding functions as $\boldsymbol{f} = (f_1, f_2, \dots, f_N)$ and $\boldsymbol{g} = (g_1, g_2, \dots, g_N)$, where f_i and g_i are the encoding functions for server i for the matrices A and B, respectively. The encoded matrices for server i are denoted by $\widetilde{A}_i$ and $\widetilde{B}_i$ for two input matrices for $i = 1, 2, \dots, N$, i.e., $\widetilde{A}_i = f_i(A)$ and $\widetilde{B}_i = g_i(B)$. We denote the answer from server i as Z_i. From all answers $Z_1, Z_2, \dots, Z_N$, the user must be able to decode the desired result AB, i.e., $AB = d(Z_1, Z_2, \dots, Z_N)$, where $d(.)$ denotes the decoding function. Hence, decodability constraint can be written as,

$$H(AB|Z_1, Z_2, \dots, Z_N) = 0. \tag{1}$$

In this paper, we study the following two models:

(a) *One-Sided Secure Matrix Multiplication with ℓ Colluding Servers:* In this model, B is a public arbitrary constant matrix available at all servers, where A is a private random matrix at the user. Our goal is to securely multiply A and B without revealing anything about A even when any ℓ servers may collude (see Fig. 1(a)), i.e., colluding servers can gather their respective received matrix $\widetilde{A}_i$ and attempt to learn about A. The user does not know which ℓ servers may collude. We use the index set $\mathcal{L} = \{i_1, i_2, \dots, i_\ell\} \subseteq [1 : N], |\mathcal{L}| = \ell$ to denote a subset of ℓ servers, and $\widetilde{A}_{\mathcal{L}} \triangleq (\widetilde{A}_{i_1}, \widetilde{A}_{i_2}, \dots, \widetilde{A}_{i_\ell})$ to denote the corresponding encoded version of A sent to servers in the set $\mathcal{L}$. For a scheme in this setting to be information-theoretically secure, the encoded matrices $\widetilde{A}_{\mathcal{L}}, \forall \mathcal{L} \subseteq [1 : N], |\mathcal{L}| = \ell$ must not leak anything about A. Thus, a scheme for this model must satisfy the following information-theoretic security constraint,

$$I(A; \widetilde{A}_{\mathcal{L}}) = 0, \forall \mathcal{L} \subseteq [1 : N], |\mathcal{L}| = \ell. \tag{2}$$

We say that the rate R is achievable if there exists a scheme satisfying the decodability and security constraints, i.e., (1) and (2). The rate is characterized by the number of desired bits per download bit. The rate is defined as,

$$R = \frac{H(AB)}{\sum\limits_{i=1}^{N} H(Z_i)}. \tag{3}$$

The capacity $C^{(\ell)}_{\text{one-sided}}$ for the one-sided model is the supremum of R over all feasible schemes.

(b) *Fully Secure Matrix Multiplication with ℓ Colluding Servers:* In this model, both A and B are private matrices at the user. Our goal is to multiply them securely when any ℓ servers may collude (see Fig. 1(b)). Hence, encoded matrices $\widetilde{A}_{\mathcal{L}}$ and $\widetilde{B}_{\mathcal{L}}, \forall \mathcal{L} \subseteq [1 : N], |\mathcal{L}| = \ell$ must not reveal anything about A and B. The security constraint for this model is,

$$I(A,B;\widetilde{A}_{\mathcal{L}},\widetilde{B}_{\mathcal{L}})=0,\forall\mathcal{L}\subseteq[1:N],|\mathcal{L}|=\ell. \tag{4}$$

We say that the rate R is achievable if there exits a scheme for which it satisfies both (1) and (4). Similarly, $C_{\text{fully}}^{(\ell)}$ is defined as the supremum of achievable rates for the fully secure matrix multiplication problem. It is clear that $C_{\text{one-sided}}^{(\ell)} \geq C_{\text{fully}}^{(\ell)}$. In the next two sections, we present our main results towards characterizing these capacities.

