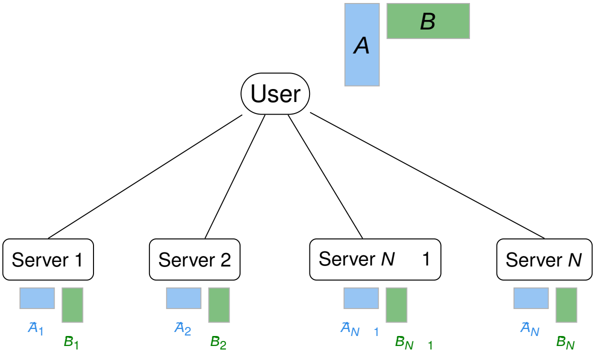
How to distribute the multiplication of Secret Matrices?

Rafael G.L. D'Oliveira
Salim El Rouayheb
Daniel Heinlein
David Karpuk

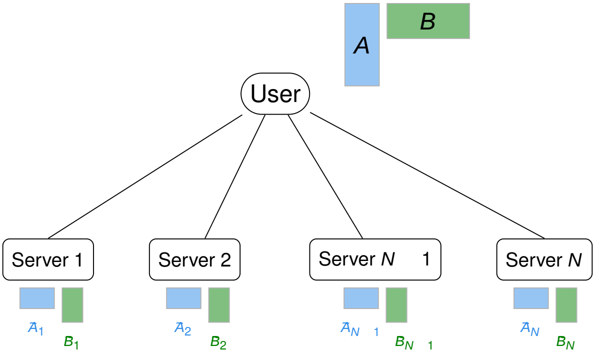
Massachusetts Institute of Technology
Rutgers University
Aalto University
and
F-Secure

Setup



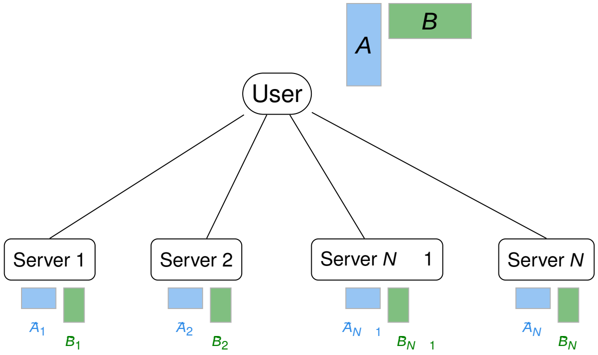
- User has two matrices $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$ and wants their product AB.

Setup



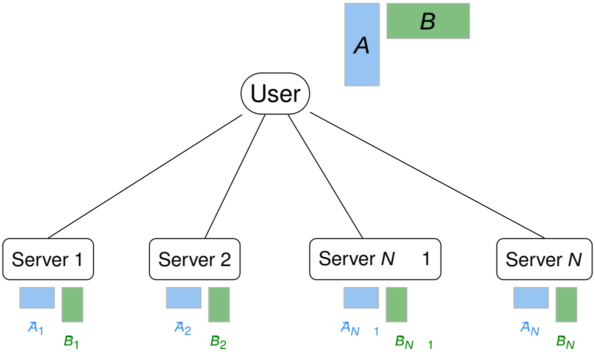
- User has two matrices $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$ and wants their product AB.
- N helper servers. Honest but curious.

Setup



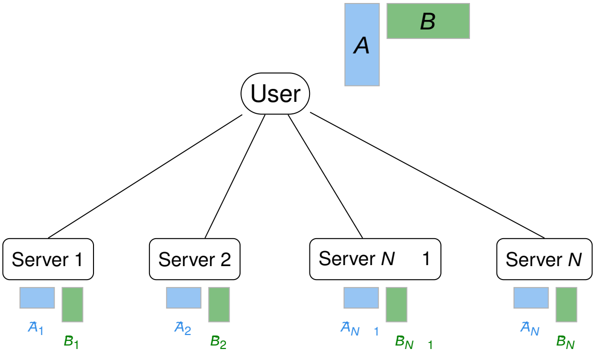
- User has two matrices $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$ and wants their product AB.
- N helper servers. Honest but curious.
- Want information theoretic Privacy even if T server collude.

Setup



- User has two matrices $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$ and wants their product AB.
- N helper servers. Honest but curious.
- Want information theoretic Privacy even if T server collude.
- Figure of merit: communication cost.

Setup

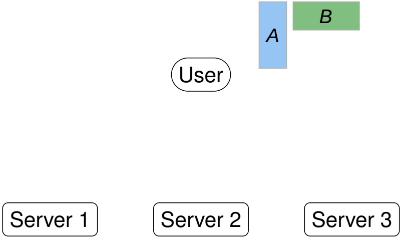


- User has two matrices $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$ and wants their product AB.
- N helper servers. Honest but curious.
- Want information theoretic Privacy even if T server collude.
- Figure of merit: communication cost.
- Matrix Multiplication is everywhere!

