

# HerA Scheme: Secure Distributed Matrix Multiplication via Hermitian Codes

Roberto A. Machado

School of Mathematical and Statistical Sciences  
Clemson University  
Clemson, SC 29634 USA  
rmachad@clemson.edu

Gretchen L. Matthews

Department of Mathematics  
Virginia Tech  
Blacksburg, VA 24061 USA  
gmatthews@vt.edu

Wellington Santos

Department of Mathematics  
Virginia Tech  
Blacksburg, VA 24061 USA  
wellington@vt.edu

**Abstract**—We consider the problem of secure distributed matrix multiplication (SDMM), where a user has two matrices and wishes to compute their product with the help of  $N$  honest but curious servers under the security constraint that any information about either  $A$  or  $B$  is not leaked to any server. This paper presents a *new scheme* that considers the inner product partition for matrices  $A$  and  $B$ . Our central technique relies on encoding matrices  $A$  and  $B$  in a Hermitian code and its dual code, respectively. We present the Hermitian Algebraic (HerA) scheme, which employs Hermitian codes and characterizes the partitioning and security capacities given entries of matrices belonging to a finite field with  $q^2$  elements. We showcase that this scheme performs the secure distributed matrix multiplication in a significantly smaller finite field and expands security allowances compared to the existing results in the literature.

**Index Terms**—secure multi-party computation, distributed computation, Hermitian codes

## I. INTRODUCTION

Matrix multiplication is an essential back-end operation of numerous applications in signal processing and machine learning. When facing applications involving massive matrices, matrix multiplication in a single computer is slow, and distributed solutions need to be adopted. In such a scenario, the goal is to speed up the computational time to perform the matrix multiplication. Thus, the multiplication task is divided into smaller sub-tasks distributed across dedicated workers.

The setting for the problem considered in this paper is as follows. A user has two matrices,  $A \in \mathbb{F}_{q^2}^{a \times b}$  and  $B \in \mathbb{F}_{q^2}^{b \times c}$ , and wishes to compute their product,  $AB \in \mathbb{F}_{q^2}^{a \times c}$ , 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 (passive) in that they are not malicious and will follow the pre-agreed-upon protocol. However, any  $T$  may collude to eavesdrop and deduce information about either  $A$  or  $B$ .

We follow the setting proposed in [1], with many follow-up works [2]–[18]. The performance metric initially used was the download cost, i.e., the total amount of data downloaded by the users from the server. Subsequent work has also considered the upload cost [11], the total communication cost [15], [17], and computational costs [9], [16].

The work of Gretchen L. Matthews was partly supported by NSF DMS-2037833, NSF DMS-2201075, and the Commonwealth Cyber Initiative.

Different partitionings of the matrices lead to different trade-offs between upload and download costs. In this paper, we consider the inner product partitioning given by  $A = [A_1 \ \cdots \ A_L]$  and  $B^\top = [B_1^\top \ \cdots \ B_L^\top]$  such that  $AB = A_1B_1 + \cdots + A_LB_L$ , where all products  $A_\ell B_\ell$  are well-defined and of the same size. Under this partitioning, a polynomial code is a polynomial  $h(x, y) = f(x, y) \cdot g(x, y)$ , whose coefficients encode the sub-matrices  $A_k B_\ell$ . The  $N$  servers compute the evaluations  $h(P_1), \dots, h(P_N)$  for certain points  $P_1, \dots, P_N$  in an Hermitian curve. The servers send these evaluations to the user. The two-variable polynomial  $h(x, y)$  is constructed to ensure that no  $T$ -subset of evaluations reveals any information about  $A$  or  $B$  ( $T$ -security), and the user can reconstruct  $AB$  given all  $N$  evaluations  $h(P_1), \dots, h(P_N)$  (decodability).

Examples of polynomial schemes using the inner product partitioning are the secure MatDot codes in [6], the DFT-codes in [11], and the FTP codes [15]. Some authors started exploring two-variable polynomials in the context of secure distributed matrix multiplication using outer product partitioning, [13], [18]. One of the literature’s main focuses was minimizing the minimum amount of helping servers  $N$ , also known as the recovery threshold, to reduce the communication cost. In [15], Machado *et al.* presented a scheme to reduce the total communication by contacting more servers. Most of the constructions rely on large finite fields and even extensions of finite fields. This paper investigates the partitioning and security capacities given matrices  $A$  and  $B$  have entries in  $\mathbb{F}_{q^2}$ , a finite field with  $q^2$  elements.

We present the Hermitian Algebraic (*HerA*) scheme, a two-variable polynomial scheme inspired by Algebraic Codes in secret sharing schemes literature, specifically the Algebraic Codes for Secret Sharing Schemes were first introduced in [19], a protocol close to optimal communication efficiency and robust security with lengths not bounded by the field size. When employing this scheme to the secure matrix multiplication problem, matrix  $A$  should be encoded in a Hermitian code while matrix  $B$  is encoded in its dual. Therefore, the recovery threshold is allowed to be larger than the field’s size  $q^2$ , which no other polynomial scheme could achieve.