III. One-Sided Secure Matrix Multiplication with ℓ Colluding Servers

We first study the model where B is public and known at all servers, and the user wants to securely compute AB without revealing A to any ℓ colluding servers. We present our proposed scheme, followed by a converse proof to show that the scheme is information-theoretically optimal.

Theorem 1. *For the (N,ℓ) one-sided secure matrix multiplication problem, in which B is known everywhere and A is kept hidden from any ℓ colluding servers while computing AB, the capacity is given by*

$$C_{\text{one-sided}}^{(\ell)}=\frac{N-\ell}{N}. \tag{5}$$

Before presenting the achievable scheme, we first show an example to highlight the intuition behind the scheme.

Example 1. $(N=4,\ell=2)$ *Consider a one-sided secure matrix multiplication problem with 4 servers, and any 2 of them can collude. The user partitions A into*

$$A=\begin{bmatrix}A_1\\A_2\end{bmatrix}, \tag{6}$$

where $A_1,A_2\in\mathbb{F}^{(m/2)\times n}$. The original matrix multiplication can be rewritten as,

$$AB=\begin{bmatrix}A_1B\\A_2B\end{bmatrix}. \tag{7}$$

The goal is now to compute A_1B and A_2B. The user generates 2 random matrices, i.e., $K_1,K_2\in\mathbb{F}^{(m/2)\times n}$, whose entries are i.i.d. uniform random variables from the field $\mathbb{F}$, and encodes the matrix for server i as,

$$\widetilde{A}_i=A_1+iA_2+i^2K_1+i^3K_2, \tag{8}$$

where each $\widetilde{A}_i$ has the same dimension as A_1 and A_2 for all $i=1,2,3,4$. Server i computes $\widetilde{A}_iB$ and returns the result to the user. The results received at the user are,

$$\begin{aligned}Z_1&=\widetilde{A}_1B=A_1B+A_2B+K_1B+K_2B,\\Z_2&=\widetilde{A}_2B=A_1B+2A_2B+4K_1B+8K_2B,\\Z_3&=\widetilde{A}_3B=A_1B+3A_2B+9K_1B+27K_2B,\\Z_4&=\widetilde{A}_4B=A_1B+4A_2B+16K_1B+64K_2B.\end{aligned} \tag{9}$$

Clearly, the results can be viewed as a system of 4 equations in 4 matrices, and rewritten in matrix form as,

$$\begin{bmatrix}Z_1\\Z_2\\Z_3\\Z_4\end{bmatrix}=\begin{bmatrix}1^0&1^1&1^2&1^3\\2^0&2^1&2^2&2^3\\3^0&3^1&3^2&3^3\\4^0&4^1&4^2&4^3\end{bmatrix}\begin{bmatrix}A_1B\\A_2B\\K_1B\\K_2B\end{bmatrix}. \tag{10}$$

Since the coefficient matrix is a Vandermonde matrix, the system is invertible with a unique solution. The user can multiply the inverse of the coefficient matrix on both sides and solve for A_1B and A_2B. However, for any 2 servers, they see a system of 2 equations in 4 matrices, hence, they will not be able to solve for A_1B and A_2B. The user is able to recover 2 desired items from a total of 4 items, hence, achieving a rate of $1/2$.

Proof of Theorem 1

We next present the general achievable scheme. We show that the capacity can be achieved by a modified Shamir's secret sharing scheme, and we then derive an information-theoretic converse proof to show its optimality.