Simplest Example: Polynomial Codes/Secret Sharing

$f(x) = A + Rx$

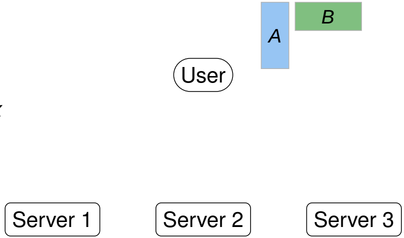
$g(x) = B + Sx$



Simplest Example: Polynomial Codes/Secret Sharing

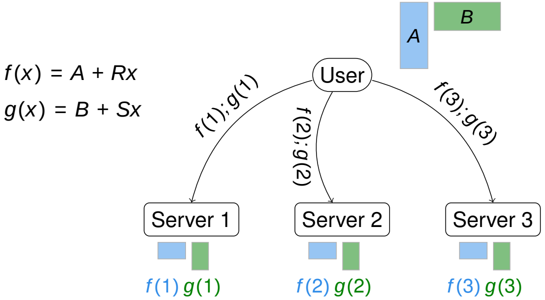
$f(x) = A + Rx$

$g(x) = B + Sx$



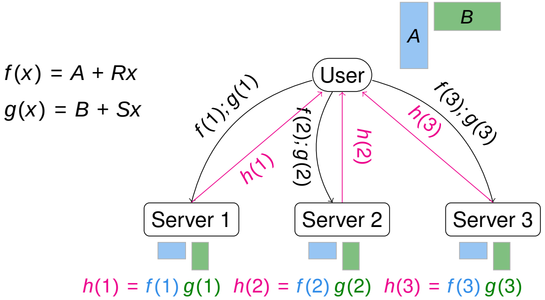
- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.

Simplest Example: Polynomial Codes/Secret Sharing



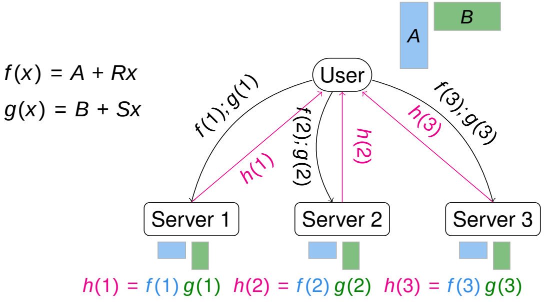
- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.

Simplest Example: Polynomial Codes/Secret Sharing



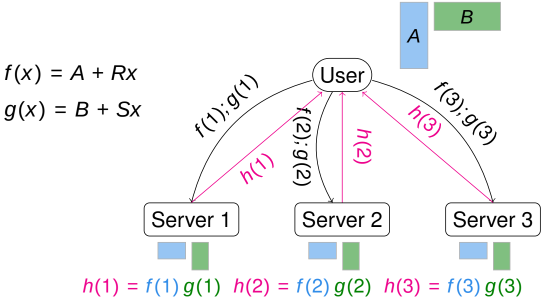
- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.
- $h(x) := f(x)g(x) = AB + (AS + RB)x + RSx^2$

Simplest Example: Polynomial Codes/Secret Sharing



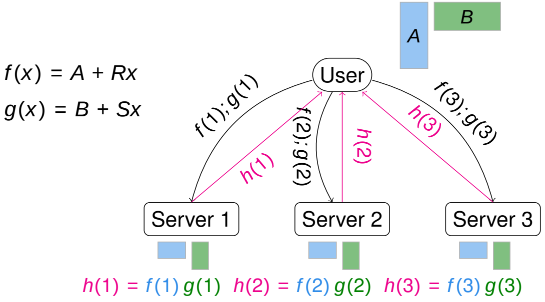
- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.
- $h(x) := f(x)g(x) = AB + (AS + RB)x + RSx^2$
- User wants $AB = h(0)$.

Simplest Example: Polynomial Codes/Secret Sharing



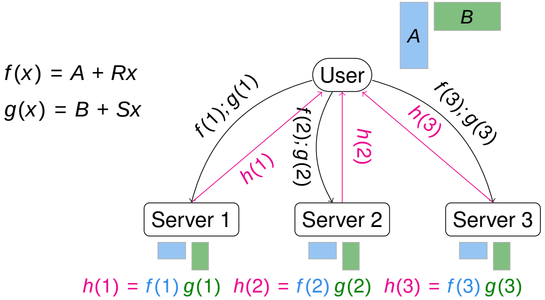
- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.
- $h(x) := f(x)g(x) = AB + (AS + RB)x + RSx^2$
- User wants $AB = h(0)$.
- Server i computes $h(i) = f(i)g(i)$ and sends it to the user.

Simplest Example: Polynomial Codes/Secret Sharing



- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.
- $h(x) := f(x)g(x) = AB + (AS + RB)x + RSx^2$
- User wants $AB = h(0)$.
- Server i computes $h(i) = f(i)g(i)$ and sends it to the user.
- User interpolates $h(x)$ and decodes $AB = h(0)$.

