Degree Tables for Secure Distributed Matrix Multiplication

Rafael G.L. D'Oliveira, Salim El Rouayheb, Daniel Heinlein, David Karpuk
ECE, Rutgers University, USA
Department of Communications and Networking, Aalto University, Finland
Departamento de Matemáticas, Universidad de los Andes, Colombia
Emails: {rafael.doliveira, salim.elrouayheb}@rutgers.edu, daniel.heinlein@aalto.fi, da.karpuk@uniandes.edu.co

***Abstract*—We consider the problem of secure distributed matrix multiplication (SDMM) in which a user wishes to compute the product of two matrices with the assistance of honest but curious servers. We construct polynomial codes for SDMM by studying a recently introduced combinatorial tool called the degree table. Maximizing the download rate of a polynomial code for SDMM is equivalent to minimizing N, the number of distinct elements in the corresponding degree table. We propose new constructions of degree tables with a low number of distinct elements. These new constructions lead to a general family of polynomial codes for SDMM, which we call GASP_r (Gap Additive Secure Polynomial codes) parametrized by an integer r. GASP_r outperforms all previously known polynomial codes for SDMM. We also present lower bounds on N and show that GASP_r achieves the lower bounds in the case of no server collusion.**

***Index Terms*—Secure distributed matrix multiplication, polynomial codes, degree table, additive combinatorics, sumsets.**

I. Introduction

We consider the problem of secure distributed matrix multiplication (SDMM): A user has two matrices, A and B, and wishes to compute their product, AB, with the assistance of N servers, without leaking any information about either A or B to any server. We assume that all servers are honest but curious, i.e., any T of them may collude to try to deduce information about either A or B.

The primary performance metric used in the literature to compare different schemes for SDMM is the *download rate*, which we denote by $\mathcal{R}$. This rate $\mathcal{R}$ is defined as the ratio of the amount of information about AB (in bits) the user downloads from the servers to the total number of downloaded bits. The goal is to construct an SDMM scheme with rate $\mathcal{R}$ as large as possible, given some limit on the number of servers or on their computational power.

The main technique used for constructing polynomial codes for SDMM can be summarized as follows. We partition the matrices A and B:

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_K \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & \cdots & B_L \end{bmatrix}, \\ \text{so that} \quad AB = \begin{bmatrix} A_1B_1 & \cdots & A_1B_L \\ \vdots & \ddots & \vdots \\ A_KB_1 & \cdots & A_KB_L \end{bmatrix}, \tag{1}$$

making sure that all products A_kB_ℓ are well-defined and of the same size. Computing the product AB is equivalent to computing all subproducts A_kB_ℓ. One then constructs a polynomial $h(x) = f(x) \cdot g(x)$ whose coefficients encode the submatrices A_kB_ℓ, and utilizes N servers to compute the evaluations $h(a_1), \ldots, h(a_N)$ for certain $a_1, \ldots, a_N$. The polynomial h is constructed so that every T-subset of evaluations reveals no information about A or B (T-security), but so that the user can reconstruct all of AB given all N evaluations (decodability).

The partition parameters K and L are inversely proportional to the amount of computation that each of the servers will have to perform. Mathematically, it is convenient to think of the number of servers, N, as a function of K, L, and the security parameter T. In this way, maximizing the download rate $\mathcal{R}$, and even the upload rate, is equivalent to minimizing N as a function of K, L, and T. Consider polynomials of the following type:

$$f(x) = \sum_{k=1}^{K} A_k x^{\alpha_k} + \sum_{t=1}^{T} R_t x^{\alpha_{K+t}} \\ g(x) = \sum_{\ell=1}^{L} B_\ell x^{\beta_\ell} + \sum_{t=1}^{T} S_t x^{\beta_{L+t}} \tag{2}$$

The R_t and S_t are random matrices used to guarantee privacy. The exponents of the terms in $h(x) = f(x) \cdot g(x)$ will be given by the sum of the exponents, denoted by the vectors α and β, in $f(x)$ and $g(x)$.