**Theorem 1.** *Let  $L$  and  $T$  be positive integers. Let  $A \in \mathbb{F}_{q^2}^{a \times b}$ ,*

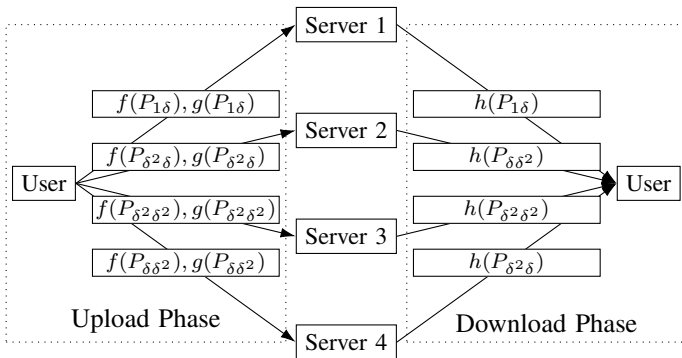


Fig. 1: An example of the *HerA* Scheme detailed in Section III. The user computes carefully chosen evaluations of the two-variable polynomials  $f(x, y)$  and  $g(x, y)$  and uploads them to the servers. This allows for reducing the minimum required size of a finite field. Then, each server computes the product of their received evaluations, which is itself an evaluation of the polynomial  $h(x) = f(x) \cdot g(x)$  and sends it back to the user who can decode  $AB$ .

$B \in \mathbb{F}_{q^2}^{b \times c}$  be two matrices and suppose the  $T$ -MDS condition is fulfilled. Then, there exists a *HerA* scheme with partitioning parameter  $L$  and security parameter  $T$ , which securely computes  $AB \in \mathbb{F}_{q^2}^{a \times c}$  utilizing  $L + 2T$  servers with a total communication rate of

$$\mathcal{R} = \left( \frac{Nb}{L} \left( \frac{1}{a} + \frac{1}{c} \right) + N \right)^{-1}. \quad (1)$$

#### A. Related Work

For distributed computations, polynomial codes were initially introduced in [20] to mitigate stragglers in distributed matrix multiplication. A series of works followed this, [21]–[24].

The literature on SDMM has also studied different variations of the model we focus on here. For instance, in [11], [25]–[27], the encoder and decoder are considered to be separate parties, in [25] servers are allowed to cooperate, and in [28] the authors consider a hybrid between SDMM and private information retrieval where the user has a matrix  $A$  and wants to privately multiply it with a matrix  $B$  belonging to some public list. *HerA* codes can be readily used or adapted to many of these settings as done with other polynomial schemes (e.g., [12], [29]).

Algebraic curves, including Hermitian ones, have gained some attention in secret sharing and fractional decoding (e.g., [19], [30]–[33]). The field trace method relevant to us was developed in [19] and later extended to Communication Efficient Quantum Secret Sharing [30].

#### B. Main Contributions

Our main contributions are summarized below.

- We present a *new* polynomial encoding scheme (called *HerA* scheme) for the secure distributed matrix multi-

plication problem, considering the inner product partition and rational functions in the Riemann-Roch space associated with a divisor of the Hermitian curve. *HerA* scheme allows for performing  $AB$  in a smaller finite field compared to the state of art in the literature.

- By carefully selecting a divisor  $mP_\infty$ , the matrix  $A$  is encoded in a Hermitian code while matrix  $B$  is encoded in its dual code, which is, by construction, also a Hermitian code. This allows us to use the inner product property of dual codes and achieve a recovery threshold of  $L + 2T$ , the same as state of the art in the literature, Theorem 1.
- As we illustrated in the examples, in Sections III and VI, *HerA* can perform information-theoretic secure distributed matrix multiplication over finite fields smaller or equal to the recovery threshold.

## II. PRELIMINARIES

This section introduces some basic notation and the main results in Hermitian codes needed for the rest of the paper. For example, we define  $[M, N] = \{M, M + 1, \dots, N\}$  and  $[M] = [1, M]$ .

We record some facts about Hermitian codes from [34].

For a prime power  $q$ , let  $\mathcal{H}_q$  denote the smooth, projective curve given by  $y^q + y = x^{q+1}$  over the finite field  $\mathbb{F}_{q^2}$ . The genus of  $\mathcal{H}_q$  is  $g = \frac{q(q-1)}{2}$ , and there are  $q + 1$  distinct  $\mathbb{F}_{q^2}$ -rational places.

Let  $P_\infty, P_1, \dots, P_n$  be the  $n + 1$  distinct  $\mathbb{F}_{q^2}$ -rational places so that  $n = q^3$ . Given  $\alpha \in \mathbb{F}_{q^2}$ , consider  $\Gamma_\alpha := \{\beta \in \mathbb{F}_{q^2} : \beta^q + \beta = \alpha^{q+1}\}$ . It is well known that for all  $\alpha \in \mathbb{F}_{q^2}$ ,  $|\Gamma_\alpha| = q$  and that the affine rational points of  $\mathcal{H}_q$  are of the form  $P_{\alpha\beta} := (\alpha, \beta) \in \mathbb{F}_{q^2} \times \Gamma_\alpha$ ; that is, the set of  $\mathbb{F}_{q^2}$  rational points of  $\mathcal{H}_q$  is

$$\mathcal{H}_q(\mathbb{F}_{q^2}) := \{P_{\alpha\beta} : \alpha \in \mathbb{F}_{q^2}, \beta \in \Gamma_\alpha\} \cup \{P_\infty\},$$

where  $P_\infty$  denotes the unique point at infinity which has projective coordinates  $(0 : 1 : 0)$ . Recall that the Riemann-Roch space of a divisor  $mP_\infty$  on  $\mathcal{H}_q$  is the subset  $\mathcal{L}(mP_\infty)$  of  $\mathbb{F}_{q^2}[x, y]$  generated by  $I(m)$ , where

$$I(m) = \{x^i y^j : 0 \leq i, 0 \leq j \leq q-1, iq + j(q+1) \leq m\}.$$

The one-point Hermitian code with design parameter  $m$  is the algebraic geometry code  $\mathcal{C}(mP_\infty) := \text{ev}(\mathcal{L}(mP_\infty))$ ; that is,

$$\mathcal{C}(mP_\infty) := \{(f(P_1), \dots, f(P_n)) : f \in \mathcal{L}(mP_\infty)\}. \quad (2)$$

Note that  $\mathcal{C}(mP_\infty)$  is a linear code of length  $n = q^3$  over the field  $\mathbb{F}_{q^2}$  and for  $m' < m$  we have  $\mathcal{C}(m'P_\infty) \subseteq \mathcal{C}(mP_\infty)$ . Moreover,  $\mathcal{C}(mP_\infty) = \{0\}$  for  $m < 0$ , and  $\mathcal{C}(mP_\infty) = \mathbb{F}_{q^2}^n$  for  $m > q^3 + q^2 - q - 2$ .