Simplest Example: Polynomial Codes/Secret Sharing



- Generate random R and S same size as A and B, resp and forms $f(x) = A + Rx; g(x) = B + Sx$.
- User sends $f(i)$ and $g(i)$ to server i.
- $h(x) := f(x)g(x) = AB + (AS + RB)x + RSx^2$
- User wants $AB = h(0)$.
- Server i computes $h(i) = f(i)g(i)$ and sends it to the user.
- User interpolates $h(x)$ and decodes $AB = h(0)$.
- Comm. cost = 3 (upload A + upload B + download AB).

Divide & Parallelize

- Let $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$
- We divide A and B as $A = \begin{bmatrix} A_1 \\ \vdots \\ A_K \end{bmatrix}$ and $B = \begin{bmatrix} B_1 & \cdots & B_L \end{bmatrix}.$

Divide & Parallelize

- Let $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$
- We divide A and B as $A = \begin{bmatrix} A_1 \\ \vdots \\ A_K \end{bmatrix}$ and $B = \begin{bmatrix} B_1 & \cdots & B_L \end{bmatrix}$.
- $AB = \begin{bmatrix} A_1B_1 & \cdots & A_1B_L \\ \vdots & \ddots & \vdots \\ A_KB_1 & \cdots & A_KB_L \end{bmatrix}$

Divide & Parallelize

- Let $A \in \mathbb{F}_q^{r \times s}$ and $B \in \mathbb{F}_q^{s \times t}$
- We divide A and B as $A = \begin{bmatrix} A_1 \\ \vdots \\ A_K \end{bmatrix}$ and $B = \begin{bmatrix} B_1 & \cdots & B_L \end{bmatrix}$.
- $AB = \begin{bmatrix} A_1 B_1 & \cdots & A_1 B_L \\ \vdots & \ddots & \vdots \\ A_K B_1 & \cdots & A_K B_L \end{bmatrix}$
- Each server does $\frac{1}{KL}$ of the work.

Total Communication Cost

- When using N servers, the total Communication Cost is

$$N \left(\underbrace{\frac{rs}{K} + \frac{st}{L}}_{\text{Upload}} + \underbrace{\frac{rt}{KL}}_{\text{Download}} \right)$$

Total Communication Cost

- When using N servers, the total Communication Cost is

$$N \quad \underbrace{\frac{rs}{K} + \frac{st}{L}}_{\text{Upload}} + \quad \underbrace{\frac{rt}{KL}}_{\text{Download}}$$

Goal: Given partition parameters K and L, and security parameter T, minimize the number of servers N.

Previous Work: Polynomial Codes for Stragglers

- Originally introduced in [Yu, Maddah-Ali, Avestimehr, ’17].
- Different Setting: mitigating stragglers
- Other Work: [Yu, Maddah-Ali, Avestimehr, ’18] ,
  [Dutta, Fahim, Haddadpour, Jeong, Cadambe, Grove, ’18],
  [Sheth, Dutta, Chaudhari, Jeong, Yang, Kohonen, Roos, Grove, ’18],
  [Li, Maddah-Ali, Yu, Avestimehr, ’18],
  etc.

Previous Work: Polynomial Codes for Security

- Distributed multiplication with information theoretic security.
- [Chang, Tandon, '18], [Kakar, Ebadifar, Sezgin, '18] and [Yang, Lee, '19]
- Related work: [Yu et al. '19], [Aliasgari et al. '19]

Can't Choose Any Polynomial

- Let $K = L = 3$ and $T = 2$.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}; \quad B = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}; \quad AB = \begin{bmatrix} A_1B_1 & A_1B_2 & A_1B_3 \\ A_2B_1 & A_2B_2 & A_2B_3 \\ A_3B_1 & A_3B_2 & A_3B_3 \end{bmatrix}$$

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
- $g(x) = B_1 + B_2x + B_3x^2 + S_1x^3 + S_2x^4$

Can't Choose Any Polynomial

- Let $K = L = 3$ and $T = 2$.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}; \quad B = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}; \quad AB = \begin{bmatrix} A_1B_1 & A_1B_2 & A_1B_3 \\ A_2B_1 & A_2B_2 & A_2B_3 \\ A_3B_1 & A_3B_2 & A_3B_3 \end{bmatrix}$$

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
- $g(x) = B_1 + B_2x + B_3x^2 + S_1x^3 + S_2x^4$
- Let $h(x) = f(x)g(x)$. Then,

$$h(x) = A_1B_1 + (A_1B_2 + A_2B_1)x + (A_1B_3 + A_2B_2 + A_3B_1)x^2 + \dots$$

Can't Choose Any Polynomial

- Let $K = L = 3$ and $T = 2$.

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}; \quad B = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}; \quad AB = \begin{bmatrix} A_1B_1 & A_1B_2 & A_1B_3 \\ A_2B_1 & A_2B_2 & A_2B_3 \\ A_3B_1 & A_3B_2 & A_3B_3 \end{bmatrix}$$

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
- $g(x) = B_1 + B_2x + B_3x^2 + S_1x^3 + S_2x^4$
- Let $h(x) = f(x)g(x)$. Then,

$$h(x) = A_1B_1 + (A_1B_2 + A_2B_1)x + (A_1B_3 + A_2B_2 + A_3B_1)x^2 + \dots$$

- Can't retrieve A_1B_2, for example.