A. Achievable Scheme

For the achievable scheme, the user first divides A into $N-\ell$ submatrices vertically, i.e.,

$$A=[A_1\ A_2\dots A_{N-\ell}]^T, \tag{11}$$

where $A_i\in\mathbb{F}^{(m/(N-\ell))\times n},\forall i$, and m is divisible by $N-\ell$. Then the matrix multiplication can be written as

$$AB=[A_1B\ \dots\ A_{N-\ell}B]^T. \tag{12}$$

The goal is to recover $A_1B,\dots,A_{N-\ell}B$. The user then encodes the submatrices of A into the following form,

$$\widetilde{A}_i=\sum_{j=1}^{N-\ell}A_jx_i^{j-1}+\sum_{k=1}^{\ell}K_kx_i^{k+(N-\ell)-1}, \tag{13}$$

where the dimension of $\widetilde{A}_i$ is the same as any A_i, and x_i is a distinct non-zero element in $\mathbb{F}$ assigned to server i. Each entry of the random matrices, $K_1,\dots,K_\ell\in\mathbb{F}^{(m/(N-\ell))\times n}$, are i.i.d. uniform random variables from the field $\mathbb{F}$. The encoded matrix $\widetilde{A}_i$ in (13) can be seen as a polynomial evaluated at point x_i. Servers then multiply their received $\widetilde{A}_i$'s with B and return the following polynomial,

$$h(x)=\sum_{j=1}^{N-\ell}A_jBx^{j-1}+\sum_{k=1}^{\ell}K_kBx^{k+(N-\ell)-1}, \tag{14}$$

at $x=x_i,i=1,\dots,N$. Recall that the goal is to recover $A_1B,\dots,A_{N-\ell}B$ from all Z_i, i.e., $h(x_i),i=1,\dots,N$. As shown in the example, due to the design of the scheme, the answers can be seen as a system of N equations in N matrices. Since the coefficient matrix is a Vandermonde matrix, the user can multiply the inverse of the coefficient matrix and solve for the desired items. However, a more efficient decoding method is to view each answer Z_i as a degree $N-1$ polynomial evaluated at point x_i. The coefficients of a degree $N-1$ polynomial can be recovered with N evaluations by polynomial interpolation. Since we can recover $N-\ell$ desired items from N answers, we achieve a rate of $(N-\ell)/N$.

We next prove that the scheme is information-theoretically secure, i.e., the security constraint (2) is satisfied. We start from the following sequence of inequalities:

$$\begin{aligned} I(A;\widetilde{A}_{\mathcal{L}}) &= I(A;\widetilde{A}_{i_1},\dots,\widetilde{A}_{i_\ell}) \\ &= H(\widetilde{A}_{i_1},\dots,\widetilde{A}_{i_\ell}) - H(\widetilde{A}_{i_1},\dots,\widetilde{A}_{i_\ell}|A) \\ &\overset{(a)}{=} H(\widetilde{A}_{i_1},\dots,\widetilde{A}_{i_\ell}) - H(K_1,\dots,K_\ell) \\ &\overset{(b)}{=} H(\widetilde{A}_{i_1},\dots,\widetilde{A}_{i_\ell}) - \ell\frac{mn}{N-\ell}\log|\mathbb{F}| \\ &\overset{(c)}{\leq} H(\widetilde{A}_{i_1})+\dots+H(\widetilde{A}_{i_\ell}) - \ell\frac{mn}{N-\ell}\log|\mathbb{F}| \\ &\overset{(d)}{\leq} \ell\frac{mn}{N-\ell}\log|\mathbb{F}| - \ell\frac{mn}{N-\ell}\log|\mathbb{F}| = 0, \end{aligned} \tag{15}$$

where (a) follows from (13) and the fact that all random matrices K_j's are independent of A, and (b) is due to the entropy of a uniformly distributed random variable being $\log|\mathbb{F}|$ and the dimension of each one of the ℓ random matrices K_j being $mn/(N-\ell)$, (c) follows by upper bounding the joint entropy using the sum of individual entropies and (d) follows from upper bounding the entropy of each element of $\widetilde{A}_{i_{(.)}}$'s by $\log|\mathbb{F}|$. Since mutual information is non-negative and from (15), it is upper bounded by zero, we conclude that the scheme is information-theoretically secure.