**Remark 1.** [34, Proposition 8.3.2]

For  $0 \leq m \leq q^3 + q^2 - q - 2$ , the dual code of the Hermitian code  $\mathcal{C}(mP_\infty)$  is

$$\mathcal{C}(mP_\infty)^\perp = \mathcal{C}(m^\perp P_\infty), \quad (3)$$

where  $m^\perp = q^3 + q^2 - q - 2 - m$ .

Remark 1 implies that  $\mathcal{C}(mP_\infty)$  is self-orthogonal if  $2m \leq q^3 + q^2 - q - 2$ , and  $\mathcal{C}(mP_\infty)$  is self-dual if  $m = \frac{q^3 + q^2 - q - 2}{2}$ .

**Lemma 1.** [34, Proposition 8.3.3]

Suppose that  $0 \leq m \leq s := q^3 + q^2 - q - 2$ . Then the following hold:

- i)  $\dim \mathcal{C}(mP_\infty) = \begin{cases} |I(m)| & \text{for } 0 \leq m \leq q^3 \\ q^3 - |I(m^\perp)| & \text{for } q^3 \leq m \leq s. \end{cases}$
- ii) For  $q^2 - q - 2 < m < q^3$  we have

$$\dim \mathcal{C}(mP_\infty) = m - \frac{q(q-1)}{2} + 1.$$

- iii) The minimum distance of  $\mathcal{C}(mP_\infty)$  is  $d \geq d^* = q^3 - m$ .

The value  $d^*$  is called the *designed minimum distance* of  $\mathcal{C}(mP_\infty)$ . If  $q^2 - q - 1 < m < q^3 - q^2 + q + 1$  then  $d = d^* = q^3 - m$ .

Theoretically, a Hermitian code can also be constructed by evaluating  $f(x, y) \in \mathcal{L}(mP_\infty)$  at a proper subset of the affine rational points, but then Remark 1 may no longer hold.

### III. A MOTIVATING EXAMPLE: $L = 2$ AND $T = 1$

We begin our description of *HerA* (Hermitian Algebraic) codes with the following example, which we present in as much detail as possible to present the crucial components of the scheme. We compare the size of the field required for a Discrete Fourier Transform code and a *HerA* code to illustrate the flexibility *HerA* codes have.

In this example, a user desires to compute the product of two matrices  $A = [A_1 \ A_2] \in \mathbb{F}_5^{a \times b}$  and  $B$  where  $B^\top = [B_1^\top \ B_2^\top] \in \mathbb{F}_5^{c \times b}$  such that  $AB = A_1B_1 + A_2B_2$  with the assistance of non-colluding helper servers. The solution via Discrete Fourier Transform codes schemes utilizes  $N = 4$  servers. It involves picking two random matrices  $R \in \mathbb{F}_5^{a \times \frac{b}{2}}$  and  $S \in \mathbb{F}_5^{\frac{b}{2} \times c}$  and constructing the one-variable polynomials  $f'(x) = A_1 + A_2x + Rx^2$  and  $g'(x) = B_1 + B_2x^{-1} + Sx^{-3}$ . The user then selects four distinct non-zero elements  $\beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{F}_5$  and uploads both  $f'(\beta_i)$  and  $g'(\beta_i)$  to Server  $i$ . Each server then computes the product  $f'(\beta_i) \cdot g'(\beta_i)$ . This is equivalent to computing an evaluation  $h'(\beta_i)$  of the polynomial  $h'(x) = (A_1B_1 + A_2B_2) + (A_2B_1 + RB_2)x + RB_1x^2 + A_1Sx^{-3} + A_2Sx^{-2} + (A_1B_2 + RS)x^{-1}$ . The user then downloads each  $h'(\beta_i)$ , obtaining four evaluations of a polynomial of degree two. Therefore, the user can retrieve  $AB = A_1B_1 + A_2B_2$  by operating  $4 \sum_{i=1}^4 h'(\beta_i)$  since  $\sum_{i=1}^4 (3^i)^s = 0 \ \forall s : 4 \nmid s$ .

The security of the Discrete Fourier Transform codes follows from the fact that  $I(f'(\beta_i), g'(\beta_i); A, B) = 0$ . As for the communication costs, first the user uploads  $f'(\beta_i)$  and  $g'(\beta_i)$ , which cost  $ab$  and  $bc$ , symbols respectively, four times. Thus, the upload cost is  $4(ab + bc)$  symbols of  $\mathbb{F}_5$ . Then, the user downloads  $h'(\beta_i)$ , which costs  $ac$  symbols of  $\mathbb{F}_5$ , four times, obtaining a download cost of  $4ac$  symbols of  $\mathbb{F}_5$ . Since the user retrieves  $AB \in \mathbb{F}_5^{a \times c}$ , which consists of  $ac$  symbols of  $\mathbb{F}_5$ , the total communication rate is given by  $\mathcal{R}' = \frac{ac}{4ab + 4bc + 4ac}$ .