It is not about the degree.

- **Scheme 1:**
  - $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
  - $g(x) = B_1 + B_2x^5 + B_3x^{10} + S_1x^{13} + S_2x^{14}$
  - $N_h = \deg h + 1 = 19$ servers.

It is not about the degree.

- **Scheme 1:**
  - $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
  - $g(x) = B_1 + B_2x^5 + B_3x^{10} + S_1x^{13} + S_2x^{14}$
  - $N_h = \deg h + 1 = 19$ servers.
- **Scheme 2:**
  - $f\ (x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
  - $g\ (x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$
- $\deg h\ \ = 22$

It is not about the degree.

- **Scheme 1:**
  - $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
  - $g(x) = B_1 + B_2x^5 + B_3x^{10} + S_1x^{13} + S_2x^{14}$
  - $N_h = \deg h + 1 = 19$ servers.
- **Scheme 2:**
  - $f\ \ (x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
  - $g\ \ (x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$
- $\deg h\ \ = 22 > 18 = \deg h$

It is not about the degree.

- **Scheme 1:**
  - $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
  - $g(x) = B_1 + B_2x^5 + B_3x^{10} + S_1x^{13} + S_2x^{14}$
  - $N_h = \deg h + 1 = 19$ servers.
- **Scheme 2:**
  - $f\ (x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
  - $g\ (x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$
- $\deg h\ = 22 > 18 = \deg h$
- But h has gaps in the degrees.
- No term of degrees $13; 14; 16; 17$ or 20.

It is not about the degree.

- **Scheme 1:**
  - $f(x) = A_1 + A_2x + A_3x^2 + R_1x^3 + R_2x^4$
  - $g(x) = B_1 + B_2x^5 + B_3x^{10} + S_1x^{13} + S_2x^{14}$
  - $N_h = \deg h + 1 = 19$ servers.
- **Scheme 2:**
  - $f\ (x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
  - $g\ (x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$
- $\deg h\ = 22 > 18 = \deg h$
- But h has gaps in the degrees.
- No term of degrees $13; 14; 16; 17$ or 20.
- Thus, only 18 points needed to interpolate h .
- $N_h\ = 18 < 19 = N_h$.

What is it about?

It is about the number of terms in the polynomial.

What is it about?

It is about the number of terms in the polynomial.

- Consider the polynomial $f(x) = ax^6 + bx^5 + cx$.
- We need $3 < \deg f + 1$ points to interpolate this polynomial.

What is it about?

It is about the number of terms in the polynomial.

- Consider the polynomial $f(x) = ax^6 + bx^5 + cx$.
- We need $3 < \deg f + 1$ points to interpolate this polynomial.

- Not any points! What does $f(0)$ tell you?

How many terms does $f(x)g(x)$ have?

- $f(x) = A_1x^{1} + A_2x^{2} + A_3x^{3} + R_1x^{4} + R_2x^{5}$
- $g(x) = B_1x^{1} + B_2x^{2} + B_3x^{3} + S_1x^{4} + S_2x^{5}$

The terms in $h(x)$ appear in the following table.

How many terms does $f(x)g(x)$ have?

- $f(x) = A_1 x^{1} + A_2 x^{2} + A_3 x^{3} + R_1 x^{4} + R_2 x^{5}$
- $g(x) = B_1 x^{1} + B_2 x^{2} + B_3 x^{3} + S_1 x^{4} + S_2 x^{5}$

The terms in $h(x)$ appear in the following table.

	1	2	3	4	5
1	1 + 1	1 + 2	1 + 3	1 + 4	1 + 5
2	2 + 1	2 + 2	2 + 3	2 + 4	2 + 5
3	3 + 1	3 + 2	3 + 3	3 + 4	3 + 5
4	4 + 1	4 + 2	4 + 3	4 + 4	4 + 5
5	5 + 1	5 + 2	5 + 3	5 + 4	5 + 5

How many terms does $f(x)g(x)$ have?

- $f(x) = A_1x^{1} + A_2x^{2} + A_3x^{3} + R_1x^{4} + R_2x^{5}$
- $g(x) = B_1x^{1} + B_2x^{2} + B_3x^{3} + S_1x^{4} + S_2x^{5}$

The terms in $h(x)$ appear in the following table.

	1	2	3	4	5
1	1 + 1	1 + 2	1 + 3	1 + 4	1 + 5
2	2 + 1	2 + 2	2 + 3	2 + 4	2 + 5
3	3 + 1	3 + 2	3 + 3	3 + 4	3 + 5
4	4 + 1	4 + 2	4 + 3	4 + 4	4 + 5
5	5 + 1	5 + 2	5 + 3	5 + 4	5 + 5

- We call this a degree table.