B. Converse

We start the converse proof from the following sequence of inequalities:

$$\begin{aligned} H(AB) =\ & H(AB) - H(AB|Z_1,\dots,Z_N) \\ & + \underbrace{H(AB|Z_1,\dots,Z_N)}_{=0} \\ &\overset{(a)}{=} I(AB;Z_1,\dots Z_N) \\ &= H(Z_1,\dots,Z_N) - H(Z_1,\dots,Z_N|AB) \\ &\overset{(b)}{\leq} H(Z_1,\dots,Z_N) - H(Z_{i_1},\dots,Z_{i_\ell}|AB) \\ &\overset{(c)}{=} H(Z_1,\dots,Z_N) - H(Z_{\mathcal{L}}), \end{aligned} \tag{16}$$

where (a) is due to decodability constraint (1), (b) follows by lower bounding the joint entropy of N items using the joint entropy of ℓ items, (c) follows from the Markov Chain $A \rightarrow \widetilde{A}_{\mathcal{L}} \rightarrow Z_{\mathcal{L}}$ and the fact that from data-processing inequality, we know $I(A;\widetilde{A}_{\mathcal{L}}) \geq I(A;Z_{\mathcal{L}})$, which is greater than $I(AB;Z_{\mathcal{L}})$. This along with the secrecy constraint (2), shows that $I(AB;Z_{\mathcal{L}})=0$, hence, we get $H(Z_{\mathcal{L}}|AB) = H(Z_{\mathcal{L}}), \forall \mathcal{L} \subseteq \{1,\dots,N\}, |\mathcal{L}|=\ell$. Since there are $\binom{N}{\ell}$ possible subsets $\mathcal{L}$ of servers of size ℓ, we sum up their entropy and have,

$$\binom{N}{\ell}H(AB) \leq \binom{N}{\ell}H(Z_1,\dots,Z_N) - \sum_{\substack{|\mathcal{L}|=\ell \\ \mathcal{L}\subseteq\{1,\dots,N\}}} H(Z_{\mathcal{L}}). \tag{17}$$

Dividing (17) by $\binom{N}{\ell}$, we have,

$$\begin{aligned} H(AB) \leq\ & H(Z_1,\dots,Z_N) \\ & - \ell\frac{1}{\binom{N}{\ell}} \sum_{\substack{|\mathcal{L}|=\ell \\ \mathcal{L}\subseteq\{1,\dots,N\}}} \frac{H(Z_{\mathcal{L}})}{\ell} \\ &\overset{(a)}{\leq} H(Z_1,\dots,Z_N) - \ell\frac{H(Z_1,\dots,Z_N)}{N} \\ &= \left(1-\frac{\ell}{N}\right)H(Z_1,\dots,Z_N) \\ &\overset{(b)}{\leq} \left(\frac{N-\ell}{N}\right)\sum_{i=1}^{N} H(Z_i), \end{aligned} \tag{18}$$

where in (a) we apply Han's inequality [16, Chapter 17] to bound the second term, and (b) follows by bounding the joint entropy using the sum of entropies. From (18), we get

$$R^{(\ell)}_{\text{one-sided}} = \frac{H(AB)}{\sum_{i=1}^{N} H(Z_i)} \leq \frac{N-\ell}{N}. \tag{19}$$

Hence, from the upper bound in (19) and a matching scheme in Section III-A, we conclude that the capacity for the one-sided matrix multiplication problem is $C^{(\ell)}_{\text{one-sided}} = (N-\ell)/N$. This completes the proof of Theorem 1.

IV. Fully Secure Matrix Multiplication with ℓ Colluding Servers

We next investigate the case where the user wants to compute AB securely while keeping both A and B information-theoretically secure from any ℓ colluding servers. We next present our main result for the fully secure matrix multiplication problem in the following Theorem.