The setting we consider for our construction is similar to the one considered for DFT codes except for the size of the field,

i.e., a user wants to compute the product of two matrices  $A = [A_1 \ A_2] \in \mathbb{F}_4^{a \times b}$  and  $B$  where  $B^\top = [B_1^\top \ B_2^\top] \in \mathbb{F}_4^{c \times b}$  such that  $AB = A_1B_1 + A_2B_2$ . We also pick two random matrices  $R \in \mathbb{F}_4^{a \times \frac{b}{2}}$  and  $S \in \mathbb{F}_4^{\frac{b}{2} \times c}$ . Based on properties of Hermitian codes, we present an *HerA* code which allows multiplying matrices in a smaller finite field utilizing  $N = 4$  servers.

Let  $\delta \in \mathbb{F}_4$  denote an algebraic element in  $\mathbb{F}_4$  such that  $\delta^2 + \delta + 1 = 0$ . Let  $P_{\alpha\beta} \in \mathbb{F}_4 \times \Gamma_\alpha$  denote rational points satisfying  $\beta^2 + \beta = \alpha^3$ . Therefore,  $P_{00} = (0, 0)$ ,  $P_{01} = (0, 1)$ ,  $P_{1\delta} = (1, \delta)$ ,  $P_{1\delta^2} = (1, \delta^2)$ ,  $P_{\delta\delta} = (\delta, \delta)$ ,  $P_{\delta\delta^2} = (\delta, \delta^2)$ ,  $P_{\delta^2\delta} = (\delta^2, \delta)$  and  $P_{\delta^2\delta^2} = (\delta^2, \delta^2)$  are all the possible affine rational points. Let  $f(x, y)$  be a two-variable polynomial generated by monomials  $\{1, x, y\}$  such that  $f(P_{00}) = A_1$ ,  $f(P_{01}) = A_2$  and  $f(P_{1\delta}) = R$ . Let  $g(x, y)$  be a two-variable polynomial generated by monomials  $\{1, x, x^2, y, xy\}$  such that  $g(P_{00}) = B_1$ ,  $g(P_{01}) = B_2$ ,  $g(P_{1\delta}) = S$ ,  $g(P_{1\delta^2}) = 0$  and  $g(P_{\delta\delta}) = 0$ . The explicit polynomials are

$$f(x, y) = A_1 + (A_1 + R + (A_1 + A_2)\delta)x + (A_1 + A_2)y$$

and

$$g(x, y) = B_1 + (S + B_1\delta^2 + B_2)x + (B_1\delta + B_2 + S\delta)x^2 + (B_1 + B_2)y + (B_1 + B_2 + S)xy$$

Then,  $h(x, y) = f(x, y) \cdot g(x, y)$  is such that  $h(P_{00}) + h(P_{01}) = A_1B_1 + A_2B_2 = AB$ .

Since  $(f(P_{00}), f(P_{01}), \dots, f(P_{\delta^2\delta^2})) \in \mathcal{C}(3P_\infty)$  and

$$(g(P_{00}), g(P_{01}), \dots, g(P_{\delta^2\delta^2})) \in \mathcal{C}(5P_\infty),$$

the dual of  $\mathcal{C}(3P_\infty)$ , it follows that

$$\sum_{\alpha\beta} f(P_{\alpha\beta}) \cdot g(P_{\alpha\beta}) = \sum_{\alpha\beta} h(P_{\alpha\beta}) = 0. \quad (4)$$

Our scheme works as follows: the user uploads the evaluations  $f(P_i)$  and  $g(P_i)$  to each Server  $i$ , where  $(P_1, P_2, P_3, P_4) = (P_{1\delta}, P_{\delta\delta^2}, P_{\delta^2\delta}, P_{\delta^2\delta^2})$ . Then, each Server  $i$  computes  $-h(P_i)$ , and sends it back to the user. The user can decode  $AB$  as follows:

$$\begin{aligned} & -h(P_{1\delta}) - h(P_{\delta\delta^2}) - h(P_{\delta^2\delta}) - h(P_{\delta^2\delta^2}) \\ &= -h(P_{1\delta}) - h(P_{1\delta^2}) - h(P_{\delta\delta}) - h(P_{\delta\delta^2}) - h(P_{\delta^2\delta}) - h(P_{\delta^2\delta^2}) \\ &= h(P_{00}) + h(P_{01}) = A_1B_1 + A_2B_2 = AB. \end{aligned}$$

The first equality follows from the fact that  $h(P_{1\delta^2}) = f(P_{1\delta^2}) \cdot g(P_{1\delta^2}) = 0 = f(P_{\delta\delta}) \cdot g(P_{\delta\delta}) = h(P_{\delta\delta})$  since  $g(P_{1\delta^2}) = g(P_{\delta\delta}) = 0$ . The second equality follows from Equation 4.

Security follows by showing that  $I(f(P_i), g(P_i); A, B) = 0$ , as is done in Lemma 3. As for the communication costs, first, the user uploads  $f(P_i)$  and  $g(P_i)$ , which cost  $2ab$  and  $2bc$ , symbols, respectively. Thus, the upload cost is  $(2ab + 2bc)$  symbols of  $\mathbb{F}_4$ . Then, the user downloads  $h(P_i)$ , which costs  $ac$  symbols of  $\mathbb{F}_4$ , four times, obtaining a download cost of  $4ac$  symbols of  $\mathbb{F}_4$ . Since the user retrieves  $AB \in \mathbb{F}_4^{a \times c}$ , which

consists of  $2ac$  symbols of  $\mathbb{F}_4$ , the total communication rate is given by  $\mathcal{R} = \frac{ac}{4ab+4bc+2ac}$ .