Properties of the Degree Table

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
- $g(x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$

h	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
12	12	15	18	21	22

Properties of the Degree Table

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
- $g(x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$

h	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
12	12	15	18	21	22

- Decodability: Red cells unique.

Properties of the Degree Table

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
- $g(x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$

h	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
12	12	15	18	21	22

- Decodability: Red cells unique.
- Security A: Green cells distinct.

Properties of the Degree Table

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
- $g(x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$

h	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
12	12	15	18	21	22

- Decodability: Red cells unique.
- Security A: Green cells distinct.
- Security B: Blue cells distinct.

Properties of the Degree Table

- $f(x) = A_1 + A_2x + A_3x^2 + R_1x^9 + R_2x^{12}$
- $g(x) = B_1 + B_2x^3 + B_3x^6 + S_1x^9 + S_2x^{10}$

h	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
12	12	15	18	21	22

- Decodability: Red cells unique.
- Security A: Green cells distinct.
- Security B: Blue cells distinct.

- **Goal:** Minimize distinct cells.

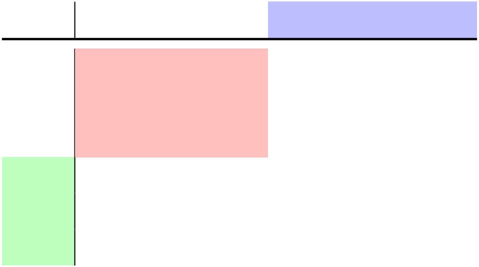
Problem Restatement: The Degree Table

	1		L	$L+1$		$L+T$
1	$1 + 1$		$1 + L$	$1 + L+1$		$1 + L+T$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
K	$K + 1$		$K + L$	$K + L+1$		$K + L+T$
$K+1$	$K+1 + 1$		$K+1 + L$	$K+1 + L+1$		$K+1 + L+T$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$K+T$	$K+T + 1$		$K+T + L$	$K+T + L+1$		$K+T + L+T$

- **Goal:** Minimize number of distinct terms.
- Subject to:
  - Decodability: Numbers in the red region are all unique.
  - *A*-Security: Numbers in the green region are all distinct.
  - *B*-Security: Numbers in the blue region are all distinct.

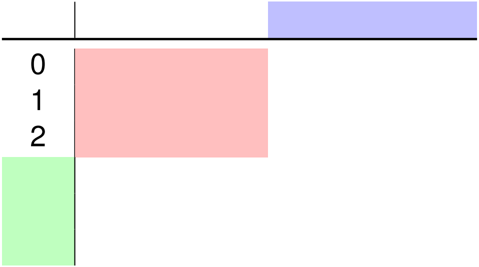
GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$



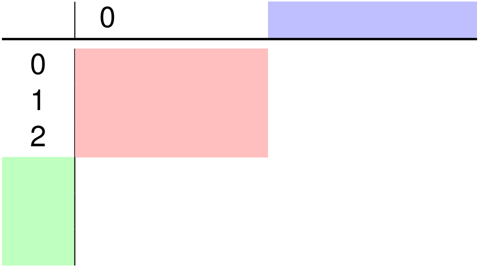
GASP_{big} [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$



GASP$_{big}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$



GASP_{big} [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	
0	0	
1		
2		

GASP$_{big}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	
0	0	
1	1	
2		

GASP$_{big}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	
0	0	
1	1	
2	2	

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3
0	0	
1	1	
2	2	

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	
0	0	3	
1	1	4	
2	2	5	

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	
0	0	3		
1	1	4		
2	2	5		

GASP_{big} [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	
0	0	3	6	
1	1	4	7	
2	2	5	8	

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6
0	0	3	6
1	1	4	7
2	2	5	8
9			

GASP$_{big}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9
0	0	3	6	
1	1	4	7	
2	2	5	8	
9				

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9
0	0	3	6	9
1	1	4	7	10
2	2	5	8	11
9	9	12	15	18

GASP_{big} [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10
0	0	3	6	9	
1	1	4	7	10	
2	2	5	8	11	
9	9	12	15	18	
10					

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10
0	0	3	6	9	10
1	1	4	7	10	11
2	2	5	8	11	12
9	9	12	15	18	19
10	10	13	16	19	20

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	
1	1	4	7	10	11	
2	2	5	8	11	12	
9	9	12	15	18	19	
10	10	13	16	19	20	
11						

GASP$_{\text{big}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

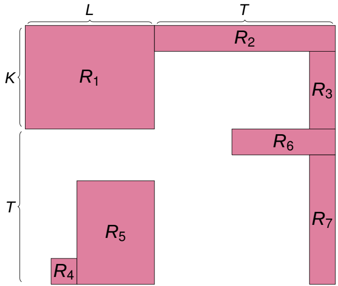
	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9	9	12	15	18	19	20
10	10	13	16	19	20	21
11	11	14	17	20	21	22