Theorem 2. *For the (N,ℓ) fully secure matrix multiplication problem, in which both A and B must be kept secure from any ℓ colluding servers while computing AB, we have the following lower bound on the capacity:*

$$C^{(\ell)}_{\text{fully}} \geq \frac{(\lceil\sqrt{N}-\ell\rceil)^2}{(\lceil\sqrt{N}-\ell\rceil+\ell)^2}. \tag{20}$$

Before presenting the proposed scheme, we first compare the achievable rate of the proposed fully secure scheme to the capacity of the one-sided secure matrix multiplication problem. Clearly, due to a stronger security requirement, it is clear that the rate of the proposed fully secure scheme to be lower than the capacity of the one-sided secure matrix multiplication problem, when the number of colluding servers ℓ is fixed at a certain value. In Fig. 2, we let $\ell=1$ and increase the number of total servers N. It can be seen that the rate of the the proposed scheme of Theorem 2 is lower, compare to the rate of Theorem 1. Notably, both schemes converge to 1 asymptotically as $N\rightarrow\infty$, however, the convergence for the one-sided case is significantly faster than the convergence for the fully secure case. We also note that for the standard unsecure distributed matrix multiplication, the capacity is 1. We can also see from Fig. 3, that the rate of the proposed

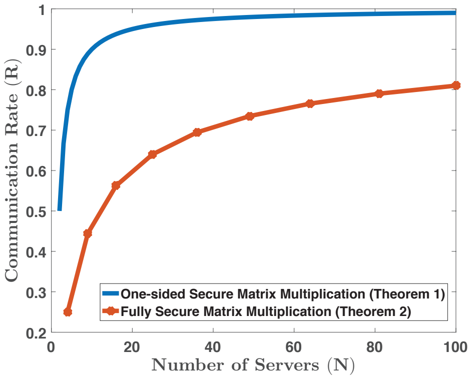

Fig. 2: Comparison between the communication rates of one-sided and fully secure schemes for $\ell = 1$ as N is varied.

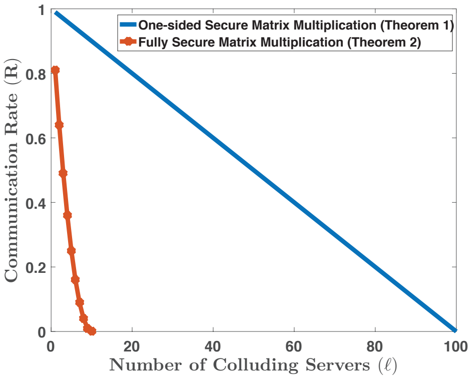

Fig. 3: The impact of number of colluding servers on the communication rate when $N = 100$.

scheme decreases a lot faster than the capacity of the one-sided secure matrix multiplication problem when N is fixed to 100 and ℓ is changing. This indicates that our proposed scheme cannot tolerate too many colluding servers due to the $\sqrt{N}$ term in (20). We next present the proposed scheme in detail.

A. Proof of Theorem 2

For the (N, ℓ) fully secure matrix multiplication problem, the user wishes to compute AB securely without revealing A and B when any ℓ servers may collude. The user breaks the input matrices into r submatrices, where $r = \lceil \sqrt{N} - \ell \rceil$. The reason for choosing this value of r will become clear when we fully describe the scheme next. The submatrices are,

$$A = \begin{bmatrix} A_1 \ A_2 \dots A_r \end{bmatrix}^T \text{ and } B = \begin{bmatrix} B_1 \ B_2 \dots B_r \end{bmatrix}, \tag{21}$$

where $A_i \in \mathbb{F}^{(m/r)\times n}$ and $B_i \in \mathbb{F}^{n\times(p/r)}, \forall i$, and m and p are divisible by r. Hence, we can write AB as,

$$AB = \begin{bmatrix} A_1B_1 & A_1B_2 & \dots & A_1B_r \\ A_2B_1 & A_2B_2 & \dots & A_2B_r \\ \vdots & \vdots & \ddots & \vdots \\ A_rB_1 & A_rB_2 & \dots & A_rB_r \end{bmatrix}, \tag{22}$$

where the original matrix multiplication can be seen as composed of r^2 smaller matrix multiplications.