We note that the total communication rate of the *HerA* code is equal to the total communication rate of the DFT code. However, we showcase that the *HerA* code utilizes a smaller field size avoiding the divisibility constraint required for DFT codes. It raises the theoretical question on the field's capacity: Is  $\mathbb{F}_4$  the smallest field to perform the product of matrices  $A$  and  $B$  given  $L = 2$  and  $T = 1$ ? On the other side, what are the maximum partitioning and security parameters allowed in secure distributed matrix multiplication over  $\mathbb{F}_4$ ?

#### IV. HERA SCHEME

This section presents the general construction for the *HerA* Scheme. The main idea is to perform the same technique as in Section III that retrieves the product  $AB$  using the inner product and encoding matrix  $A$  in a Hermitian code while matrix  $B$  is encoded in its dual code. Consider the matrices  $A \in \mathbb{F}_{q^2}^{a \times b}$ ,  $B \in \mathbb{F}_{q^2}^{b \times c}$ .

**Choosing Parameters  $L$ ,  $T$  and  $m$ :** We begin by choosing parameters  $L$  and  $T$  such that  $2(L+T) \leq q^3 - \frac{q(q-1)}{2}$  and set  $m = L + T + \frac{q(q-1)}{2} - 1$ . We remark that the bound  $2(L+T) \leq q^3 - \frac{q(q-1)}{2}$  explicates a finite field's partitioning and security capacities with  $q^2$  elements.

**Choosing the Polynomials:** As described in the introduction, we consider the setting where the user partitions the matrices  $A \in \mathbb{F}_{q^2}^{a \times b}$  and  $B \in \mathbb{F}_{q^2}^{b \times c}$  as  $A = [A_1 \dots A_L]$  and as  $B^T = [B_1^T \dots B_L^T]$  such that  $AB = A_1B_1 + \dots + A_LB_L$ , where each  $A_i \in \mathbb{F}_{q^2}^{a \times \frac{b}{L}}$ ,  $B_i \in \mathbb{F}_{q^2}^{\frac{b}{L} \times c}$ . In order to obtain  $T$ -security,  $R_1, \dots, R_T \in \mathbb{F}_{q^2}^{a \times \frac{b}{L}}$  and  $S_1, \dots, S_T \in \mathbb{F}_{q^2}^{\frac{b}{L} \times c}$  are chosen independently and uniformly at random. We then choose  $\{P_1, \dots, P_{L+T}\} \subseteq \mathcal{H}_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$ ,  $f \in \mathcal{L}(mP_\infty)$ , and  $g \in \mathcal{L}(m^\perp P_\infty)$  such that  $f(P_i) = A_i$ ,  $g(P_i) = B_i$  for every  $i \in [L]$ ;  $f(P_{L+i}) = R_i$ ,  $g(P_{L+i}) = S_i$  for every  $i \in [T]$ ; and  $g(P_{L+T+i}) = 0$  for every  $i \in [q^3 - 2(L+T)]$ .

**Upload Phase:** The *HerA* scheme uses  $L + 2T$  servers. The user uploads  $f(P_{L+i}), g(P_{L+i})$  to the server  $N_i$ ,  $i \in [T]$  and  $f(P_{q^3-L-T+i}), g(P_{q^3-L-T+i})$  to the server  $N_{T+i}$ ,  $i \in [L+T]$ .

**Download Phase:** Each server  $N_i$ ,  $i \in [T]$  computes  $-h(P_{L+i}) = -f(P_{L+i})g(P_{L+i})$ , and each server  $N_{T+i}$ ,  $i \in [L+T]$  computes  $-h(P_{q^3-L-T+i}) = -f(P_{q^3-L-T+i})g(P_{q^3-L-T+i})$  and sends these values to the user.

**User Decoding:** In Lemma 2, we show that the user can decode  $AB = h(P_1) + h(P_2) + \dots + h(P_L)$  from  $\{-h(P_{L+i})\}_{i=1}^T \cup \{-h(P_{q^3-L-T+i})\}_{i=1}^{L+T}$ .

#### V. PROOF OF THEOREM 1

We split the proof into lemmas. We show that *HerA* schemes are decodable in Lemma 2 and  $T$ -secure, in Lemma 3. These statements combined prove Theorem 1.

**Lemma 2.** *Given a prime power  $q$  and positive integers  $L$  and  $T$  such that  $L+T \leq \frac{q^3}{2}$ , define  $m := L+T + \frac{q(q-1)}{2} - 1$  and let  $\mathcal{L}(mP_\infty)$  be the Riemann-Roch space of the divisor*

*$mP_\infty$  on the Hermitian curve  $\mathcal{H}_q : y^q + y = x^{q+1}$ . If  $A = [A_1 \dots A_L] \in \mathbb{F}_{q^2}^{a \times b}$  and  $B^T = [B_1^T \dots B_L^T] \in \mathbb{F}_{q^2}^{c \times b}$ . Then,  $h(P_1) + \dots + h(P_L)$  can be decoded using  $L + 2T$  servers.*