The degree table was first introduced in [1]. The degree table of $h(x)$, depicted in Table I, shows the exponents in $h(x)$ as a function of α and β. In Theorem 1 of [1], it is shown that if the degree table satisfies the following conditions: (i) the numbers in the red block are unique in the table and; and (ii) numbers in the green/blue block are pairwise distinct, then there exists evaluation points such that the polynomial code in Equation (2) is decodable and T-secure. More so, the number of servers, N, is the number of distinct terms in the table. Thus, the main question we are interested in is how to choose the degree table, i.e., α and β, to minimize the number of servers, N.

A. Related Work

One distinguishing factor of the SDMM problem is that both matrices, A and B, must be kept secure. In the case where

978-1-5386-6900-6/19/$31.00 ©2019 IEEE

	β_1	$\cdots$	β_L	β_{L+1}	$\cdots$	β_{L+T}
α_1	$\alpha_1+\beta_1$	$\cdots$	$\alpha_1+\beta_L$	$\alpha_1+\beta_{L+1}$	$\cdots$	$\alpha_1+\beta_{L+T}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
α_K	$\alpha_K+\beta_1$	$\cdots$	$\alpha_K+\beta_L$	$\alpha_K+\beta_{L+1}$	$\cdots$	$\alpha_K+\beta_{L+T}$
α_{K+1}	$\alpha_{K+1}+\beta_1$	$\cdots$	$\alpha_{K+1}+\beta_L$	$\alpha_{K+1}+\beta_{L+1}$	$\cdots$	$\alpha_{K+1}+\beta_{L+T}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
α_{K+T}	$\alpha_{K+T}+\beta_1$	$\cdots$	$\alpha_{K+T}+\beta_L$	$\alpha_{K+T}+\beta_{L+1}$	$\cdots$	$\alpha_{K+T}+\beta_{L+T}$

Table I: The Degree Table. The α_i's and β_i's are the exponents of the polynomials $f(x)$ and $g(x)$ in (2) used to encode A and B, respectively. The table entries are the monomial degrees in $f(x) \cdot g(x)$. The problem is to choose the degrees α_i's and β_i's to minimize the number of distinct entries in the table subject to: (i) Decodability, the numbers in the red block must be distinct to all the other ones; (ii) T-security, all numbers in the green/blue block must be pairwise distinct.

only one of the matrices must be kept secure, one can use methods like Shamir's secret sharing [2], Staircase codes [3], or Lagrange Coded Computing [4].

For distributed computations, polynomial codes were originally introduced in [5] in a slightly different setting, namely to mitigate stragglers in distributed matrix multiplication. This was followed by a series of works, [6]–[9].

In [10], a similar setting is studied with two major differences. Workers can communicate with each other and an extra security constraint, where a third party wants the final result of the computation and should not learn anything about the inputs.

Our setting was first considered in [11] for $K = L$, in which a polynomial scheme with $N = (K + T)^2$ was presented. In [12], this was improved to $N = (K + T)(L + 1) - 1$. The same N was obtained in [13] for $T = 1$.

The degree table was introduced in [1] together with two schemes, $\mathsf{GASP}_{\text{big}}$ and $\mathsf{GASP}_{\text{small}}$. We omit restating the formulas for N given in [1], since $\mathsf{GASP}_{\text{big}}$ is GASP_r with $r = \min\{K, T\}$ and $\mathsf{GASP}_{\text{small}}$ is GASP_r with $r = 1$ using the newly introduced common generalization called GASP_r in Definition 1 and we give a new formula to compute the N value of GASP_r in Theorem 1.

The study of the degree table is related to the topic of sumsets in additive combinatorics [14], [15]. In that context, a problem, usually referred to as the *inverse problem* is to obtain structural information on two finite sets, A and B, given that the cardinality of $A + B = \{a + b : a \in A, b \in B\}$ is small.