Similar to one-sided secure matrix multiplication problem, the user generates ℓ random matrices $K_1^{(A)}, \dots, K_\ell^{(A)} \in \mathbb{F}^{(m/r)\times n}$ for A, and ℓ random matrices $K_1^{(B)}, \dots, K_\ell^{(B)} \in \mathbb{F}^{n\times(p/r)}$ for B, where each of their entries is an i.i.d. uniform random variable. The user encodes A and B for server i as:

$$\widetilde{A}_i = \sum_{j=1}^{r} A_j x_i^{j-1} + \sum_{k=1}^{\ell} K_k^{(A)} x_i^{k+r-1}, \tag{23}$$

$$\widetilde{B}_i = \sum_{j=1}^{r} B_j x_i^{(j-1)(r+\ell)} + \sum_{k=1}^{\ell} K_k^{(B)} x_i^{(k+r-1)(r+\ell)}, \tag{24}$$

where $\widetilde{A}_i \in \mathbb{F}^{(m/r)\times n}$ and $\widetilde{B}_i \in \mathbb{F}^{n\times(p/r)}$. The degrees of (23) and (24) are chosen in a way that each item is guaranteed to be the only item at a certain degree after multiplication. This methodology is similar to the one proposed in [9], [10] for distributed matrix multiplication problem. Essentially, computing $\widetilde{A}_i\widetilde{B}_i$ is equivalent to evaluating the following polynomial with 4 different types of terms:

$$\begin{aligned} h(x) = & \underbrace{\sum_{j=1}^{r}\sum_{j'=1}^{r} A_j B_{j'} x^{j+(j'-1)(r+\ell)-1}}_{\text{desired}} \\ & + \sum_{j=1}^{r}\sum_{k'=1}^{\ell} A_j K_{k'}^{(B)} x^{j+(k'+r-1)(r+\ell)-1} \\ & + \sum_{k=1}^{\ell}\sum_{j'=1}^{r} K_k^{(A)} B_{j'} x^{k+r+(j'-1)(r+\ell)-1} \\ & + \sum_{k=1}^{\ell}\sum_{k'=1}^{\ell} K_k^{(A)} K_{k'}^{(B)} x^{k+r+(k'+r-1)(r+\ell)-1}. \end{aligned} \tag{25}$$

Due to the design of the scheme, each degree has exactly one item as its coefficient in (25). Note that the polynomial has degree $(r+\ell)^2-1$, hence, evaluations at $(r+\ell)^2$ distinct points are sufficient to solve for all coefficients of the polynomial. This indicates that we need at least $(r+\ell)^2$ responses, one from each server to recover the desired result, i.e., $N \geq (r+\ell)^2$. However, the user is only interested in the first double summation term in (25), which has a total of r^2 items in the form of $A_jB_{j'}$. Since the user can recover r^2 items out of $(r+\ell)^2$ items, the achievable scheme yields a rate of $r^2/(r+\ell)^2 = (\lceil\sqrt{N}-\ell\rceil)^2/(\lceil\sqrt{N}-\ell\rceil+\ell)^2$.

We next show that the proposed scheme is information-theoretically secure:

$$\begin{aligned} & I(A,B;\widetilde{A}_\mathcal{L},\widetilde{B}_\mathcal{L}) = I(A,B;\widetilde{A}_\mathcal{L}) + I(A,B;\widetilde{B}_\mathcal{L}|\widetilde{A}_\mathcal{L}) \\ = \ & H(\widetilde{A}_\mathcal{L}) - H(\widetilde{A}_\mathcal{L}|A,B) \\ & \qquad + H(\widetilde{B}_\mathcal{L}|\widetilde{A}_\mathcal{L}) - H(\widetilde{B}_\mathcal{L}|\widetilde{A}_\mathcal{L},A,B) \\ \overset{(a)}{=} \ & H(\widetilde{A}_\mathcal{L}) - H(K_1^{(A)},\dots,K_\ell^{(A)}) \\ & \qquad + H(\widetilde{B}_\mathcal{L}) - H(K_1^{(B)},\dots,K_\ell^{(B)}) \\ \overset{(b)}{=} \ & H(\widetilde{A}_\mathcal{L}) - \ell\frac{mn}{r}\log|\mathbb{F}| + H(\widetilde{B}_\mathcal{L}) - \ell\frac{np}{r}\log|\mathbb{F}| \\ \overset{(c)}{\leq} \ & H(\widetilde{A}_{i_1}) + \dots + H(\widetilde{A}_{i_\ell}) - \ell\frac{mn}{r}\log|\mathbb{F}| \\ & \qquad + H(\widetilde{B}_{i_1}) + \dots + H(\widetilde{B}_{i_\ell}) - \ell\frac{np}{r}\log|\mathbb{F}| \\ \overset{(d)}{\leq} \ & \ell\frac{mn}{r}(\log|\mathbb{F}| - \log|\mathbb{F}|) + \ell\frac{np}{r}(\log|\mathbb{F}| - \log|\mathbb{F}|) = 0, \end{aligned} \tag{26}$$

where (a) follows from (23), (24) and the fact that random matrices are independent of A and B, and $\widetilde{B}_{\mathcal{L}}$ is independent of $\widetilde{A}_{\mathcal{L}}$, (b) follows by summing the entropy of each uniformly distributed random variable in all $K_j^{(A)}$ and $K_{j'}^{(B)}$, (c) follows by upper bounding the joint entropy using the sum of individual entropies, (d) follows from upper bounding the entropy of each element of $\widetilde{A}_{i_{(.)}}$ and $\widetilde{B}_{i_{(.)}}$ by $\log|\mathbb{F}|$. Hence, the proposed scheme is information-theoretically secure. This completes the proof of Theorem 2.

B. Improving Theorem 2 by Aligned Secret Sharing

Due to the design of our proposed scheme, each item is the coefficient of a distinct degree. However, in a fully secure matrix multiplication problem, only items with the form of $A_jB_{j'}$ are useful. Hence, if we can ensure that each item with the form of $A_jB_{j'}$ is the only coefficient of some distinct degrees while *aligning* the other undesired items, we can potentially achieve a better rate. We present the following example to demonstrate the idea of *aligned secret sharing*.

Example 2. *Consider the* $(N,\ell) = (8,1)$ *fully secure matrix multiplication problem where there are* 8 *servers, and none of them collude. For this example, from Theorem 2, we can achieve a rate of* $(\lceil\sqrt{N-\ell}\rceil)^2/(\lceil\sqrt{N-\ell}\rceil+\ell)^2 = 2^2/(2+1)^2 = 4/9$. *We now show how to improve upon this rate through the aligned secret sharing scheme.*

The user partitions A *and* B *into the following*

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \text{ and } B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \tag{27}$$

where $A_1, A_2 \in \mathbb{F}^{(m/2)\times n}$, *and* $B_1, B_2 \in \mathbb{F}^{n\times(p/2)}$. *The user generates one random matrix for each* A *and* B, *i.e.,* $K^{(A)} \in \mathbb{F}^{(m/2)\times n}$ *and* $K^{(B)} \in \mathbb{F}^{n\times(p/2)}$. *Instead of following the proposed scheme in Section IV-A, we align the undesired terms in the forms of* $A_jK^{(B)}, K^{(A)}B_{j'}$ *and* $K^{(A)}K^{(B)}$ *by selecting different degrees for the encoding polynomial. For each server, the encoding of the user is:*