*Proof.* Let  $\{P_i\}_{i=1}^{q^3-L-T} \subseteq \mathcal{H}_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$ ,  $f \in \mathcal{L}(mP_\infty)$  and  $g \in \mathcal{L}(m^\perp P_\infty)$  be polynomials such that

$$\begin{aligned} f(P_i) &= A_i, & g(P_i) &= B_i \forall i \in [L], \\ f(P_{L+i}) &= R_i, & g(P_{L+i}) &= S_i \forall i \in [T], \\ g(P_{L+T+i}) &= 0 \forall i \in [q^3 - 2(L+T)], \end{aligned}$$

using the inner product partitioning  $A = [A_1 \dots A_L]$  and  $B^T = [B_1^T \dots B_L^T]$  and uniformly distributed random  $\mathbb{F}_{q^2}$ -matrices  $R_i, S_i$ . Therefore,  $h(x, y) = f(x, y)g(x, y)$  is such that  $h(P_i) = A_iB_i$  for all  $i \in [L]$ . Note that

$$(f(P_1), \dots, f(P_{q^3})) \in \mathcal{C}(mP_\infty)$$

and

$$(g(P_1), \dots, g(P_{q^3})) \in \mathcal{C}(m^\perp P_\infty)$$

since  $f \in \mathcal{L}(mP_\infty)$  and  $g \in \mathcal{L}(m^\perp P_\infty)$ . The dual-code property implies that

$$\begin{aligned} 0 &= \sum_{i=1}^{q^3} f(P_i)g(P_i) \\ &= \sum_{i=1}^L f(P_i)g(P_i) + \sum_{i=L+1}^{L+T} f(P_i)g(P_i) \\ &\quad + \sum_{i=q^3-L-T+1}^{q^3} f(P_i)g(P_i). \end{aligned}$$

So,  $\sum_{i=1}^L f(P_i)g(P_i) =$

$$-\sum_{i=L+1}^{L+T} f(P_i)g(P_i) - \sum_{i=q^3-L-T+1}^{q^3} f(P_i)g(P_i), \quad (5)$$

proving that  $AB$  is performed using  $L + 2T$  servers.  $\square$

**Definition 1.** The functions  $f_1(x, y), \dots, f_T(x, y) \in \mathcal{L}(mP_\infty)$  and the set  $\mathcal{T} \subset \{1, 2, \dots, q^3\}$  satisfy the  $T$ -MDS condition if

$$\mathbf{F}^{(\mathcal{T})} = \begin{pmatrix} f_1(P_{i_1}) & f_1(P_{i_2}) & \dots & f_1(P_{i_T}) \\ f_2(P_{i_1}) & f_2(P_{i_2}) & \dots & f_2(P_{i_T}) \\ \vdots & \vdots & \ddots & \vdots \\ f_T(P_{i_1}) & f_T(P_{i_2}) & \dots & f_T(P_{i_T}) \end{pmatrix}$$

has full rank for any different  $i_1, i_2, \dots, i_T \in \mathcal{T}$  and  $P_{i_1}, P_{i_2}, \dots, P_{i_T} \in \mathcal{H}_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$ .

**Lemma 3.** *Let  $f(x, y) = \sum_{i=0}^{L+T} f_i(x, y)$  be a polynomial encoding the matrix  $A$  satisfying conditions in Lemma 2. If  $f_{L+1}(x, y), f_{L+2}(x, y), \dots, f_{L+T}(x, y)$  and there is  $\mathcal{T} \subset \{L+1, \dots, L+T\} \cup \{q^3 - 2(L+T), q^3\}$  with  $|\mathcal{T}| = L + 2T$  satisfying the  $T$ -MDS condition, then  $I(A; f(P_{i_1}), f(P_{i_2}), \dots, f(P_{i_T})) = 0$ . A similar argument holds for  $g(x, y)$  and  $B$ , implying that the *HerA* scheme is  $T$ -secure.*

*Proof.* Since  $f(x, y)$  is independent of  $B$  and  $g(x, y)$  is independent of  $A$ , proving  $T$ -security is equivalent to showing that  $I(A; f(P_{i_1}), \dots, f(P_{i_T})) = I(B; g(P_{i_1}), \dots, g(P_{i_T})) = 0$ , for any

$$\mathcal{T} = \{i_1, \dots, i_T\} \subset \{L+1, \dots, L+T\} \cup \{q^3 - 2(L+T), q^3\}.$$

We prove the claim for  $f(x, y)$  since the proof for  $g(x, y)$  is analogous.

Since  $m = L + T + \frac{q(q-1)}{2} - 1$ ,  $|I(m)| = L + T$ . Therefore, there exists  $\{P_1, \dots, P_{L+T}\} \subseteq \mathcal{H}_q(\mathbb{F}_{q^2}) \setminus \{P_\infty\}$  and  $f(x, y)$  expressed as

$$f(x, y) = \sum_{i=1}^{L+T} f_i(x, y) f(P_i),$$

where each  $f_i(x, y) \in \mathcal{L}(mP_\infty)$  with

$$f_i(x, y) = \begin{cases} 1 & \text{if } (x, y) = P_i \\ 0 & \text{if } (x, y) = P_j \text{ and } j \in [L+T] \setminus \{i\}. \end{cases}$$

Then,

$$\begin{aligned} & I(A; f(P_{i_1}), \dots, f(P_{i_T})) \\ &= H(f(P_{i_1}), \dots, f(P_{i_T})) - H(f(P_{i_1}), \dots, f(P_{i_T})|A) \\ &\leq \sum_{j \in \mathcal{T}} H(f(P_j)) - H(f(P_{i_1}), \dots, f(P_{i_T})|A) \\ &= \frac{Tab}{L} \log(q^2) - \frac{\text{rank}(\mathbf{F}^{(T)})ab}{L} \log(q^2). \end{aligned}$$

Since the evaluation points  $E = \{P_i : i \in \mathcal{T}\}$  are such that  $\mathbf{F}^{(T)}$  has full rank for any  $T$  different  $P_i$ 's in  $E$ , the  $f^{(T)}(P_{i_j})$ 's are uniformly distributed in the space of the matrices  $M_{a \times \frac{b}{T}}(\mathbb{F}_{q^2})$ . Thus,  $H(f^{(T)}(P_{i_1}), \dots, f^{(T)}(P_{i_T})) = \frac{Tab}{L} \log(q^2)$ ; therefore,  $I(A; f(P_{i_1}), \dots, f(P_{i_T})) = 0$ .  $\square$

## VI. EXAMPLE: $L = 2, T = 2$ AND $q = 3$

**Choosing Parameters  $L, T$  and  $m$ :** Since  $L = T = 2$ , the *HerA* scheme can be performed on  $\mathbb{F}_{3^2}$  ( $q = 3$ ), setting  $m = 6$ .