In our case, the decodability condition implies that at least KL integers appear only with multiplicity one in the sumset and therefore classical theorems about sumsets yield only meaningful results if KL is small.

B. Summary of the paper

The rest of the paper is organized as follows.

- In Section II, we summarize the highlights of this paper:
  - We present a family of SDMM schemes, called GASP_r, which generalizes the best schemes previously known and outperforms them for many parameters.
  - We give a formula for the number of servers, N, in GASP_r.
  - We give lower bounds on N in general and show that GASP_r is asymptotically optimal for certain parameters, namely when $K = L = T = n^2 \geq 4$.
- In Section III, we demonstrate our polynomial code GASP_r via an explicit example, in order to show the subtleties of the construction.
- In Section IV, we show the idea of the proof ultimately leading to a formula of the N parameter of GASP_r and for certain parameters, we prove the optimal choice of r.
- In Section V, we give lower bounds on the constructions using the degree table and show the optimality of GASP_r for certain parameters.

II. Main Results

We start by introducing our main contribution, GASP_r codes.

Definition 1. Given the partitioning parameters, K and L, the security parameter T, and $1 \leq r \leq \min\{K, T\}$, we define the polynomial code GASP_r as the polynomials in Equation 2 with exponents α and β given as

- $\alpha = (0, 1, \ldots, K-1, KL, KL+1, \ldots, KL+r-1, KL+K, KL+K+1, \ldots, KL+r-1, \ldots)$ of length $K + T$,
- $\beta = (0, K, \ldots, K(L-1), KL, KL+1, \ldots, KL+T-1)$,

if $L \leq K$. If $K < L$ we just interchange the roles of K and L in the definition.

We call the parameter r the chain length.

In the remainder of this work, we assume, without loss of generality, that $L \leq K$. In the case where $K < L$ one needs only to interchange the roles of K and L in all the expressions.

The following example will make Definition 1 clearer.

Example 1. For $K = L = T = 4$ we have four GASP_r codes, all of which have the same $\beta = (0, 4, 8, 12, 16, 17, 18, 19)$.

- For $r = 1$: $\alpha = (0, 1, 2, 3, 16, 20, 24, 28)$.
- For $r = 2$: $\alpha = (0, 1, 2, 3, 16, 17, 20, 21)$.
- For $r = 3$: $\alpha = (0, 1, 2, 3, 16, 17, 18, 20)$.
- For $r = 4$: $\alpha = (0, 1, 2, 3, 16, 17, 18, 19)$.

This family of codes are a generalization of the codes $\mathsf{GASP}_{\text{small}} = \mathsf{GASP}_1$ and $\mathsf{GASP}_{\text{big}} = \mathsf{GASP}_{\min\{K,T\}}$ presented in [1].

We are interested in finding the best chain length r, i.e., the one which minimizes the number of servers needed, for any given parameters.

Definition 2. Let K and L be the partitioning parameters, T be the security parameter, and $N(r)$, the number of distinct terms in the degree table constructed by GASP_r. The optimal chain length is defined as

$$r^* = \underset{r \in \{1,2,\ldots,\min\{K,T\}\}}{\arg\min} N(r).$$

In Theorem 1 we show how to calculate the number of distinct terms in the degree table, i.e., the number of servers

needed for the scheme, for GASP_r. Due to space constraints, we relegate the proof of this theorem to [16].