Number of Terms

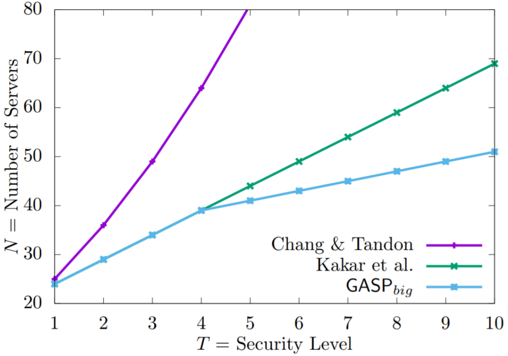
Theorem [D'Oliveira, SER, Karpuk, ISIT '19]

The number of terms in $\mathrm{GASP}_{\mathrm{big}}$, for $L \leq K$, is

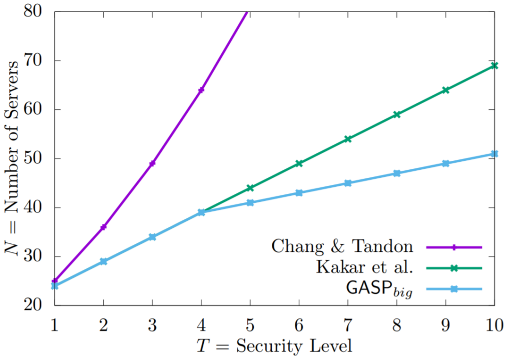
$$N = \begin{cases} 2K + 2T - 1 & \text{if } L = 1 \\ (K+T)(L+1) - 1 & \text{if } L \geq 2;\ T < K \\ 2KL + 2T - 1 & \text{if } L \geq 2;\ T \geq K \end{cases}$$



How good is GASP$_{\text{big}}$?

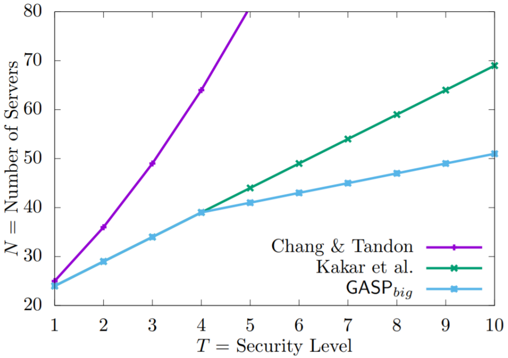


How good is GASP$_{\text{big}}$?



- Lagrange coding [Yu et al.,'19] achieves same rate for $T \quad \min\{K; L\}$.

How good is GASP$_{\text{big}}$?



- Lagrange coding [Yu et al.,'19] achieves same rate for $T \geq \min\{K, L\}$.
- Can we do better?

GASP$_{\text{small}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13

GASP$_{\text{small}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9						

GASP$_{small}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9						
12						

GASP$_{\text{small}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9						
12						
15						

GASP$_{\text{small}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9	9	12	15			
12	12	15	18			
15	15	18	21			

$\text{GASP}_{\text{small}}$ [D'Oliveira, SER, Karpuk, ISIT '19]

$K = L = T = 3$

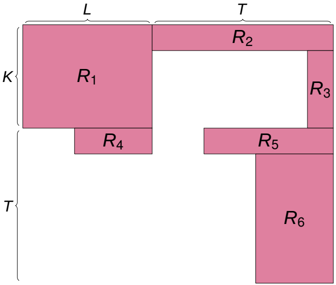
	0	3	6	9	10	11
0	0	3	6	9	10	11
1	1	4	7	10	11	12
2	2	5	8	11	12	13
9	9	12	15	18	19	20
12	12	15	18	21	22	23
15	15	18	21	24	25	26

Number of Terms

Theorem [D'Oliveira, SER, Karpuk, ISIT '19]

The number of terms in $\text{GASP}_{\text{small}}$, for $K \geq L$, is

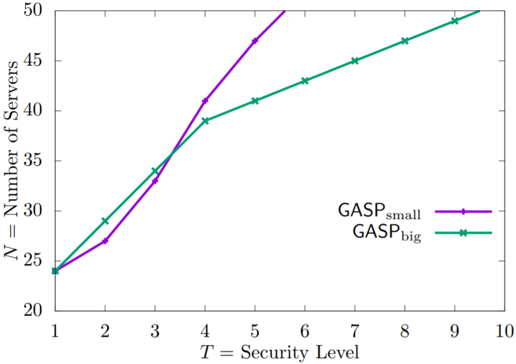
$$N = \begin{cases} 2K + T^2 & \text{if } L = 1; T < K \\ KT + K + T & \text{if } L = 1; T \geq K \\ KL + K + L & \text{if } L \geq 2; 1 = T < K \\ KL + K + L + T^2 + T - 3 & \text{if } L \geq 2; 2 \leq T < K \\ KL + KT + L + 2T - 3 - \left\lfloor \frac{T-2}{K} \right\rfloor & \text{if } L \geq 2; K \leq T \leq K(L-1)+1 \\ 2KL + KT - K + T & \text{if } L \geq 2; K(L-1)+1 < T \end{cases}$$



What is small T?