**Choosing the Polynomials:** Since  $L = 2$ , the matrices  $A \in \mathbb{F}_9^{a \times b}$  and  $B \in \mathbb{F}_9^{b \times c}$  are partitioned as  $A = [A_1 A_2]$  and  $B^T = [B_1^T B_2^T]$  such that  $AB = A_1 B_1 + A_2 B_2$ , with  $A_i \in \mathbb{F}_9^{a \times \frac{b}{2}}$ ,  $B_i \in \mathbb{F}_9^{\frac{b}{2} \times c}$ , for  $i = 1, 2$ . In order to obtain 2-security,  $R_1, R_2 \in \mathbb{F}_9^{a \times \frac{b}{2}}$  and  $S_1, S_2 \in \mathbb{F}_9^{\frac{b}{2} \times c}$  are chosen independently and uniformly at random. Choose  $\{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8\} \subseteq \mathcal{H}_3(\mathbb{F}_9) \setminus \{P_\infty\}$ ,  $f \in \mathcal{L}(6P_\infty)$ ,  $g \in \mathcal{L}(25P_\infty)$  such that  $f(P_i) = A_i$ ,  $g(P_i) = B_i$  for every  $i \in [2]$ ;  $f(P_{2+i}) = R_i$ ,  $g(P_{2+i}) = S_i$  for every  $i \in [2]$  and  $g(P_j) = 0$  for all  $P_j \in \mathcal{H}_3(\mathbb{F}_9) \setminus (\{P_\infty\} \cup \{P_i\}_{i=1}^8)$ .

In this example, for

$$P_1 = (0, 0), P_2 = (0, \delta + 1), P_3 = (1, 2), P_4 = (\delta, 1),$$

$$P_5 = (2, 2), P_6 = (\delta + 1, 2), P_7 = (\delta + 2, \delta + 2), P_8 = (2\delta, 1),$$

with  $\delta \in \mathbb{F}_9$  a primitive element, transforming it in an interpolation problem. With the help of the Mathematics

Software SageMath [35], we found a solution for the system of equations leading to the following encoding polynomials

$$f(x, y) = f_1(x, y)A_1 + f_2(x, y)A_2 + f_3(x, y)R_1 + f_4(x, y)R_2$$

where  $f_1(x, y) = 1 + \delta x^2 + (\delta + 1)y$ ,  $f_2(x, y) = 2\delta x + 2x^2 + (2\delta + 2)y$ ,  $f_3(x, y) = (\delta + 1)x + 2\delta x^2$  and  $f_4(x, y) = 2x + x^2$  and

$$g(x, y) = g_1(x, y)B_1 + g_2(x, y)B_2 + g_3(x, y)S_1 + g_4(x, y)S_2$$

where  $g_1(x, y) = (1 + 2x + xy + 2x^3y + x^4y + x^5y + x^6y + y^2 + 2x^2y^2 + x^3y^2 + x^4y^2 + x^5y^2) + \delta(2x^4 + 2x^7 + x^8) + (\delta + 2)(x^2 + x^3) + (2\delta + 2)x^6$  and  $g_2(x, y) = (2x^2 + 2x^3 + x^2y + x^3y + x^4y + y^2) + \delta(2x + 2x^5 + 2x^6 + 2x^7 + 2xy + 2x^5y + xy^2 + 2x^4y^2 + x^5y^2) + (\delta + 1)(y + x^2y^2 + x^3y^2) + (\delta + 2)x^8 + (2\delta + 1)(x^7y + x^6y)$   $g_3(x, y) = (x^6y + 2x^7y + x^7) + \delta(x^5 + x^6 + x^3y) + (\delta + 1)(x^2 + x^3 + x^5y^2 + x^8) + (\delta + 2)(xy + x^2y^2) + 2\delta(x^2y + x^5y + xy^2) + (2\delta + 2)(x + x^4)$   $g_4(x, y) = (x^2y^2 + 2x^5y^2) + \delta(x + x^7 + x^6y + x^7y) + (\delta + 1)(x^3 + x^6 + x^3y) + (\delta + 2)(x^4 + x^3y^2) + 2\delta(xy + x^5y) + (2\delta + 1)(x^2 + x^8 + xy^2) + (2\delta + 2)x^2y$ .

**Upload Phase:** The *HerA* scheme uses 6 servers. The user uploads  $f(P_{2+i}), g(P_{2+i})$  to the  $i$ -th server.

**Download Phase:** Each server computes  $-h(P_{2+i}) = -f(P_{2+i})g(P_{2+i})$  and sends the values back to the user.

**User Decoding:** In Lemma 2, we show that the user can decode  $AB = h(P_1) + h(P_2)$  from

$$\{-h(P_{2+i})\}_{i=1}^2 \cup \{-h(P_{4+i})\}_{i=1}^4.$$

It can be checked that  $f_3(x, y), f_4(x, y)$  and  $\mathcal{T} = \{3, 4, \dots, 8\}$  satisfy the 2-MDS condition. Similarly,  $g_3(x, y), g_4(x, y)$  and the same  $\mathcal{T}$  satisfy the 2-MDS condition. Therefore, the scheme is 2-secure.