Theorem 1. *Let K and L be the partitioning parameters, T be the security parameter, and r be the chain length. Then, the degree table constructed by GASP_r has the number of terms given by $N = KL + K + T - 1 + T(L+T) - S$, where*

$$\begin{aligned} S = {} & \max\{0, \min\{r, \varphi\}\}L + 2\max\{0, r - z + 1\} + \gamma \\ & + (T - r)L + \max\{0, K + T - KL - 1\} \\ & + \eta \max\{0, T - K + r - 1\} + (T - 1 - \eta)(T - 1), \end{aligned} \tag{3}$$

$$\varphi = T - 1 - KL + 2K, \eta = \left\lfloor \frac{T-1}{r} \right\rfloor, z = \max\{1, \varphi + 1\},$$

$$\gamma = \begin{cases} 0 & \text{if } r < z \\ K(x-a)(x+a-1)/2 - ab + xy + x & \text{else} \end{cases}$$

with a, b, x, y defined by

$$T - 1 - r = aK + b \text{ and } 0 \le b \le K - 1,$$
$$T - 1 - z = xK + y \text{ and } 0 \le y \le K - 1.$$

The key to proving Theorem 1 is to determine the parameter S, called the score of the chain. In Section III, we give insights, using an example, on what S represents in the degree table, and in Section IV, we show how to compute it.

Theorem 1 allows us to infer the optimal chain length, r^*, by calculating S for every $1 \le r \le \min\{K, T\}$.

Under some conditions we are able to give a simple expression for the optimal chain length. A particularly revealing special case of this is presented in Corollary 1.

Corollary 1. In the setting of Theorem 1, if $K = L = T = n^2$ for $1 \le n$, then $r^* = n$. Hence, for $n = 1$, $N = 3$ and for $n \ge 2$, we have $N = n^4 + 2n^3 + 2n^2 - n - 2$.

In Theorem 2 we give three lower bounds for the number of distinct terms, N, of any degree table.

Theorem 2. *Let K and L be the partitioning parameters, T be the security parameter, α and β be vectors such that the degree table in Table I is decodable and T-secure, and N be the number of distinct terms in this degree table. Then the following three inequalities hold.*

1) $KL + \max\{K, L\} + 2T - 1 \le N$.
2) *If* $3\max\{K, L\} + 3T - 2 < KL$ *or* $2 \le K = L$, *then* $KL + \max\{K, L\} + 2T \le N$.
3) $KL + K + L + 2T - 1 - T\min\{K, L, T\} \le N$.

We note that Inequality 3 in Theorem 2 is stronger than Inequality 1 if and only if $T^2 < \min\{K, L\}$. Inequality 2 is always stronger than Inequality 1 by one if its condition is met and hence Inequality 3 is stronger than Inequality 2 if its condition is met if and only if $T^2 + 1 < \min\{K, L\}$.

By comparing the bounds in Theorem 2 to the number of distinct terms in GASP_r, counted via Theorem 1, we show in Corollaries 4 and 5 that GASP_{r^*} is optimal whenever one of the three parameters is one, $K = 1$, $L = 1$ or $T = 1$.

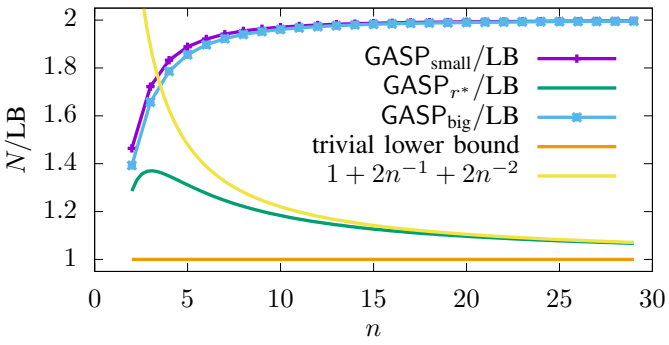

Figure 1: Comparison of $\mathsf{GASP}_{\text{small}} = \mathsf{GASP}_1$, GASP_{r^*} with $r^* = n$ due to Corollary 1, and $\mathsf{GASP}_{\text{big}} = \mathsf{GASP}_{\min\{K,T\}}$ in the setting of $K = L = T = n^2$. The term "LB" refers to the left hand side of Inequality 2 in Theorem 2.

In the setting of Corollary 1, we can show that GASP_{r^*} is asymptotically optimal.