Theorem [D'Oliveira, SER, Karpuk, ISIT '19]

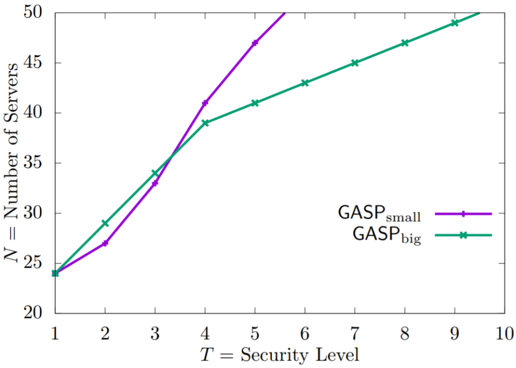
$\mathrm{GASP}_{\mathrm{small}}$ outperforms $\mathrm{GASP}_{\mathrm{big}}$ for $T < \min\{K, L\}$.



What is small T?

Theorem [D'Oliveira, SER, Karpuk, ISIT '19]

$\text{GASP}_{\text{small}}$ outperforms GASP_{big} for $T < \min\{K, L\}$.



- Can we do better?

GASP$_r$: Gap Additive Secure Polynomial codes

	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

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

	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

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

	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

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

	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

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

Theorem [D'Oliveira, SER, Heinlein, Karpuk, ITW '19]

- Partitioning parameters: K and L
- Security parameter: T
- Chain length: r

Then, the degree table constructed by GASP$_r$ has

$$N = KL + K + T \quad 1 + T \quad (L + T) \quad S(r)$$

Theorem [D'Oliveira, SER, Heinlein, Karpuk, ITW '19]

- Partitioning parameters: K and L
- Security parameter: T
- Chain length: r

Then, the degree table constructed by GASP$_r$ has

$$N = KL + K + T \quad 1 + T \quad (L + T) \quad S(r)$$

where

$$S(r) = \max f0; \min fr; ' ggL + 2 \max f0; r \quad z + 1g + \quad + (T \quad r)L + \max f0; K + T \quad KL \quad 1g +$$
$$+ \quad \max f0; T \quad K + r \quad 1g + (T \quad 1 \quad)(T \quad 1)$$

$$' = T \quad 1 \quad KL + 2K; \qquad = b(T \quad 1)=rc; \quad z = \max f1; ' + 1g;$$

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

with $a; b; x; y$ defined by

$$T \quad 1 \quad r = aK + b \text{ and } 0 \quad b \quad K \quad 1;$$
$$T \quad 1 \quad z = xK + y \text{ and } 0 \quad y \quad K \quad 1:$$

Lower Bounds

Theorem [D'Oliveira, SER, Heinlein, Karpuk, ITW '19]

- Partitioning parameters: K and L
- Security parameter: T
- Number of distinct terms: N

Then the following three inequalities hold.

1. $KL + \max\{K; L\} + 2T - 1 \leq N$.

Lower Bounds

Theorem [D'Oliveira, SER, Heinlein, Karpuk, ITW '19]

- Partitioning parameters: K and L
- Security parameter: T
- Number of distinct terms: N

Then the following three inequalities hold.

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

Lower Bounds

Theorem [D'Oliveira, SER, Heinlein, Karpuk, ITW '19]

- Partitioning parameters: K and L
- Security parameter: T
- Number of distinct terms: N

Then the following three inequalities hold.

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

Main Idea Behind Lower Bound

- Result from additive combinatorics on the minimal size of sum sets.

Main Idea Behind Lower Bound

- Result from additive combinatorics on the minimal size of sum sets.

Lemma [Tao,Vu, “Additive Combinatorics”]

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.

Main Idea Behind Lower Bound

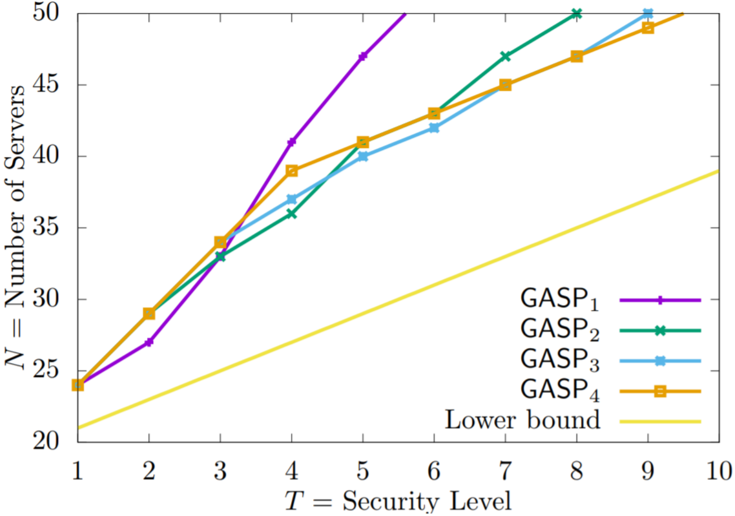
- Result from additive combinatorics on the minimal size of sum sets.