## VII. CONCLUSION

This paper proposes a secret-sharing-based scheme for secure distributed matrix multiplication. We give a general framework based on Hermitian codes achieving the same recovery threshold as the state-of-the-art in the literature while still using small finite fields, allowing more partitioning and security in a fixed finite field  $\mathbb{F}_{q^2}$ .

## REFERENCES

- [1] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [2] J. Kakar, S. Ebadifar, and A. Sezgin, "On the capacity and straggler-robustness of distributed secure matrix multiplication," *IEEE Access*, vol. 7, pp. 45 783–45 799, 2019.
- [3] H. Yang and J. Lee, "Secure distributed computing with straggling servers using polynomial codes," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 1, pp. 141–150, 2018.
- [4] R. G. L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "Gasp codes for secure distributed matrix multiplication," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 1107–1111.
- [5] R. G. L. D'Oliveira, S. El Rouayheb, D. Heinlein, and D. Karpuk, "Degree tables for secure distributed matrix multiplication," in *2019 IEEE Information Theory Workshop (ITW)*, 2019.

- [6] M. Aliasgari, O. Simeone, and J. Kliewer, "Distributed and private coded matrix computation with flexible communication load," *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1092–1096, 2019.
- [7] —, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2722–2734, 2020.
- [8] J. Kakar, A. Khristoforov, S. Ebadifar, and A. Sezgin, "Uplink-downlink tradeoff in secure distributed matrix multiplication," *ArXiv*, vol. abs/1910.13849, 2019.
- [9] R. G. L. D'Oliveira, S. E. Rouayheb, D. Heinlein, and D. Karpuk, "Notes on communication and computation in secure distributed matrix multiplication," in *2020 IEEE Conference on Communications and Network Security (CNS)*, 2020, pp. 1–6.
- [10] Q. Yu and A. S. Avestimehr, "Entangled polynomial codes for secure, private, and batch distributed matrix multiplication: Breaking the "cubic" barrier," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 245–250.
- [11] N. Mital, C. Ling, and D. Gündüz, "Secure distributed matrix computation with discrete fourier transform," *IEEE Transactions on Information Theory*, pp. 1–1, 2022.
- [12] R. Bitar, M. Xhemrishi, and A. Wachter-Zeh, "Adaptive private distributed matrix multiplication," *arXiv preprint arXiv:2101.05681*, 2021.
- [13] B. Hasircioğlu, J. Gómez-Vilardebó, and D. Gündüz, "Speeding up private distributed matrix multiplication via bivariate polynomial codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1853–1858.
- [14] H. H. López, G. L. Matthews, and D. Valvo, "Secure matdot codes: a secure, distributed matrix multiplication scheme," in *2022 IEEE Information Theory Workshop (ITW)*, 2022, pp. 149–154.
- [15] R. A. Machado, R. G. L. D'Oliveira, S. E. Rouayheb, and D. Heinlein, "Field trace polynomial codes for secure distributed matrix multiplication," in *2021 XVII International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*, 2021, pp. 188–193.
- [16] R. A. Machado and F. Manganiello, "Root of unity for secure distributed matrix multiplication: Grid partition case," in *2022 IEEE Information Theory Workshop (ITW)*, 2022, pp. 155–159.
- [17] R. G. L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "Gasp codes for secure distributed matrix multiplication," *IEEE Transactions on Information Theory*, pp. 1–1, 2020.
- [18] B. Hasircioğlu, J. Gómez-Vilardebó, and D. Gündüz, "Bivariate polynomial codes for secure distributed matrix multiplication," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 955–967, 2022.
- [19] U. Martínez-Peñas, "Communication efficient and strongly secure secret sharing schemes based on algebraic geometry codes," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4191–4206, 2018.
- [20] 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*, 2017, pp. 4403–4413.
- [21] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2022–2026.
- [22] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Transactions on Information Theory*, 2019.
- [23] U. Sheth, S. Dutta, M. Chaudhari, H. Jeong, Y. Yang, J. Kohonen, T. Roos, and P. Grover, "An application of storage-optimal matdot codes for coded matrix multiplication: Fast k-nearest neighbors estimation," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1113–1120.
- [24] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2017.
- [25] 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)*. IEEE, 2018, pp. 1231–1235.
- [26] Z. Jia and S. A. Jafar, "On the capacity of secure distributed matrix multiplication," *arXiv preprint arXiv:1908.06957*, 2019.
- [27] H. Akbari-Nodehi and M. A. Maddah-Ali, "Secure coded multi-party computation for massive matrix operations," *IEEE Transactions on Information Theory*, vol. 67, no. 4, pp. 2379–2398, 2021.
- [28] M. Kim, H. Yang, and J. Lee, "Private coded matrix multiplication," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1434–1443, 2019.
- [29] J. Zhu, Q. Yan, and X. Tang, "Improved constructions for secure multi-party batch matrix multiplication," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7673–7690, 2021.
- [30] K. Senthoo and P. K. Sarvepalli, "Concatenating extended css codes for communication efficient quantum secret sharing," *arXiv preprint arXiv:2211.06910*, 2022.
- [31] M. Cheraghchi, "Nearly optimal robust secret sharing," *Designs, Codes and Cryptography*, vol. 87, no. 8, pp. 1777–1796, 2019.
- [32] G. L. Matthews, A. W. Murphy, and W. Santos, "Fractional decoding of codes from Hermitian curves," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 515–520.
- [33] A. W. Murphy, "Codes from norm-trace curves: local recovery and fractional decoding," Ph.D. dissertation, Virginia Tech, 2022.
- [34] H. Stichtenoth, *Algebraic Function Fields and Codes*. Springer Berlin, Heidelberg, 2009, vol. 2.
- [35] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 9.4)*, 2021, <https://www.sagemath.org>.