Corollary 2. In the setting of Corollary 1, if $K = L = T = n^2 \ge 4$, then the following lower bound on N holds

$$N \ge n^4 + 3n^2. \tag{4}$$

Moreover, GASP_n is asymptotically optimal and within 38% of the lower bound.

Proof. Inequality (2) in Theorem 2 is $KL + \max\{K, L\} + 2T = n^4 + 3n^2$ and the fraction of the size of the degree table constructed by GASP_n divided by the left hand side of Inequality (2) in Theorem 2 is

$$\begin{aligned} & \frac{n^4 + 2n^3 + 2n^2 - n - 2}{n^4 + 3n^2} \le \frac{n^4 + 2n^3 + 2n^2}{n^4} \\ & = 1 + 2n^{-1} + 2n^{-2} \in 1 + \Theta(n^{-1}), \end{aligned} \tag{5}$$

i.e., the left hand side is asymptotically optimal and its maximum is < 1.38 by $n \approx 3$. □

In Figure 1, we draw $\mathsf{GASP}_{\text{small}}$, $\mathsf{GASP}_{\text{medium}}$, and $\mathsf{GASP}_{\text{big}}$. The graphs are normalized to the left hand side of Inequality (2) in Theorem 2 and we draw the right hand side of Inequality 5.

Remark 1. The lower bounds presented here are with respect to the degree table construction. Thus, when we say a construction is optimal, we mean with respect to the degree table.

III. A Motivating Example: $K = L = T = 4$

In this example we consider the multiplication of two matrices A and B over a finite field $\mathbb{F}_q$, partitioned as:

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \end{bmatrix}$$

	0	4	8	12	16	17	18	19
0	0	4	8	12	16	17	18	19
1	1	5	9	13	17	18	19	20
2	2	6	10	14	18	19	20	21
3	3	7	11	15	19	20	21	22
16	16	20	24	28	32	33	34	35
20	20	24	28	32	36	37	38	39
24	24	28	32	36	40	41	42	43
28	28	32	36	40	44	45	46	47

(a) $r=1$, $S=14$, $N=41$

	0	4	8	12	16	17	18	19
0	0	4	8	12	16	17	18	19
1	1	5	9	13	17	18	19	20
2	2	6	10	14	18	19	20	21
3	3	7	11	15	19	20	21	22
16	16	20	24	28	32	33	34	35
17	17	21	25	29	33	34	35	36
20	20	24	28	32	36	37	38	39
21	21	25	29	33	37	38	39	40

(b) $r=2$, $S=19$, $N=36$

	0	4	8	12	16	17	18	19
0	0	4	8	12	16	17	18	19
1	1	5	9	13	17	18	19	20
2	2	6	10	14	18	19	20	21
3	3	7	11	15	19	20	21	22
16	16	20	24	28	32	33	34	35
17	17	21	25	29	33	34	35	36
18	18	22	26	30	34	35	36	37
20	20	24	28	32	36	37	38	39

(c) $r=3$, $S=18$, $N=37$

	0	4	8	12	16	17	18	19
0	0	4	8	12	16	17	18	19
1	1	5	9	13	17	18	19	20
2	2	6	10	14	18	19	20	21
3	3	7	11	15	19	20	21	22
16	16	20	24	28	32	33	34	35
17	17	21	25	29	33	34	35	36
18	18	22	26	30	34	35	36	37
19	19	23	27	31	35	36	37	38

(d) $r=4$, $S=16$, $N=39$

Table II: The degree tables for GASP_r, for all r, in the setting where $K = L = T = 2^2$. As per Corollary 1, $r^* = 2$ achieves $N = 36$. The lower bound for these tables, given in (4), is $N \geq 28$. The gray region in the lower half of the degree table consists of the terms which have already appeared before. Their quantity is precisely the score, S, appearing in Theorem 1.