Lemma [Tao,Vu, “Additive Combinatorics”]

Let A and B be sets of integers. Then $|A| + |B| - 1 \leq |A + B|$ and if $2 \leq |A|, |B|$, then equality holds iff A and B are arithmetic progressions with the same common difference.

Current Situation

- $K = L = 4$
- GASP_r for $r = 1, \ldots, K$.



Optimality*

Corollary

If either $K = 1$, $L = 1$, or $T = 1$, then GASP_r is optimal.

Optimality*

Corollary

If either $K = 1$, $L = 1$, or $T = 1$, then GASP_r is optimal.

Corollary

If $K = L = T = n^2 \quad 4$, then GASP_n is asymptotically optimal.

Optimality*

Corollary

If either $K = 1$, $L = 1$, or $T = 1$, then GASP_r is optimal.

Corollary

If $K = L = T = n^2 \quad 4$, then GASP_n is asymptotically optimal.

Is it all worth it?

- $r = s = t = n$ (square matrices).

Is it all worth it?

- $r = s = t = n$ (square matrices).
- Security parameter T is constant.

Is it all worth it?

- $r = s = t = n$ (square matrices).
- Security parameter T is constant.
- Servers multiply two n n in time $O(n^{!})$.

Is it all worth it?

- $r = s = t = n$ (square matrices).
- Security parameter T is constant.
- Servers multiply two $n \quad n$ in time $O(n^{!})$.
- Partitioning parameters $K = L = n^{''}$.

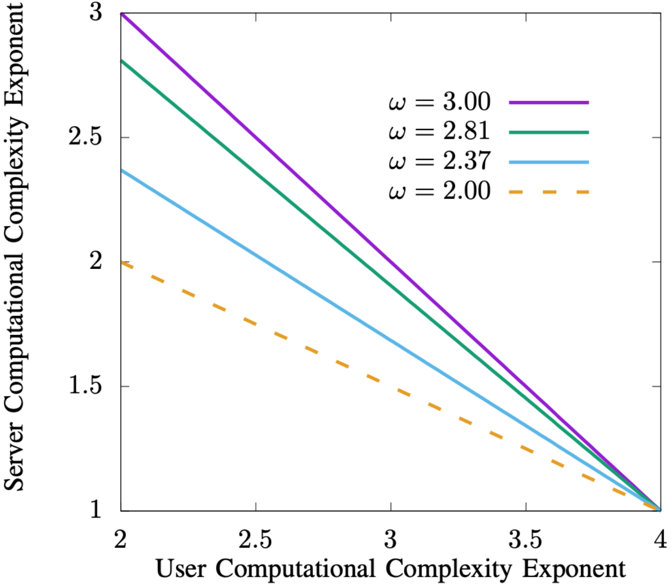
Is it all worth it?

- $r = s = t = n$ (square matrices).
- Security parameter T is constant.
- Servers multiply two $n \times n$ in time $O(n^{\omega})$.
- Partitioning parameters $K = L = n^{\epsilon}$.

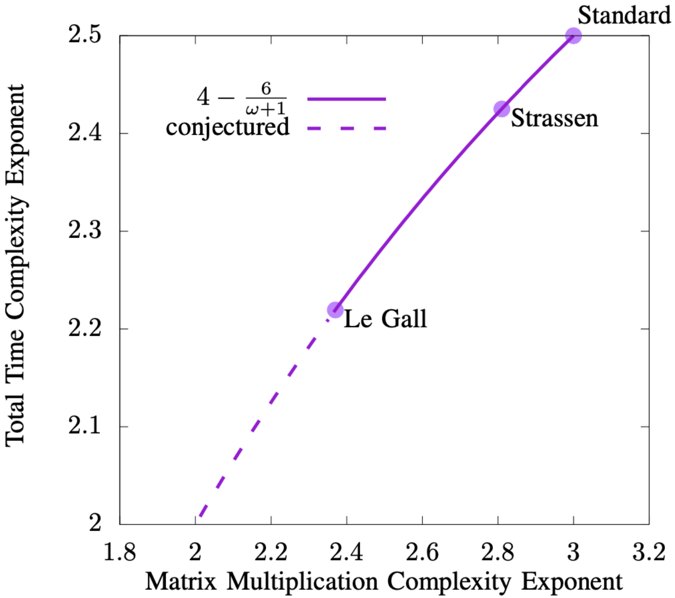
Theorem [D'Oliveira, SER, Heinlein, Karpuk '20]

By using GASP, the user can perform the matrix multiplication in time $O(n^{4-\frac{6}{\omega+1}}\log(n)^2)$ as opposed to the $O(n^{\omega})$ time it would take to do locally.

Is it all worth it?



Is it all worth it?



Open Problems

- Are there better schemes for the degree table?
- Are there better bounds?
- What about information theoretical bounds?
- Are polynomial codes optimal?

Thanks!