$$\widetilde{A}_i = A_1 + A_2x_i + K^{(A)}x_i^2 \tag{28}$$

$$\widetilde{B}_i = B_1 + B_2x_i^3 + K^{(B)}x_i^5, \tag{29}$$

where $\widetilde{A}_i$ *and* $\widetilde{B}_i$ *have the same dimension as* A_i *and* B_i *for* $i = 1,\ldots,8$. *Each server* i *evaluates the polynomial*

$$h(x_i) = A_1B_1 + A_2B_1x_i + K^{(A)}B_1x_i^2 + A_1B_2x_i^3 + A_2B_2x_i^4 + (K^{(A)}B_2 + A_1K^{(B)})x_i^5 + A_2K^{(B)}x_i^6 + K^{(A)}K^{(B)}x_i^7, \tag{30}$$

for $i = 1,\ldots,8$. *Clearly, the desired items are the only coefficients of their respective degrees, consequently, the user can decode them using polynomial interpolation. Since the degree of the polynomial is now* 7, *evaluation at* 8 *points are sufficient and there are* 4 *desired items. The rate is now* $4/8 = 1/2$ *which is larger than* $4/9$.

V. Conclusions

In this paper, we studied one-sided and fully secure matrix multiplication problems. We proposed a secret sharing based scheme for the one-sided secure matrix multiplication model, where B is a public matrix and A is a private matrix that must be kept information-theoretically secure while computing AB when any ℓ servers may collude. We completely characterized the capacity for this model as $(N-\ell)/N$. We also presented a novel achievable scheme for the fully secure matrix multiplication model, where both A and B are private matrices that must be kept information-theoretically secure against any ℓ colluding servers. We also presented an improvement for this general scheme through the idea of aligned secret sharing. There are several interesting open problems: (a) finding a converse (upper bound) for the fully secure matrix multiplication problem; and (b) generalizing these ideas for other secure distributed computation tasks.

References

[1] A. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science*, Nov. 1982, pp. 160–164.

[2] D. Chaum, I. B. Damgård, and J. van de Graaf, "Multiparty computations ensuring privacy of each party's input and correctness of the result," in *Advances in Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 87–119.

[3] M. Jakobsson and A. Juels, "Mix and match: Secure function evaluation via ciphertexts," in *Advances in Cryptology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 162–177.

[4] S. Rane, W. Sun, and A. Vetro, "Secure function evaluation based on secret sharing and homomorphic encryption," in *47th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2009, pp. 827–834.

[5] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 24–43.

[6] K. M. Khan and M. Shaheen, "Secure cloud services: Matrix multiplication revisited," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, Dec. 2013, pp. 9–14.

[7] X. Bultel, R. Ciucanu, M. Giraud, and P. Lafourcade, "Secure Matrix Multiplication with MapReduce," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*. New York, NY, USA: ACM, 2017, pp. 11:1–11:10.

[8] D. H. Duong, P. K. Mishra, and M. Yasuda, "Efficient secure matrix multiplication over LWE-based homomorphic encryption," *Tatra Mountains Mathematical Publications*, vol. 67, no. 1, pp. 69–83, 2016.

[9] Q. Yu, M. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," in *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017, pp. 4403–4413.

[10] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *CoRR*, vol. abs/1801.07487, 2018. [Online]. Available: http://arxiv.org/abs/1801.07487

[11] 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. [Online]. Available: http://arxiv.org/abs/1801.10292

[12] R. Bitar, P. Parag, and S. E. Rouayheb, "Minimizing latency for secure distributed computing," in *2017 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2017, pp. 2900–2904.

[13] Q. Yu, N. Raviv, J. So, and A. S. Avestimehr, "Lagrange Coded Computing: Optimal Design for Resiliency, Security and Privacy," *CoRR*, vol. abs/1806.00939, 2018. [Online]. Available: http://arxiv.org/abs/1806.00939

[14] H. A. Nodehi and M. A. Maddah-Ali, "Limited-sharing multi-party computation for massive matrix operations," in *2018 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 1231–1235.

[15] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.