so that all the products $A_k B_\ell$ are of the same size. The product AB is given by

$$AB = \begin{bmatrix} A_1B_1 & A_1B_2 & A_1B_3 & A_1B_4 \\ A_2B_1 & A_2B_2 & A_2B_3 & A_2B_4 \\ A_3B_1 & A_3B_2 & A_3B_3 & A_3B_4 \\ A_4B_1 & A_4B_2 & A_4B_3 & A_4B_4 \end{bmatrix}$$

We construct a scheme which computes each term $A_k B_\ell$, and therefore all of AB via polynomial interpolation. The scheme must be private for any $T = 4$ servers colluding to infer any information about the A or B.

Let $R_1, \ldots, R_4$ and $S_1, \ldots, S_4$ be matrices picked independently and uniformly at random with entries in $\mathbb{F}_q$, of sizes equal to the A_k and B_ℓ, respectively. Define the polynomials

$$f(x) = A_1 x^{\alpha_1} + A_2 x^{\alpha_2} + A_3 x^{\alpha_3} + A_3 x^{\alpha_4} + R_1 x^{\alpha_5} + R_2 x^{\alpha_6} + R_3 x^{\alpha_7} + R_4 x^{\alpha_8}$$

$$g(x) = B_1 x^{\beta_1} + B_2 x^{\beta_2} + B_3 x^{\beta_3} + B_4 x^{\beta_4} + S_1 x^{\beta_5} + S_2 x^{\beta_6} + S_3 x^{\beta_7} + S_4 x^{\beta_8}$$

We recover the products $A_k B_\ell$ by interpolating the product $h(x) = f(x)g(x)$. Specifically, for some evaluation points $a_n \in \mathbb{F}_q$, we send $f(a_n)$ and $g(a_n)$ to server $n = 1, \ldots, N$, who then responds with $h(a_n) = f(a_n)g(a_n)$. These evaluations suffice to interpolate all of $h(x)$. In particular, we are able to retrieve the coefficients of $h(x)$, which in turn will allow us to decode all the $A_k B_\ell$.

In [1], it was shown that if the degree table of α and β satisfy the conditions in Table I, then the number of evaluation points needed, N, is equal to the number of distinct terms in the degree table.

In Table II, we show the degree tables of GASP_r for all r. The upper half of the degree table coincides for every r, and consists of the numbers from 0 to $KL + K + T - 2 = 22$.

The gray region in the lower half of the degree table consists of the terms which have already appeared before, by ordering them up to down. The number of terms in the gray region is precisely the score, S, appearing in Theorem 1.

We calculate the number of distinct terms in the degree table as follows. As seen previously, the upper half of the degree table has $KL + K + T - 1 = 23$ distinct terms. The lower half has a total of $T(L + T) = 32$ terms, S of which appear elsewhere. Thus $N = KL+K+T-1+T(L+T)-S = 55-S$.

In Theorem 1 we show how to compute the score, S, for any r. In general, we can determine the best chain length, r^*, by computing all $\min\{K, T\} = 4$ possibilities for r and choosing the one which maximizes the score, S. In this case $r^* = 2$ which could have also been obtained directly through Corollary 1. Thus, for this case, GASP_2 is the best known scheme requiring $N = 36$ servers.

Using the best lower bound for this case in Theorem 2, we obtain $N \geq KL + \max\{K, L\} + 2T = 28$.

IV. GASP_r

A. The Number of Distinct Terms in GASP_r

In this section we will show the key ingredient for proving Theorem 1, the computation of the score of, S, of the chain.

Definition 3. Let K and L be the partitioning parameters, T be the security parameter, and r be the chain length of the code GASP_r.

For $1 \leq i \leq T$ we define L_i (and R_i) to be the set of integers that are in the first L (last T) entries of row $K + i$ such that these integers appear in the first $K + i - 1$ rows of the degree table constructed by GASP_r.

We call the cardinalities, $|L_i|$ and $|R_i|$, the left and, respectively, right score of the row $K + i$.

In Table II, the left scores are represented by the gray regions in the lower left side of the degree table, and the right scores by the gray regions in the lower right side.

Definition 4. In the setting of Definition 3, we define the score of row i as $S_i = |L_i| + |R_i|$ and the score of the chain r as $S = S_1 + \ldots + S_T$.

By the arguments in Section III, we showed that $N = KL + K + T - 1 + T(L + T) - S$. Thus, determining N is a matter of determining S which is a function of left and right scores. In Lemma 1 we show how to find these scores.

Lemma 1. *In the setting of Definition 3, it follows that*

$$|L_i| = \begin{cases} \min\{L, 2 + \lfloor (T - 1 - i)/K \rfloor\} & \text{if } 1 \leq i \leq r \\ L & \text{if } r + 1 \leq i \leq T \end{cases}$$

and

$$|R_i| = \begin{cases} \max\{0, K+T-KL-1\} & \text{if } i=1 \\ \max\{0, T-K+r-1\} & \text{if } 2 \le i \\ & \text{and } i \equiv 1 \pmod r \\ T-1 & \text{if } i \not\equiv 1 \pmod r \end{cases}.$$

The proof is rather technical and can be found in [16].

This allows us to compute the score of GASP_r in a straightforward but technical way ultimately leading to Theorem 1.

B. Determining the Optimal Chain Length, r^*

As stated previously, the optimal chain length, r^*, can be found by calculating S, using Theorem 1, for every $1 \le r \le \min\{K, T\}$.

In Corollary 1 we showed that for $K = L = T = n^2$, $r^* = n$. The proof of this follows from Theorem 1.

Proof of Corollary 1. Plugging $K = L = T = n^2$ in the terms of Theorem 1 yields $\gamma = -1$ if $r = n^2$ and $\gamma = 0$ if $r \le n^2 - 1$. Then,

$$\arg\max_r\{S\} = \arg\min_r\{\underbrace{r(n^2-2)+\eta(n^2-r)-\gamma}_{=g(r)}\},$$

so that a comparison of $g(n)$ to $g(n^2)$ (to eliminate γ), the application of $x - 1 < \lfloor x \rfloor$ in η, and a comparison of $g(n)$ to $g(r)$ in general complete the proof. □

The following corollary also follows from Theorem 1.

Corollary 3. In the setting of Theorem 1, if $r < z$, then $r^* = \min\{K, T, \varphi\}$. In particular, if $\max\{K, 1 + K(L-1)\} \le T$, then $r^* = K$. If $T \le K$ and $L = 1$, then $r^* = T$.

V. Lower Bounds for the Degree Table

In this section we will prove Inequality 1 in Theorem 2. The proof for Inequalities 2 and 3 in Theorem 2 and Corollary 2, 4, and 5 can be found in [16].

We will need the following lemma from the theory of sumsets (see [15, Lemma 5.3, Proposition 5.8]).

Lemma 2 ([15])**.** *Let A and B be sets of integers. Then $|A| + |B| - 1 \le |A + B|$ and if $2 \le |A|, |B|$, then equality holds iff A and B are arithmetic progressions with the same common difference.*

Using this lemma, we can prove Inequality 1 in Theorem 2.

Proof of Inequality 1 in Theorem 2. Without loss of generality, let $L \le K$. Due to the decodability property, all integers in the first K rows and first L columns are distinct among themselves and among all other entries in the table, so that we count KL for the bound and omit these integers. Next, we omit all integers in the last T rows and first L columns. The remaining entries correspond to a sumset formed by all entries of α and the last T entries of β. The minimum size of this sumset is bounded by Lemma 2 as greater or equal than $(K + T) + (T) - 1$. □

Inequality 1 in Theorem 2 shows that GASP_{r^*} is optimal for $K = 1$ or $L = 1$.

Corollary 4. If $K = 1$ or $L = 1$, then the $\mathsf{GASP}_1 = \mathsf{GASP}_{\text{big}}$ is optimal.

Inequality 3 in Theorem 2 shows that GASP_{r^*} is optimal for $T = 1$.

Corollary 5. If $T = 1$, then $\mathsf{GASP}_{\text{small}} = \mathsf{GASP}_1 = \mathsf{GASP}_{\text{big}}$ is optimal.

Acknowledgment

The first two authors were partially supported by the NSF under Grant CNS-1801630. The third author was supported by the Academy of Finland under Grant #289002. The development of parts of the results presented here started when the second and third authors attended the Dagstuhl Seminar 18511 – "Algebraic Coding Theory for Networks, Storage, and Security". The authors are grateful to the organizers of the seminar and to Schloss Dagstuhl for this opportunity and to Alessandro Neri for insightful discussions.

References

[1] R.G.L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "GASP Codes for Secure Distributed Matrix Multiplication," *arXiv:1812.09962*, 2018.

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

[3] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure distributed computing," in *International Symposium on Information Theory*, pp. 2900-2904, June 2017.

[4] Q. Yu, S. Li, N. Raviv, S.M.M. Kalan, M. Soltanolkotabi, A.S. Avestimehr, "Lagrange Coded Computing: Optimal Design for Resiliency, Security and Privacy," *arXiv:1806.00939*, 2018.

[5] Q. Yu, M.A. Maddah-Ali, A.S. Avestimehr, " Polynomial Codes: an Optimal Design for High-Dimensional Coded Matrix Multiplication," *arXiv:1705.10464*, 2017.

[6] Q. Yu, M.A. Maddah-Ali, A.S. Avestimehr, "Straggler Mitigation in Distributed Matrix Multiplication: Fundamental Limits and Optimal Coding," *arXiv:1801.07487*, 2018.

[7] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, P. Grover, "On the Optimal Recovery Threshold of Coded Matrix Multiplication," *arXiv:1801.10292*, 2018.

[8] U. Sheth, S. Dutta, M. Chaudhari, H. Jeong, Y. Yang, J. Kohonen, T. Roos, P. Grover, "An Application of Storage-Optimal MatDot Codes for Coded Matrix Multiplication: Fast k-Nearest Neighbors Estimation," *arXiv:1811.11811*, 2018.

[9] S. Li, M.A. Maddah-Ali, Q. Yu and A.S. Avestimehr, "A Fundamental Tradeoff Between Computation and Communication in Distributed Computing," in *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109-128, Jan. 2018.

[10] H.A. Nodehi and M.A. Maddah-Ali, "Limited-Sharing Multi-Party Computation for Massive Matrix Operations," in *IEEE Transactions on Information Theory*, pp. 1231-1235, 2018.

[11] W.-T. Chang, R. Tandon, " On the Capacity of Secure Distributed Matrix Multiplication," *arXiv:1806.00469*, 2018.

[12] J. Kakar, S. Ebadifar, and A. Sezgin, " Rate-Efficiency and Straggler-Robustness through Partition in Distributed Two-Sided Secure Matrix Computation," *arXiv:1810.13006*, 2018.

[13] H. Yang and J. Lee, "Secure Distributed Computing With Straggling Servers Using Polynomial Codes," in *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 141-150, Jan. 2019.

[14] A. Geroldinger, and I.Z. Ruzsa, "Combinatorial number theory and additive group theory," in *Advanced Courses in Mathematics. CRM Barcelona*, 2009.

[15] T. Tao, and V. Vu, "Additive combinatorics," in *Cambridge Studies in Advanced Mathematics*, vol. 105, 2006.

[16] R.G.L. D'Oliveira, S. El Rouayheb, D. Heinlein, and D. Karpuk, "Degree Tables for Secure Distributed Matrix Multiplication," http://eceweb1.rutgers.edu/∼csi/dtsdmm.pdf, 2019.