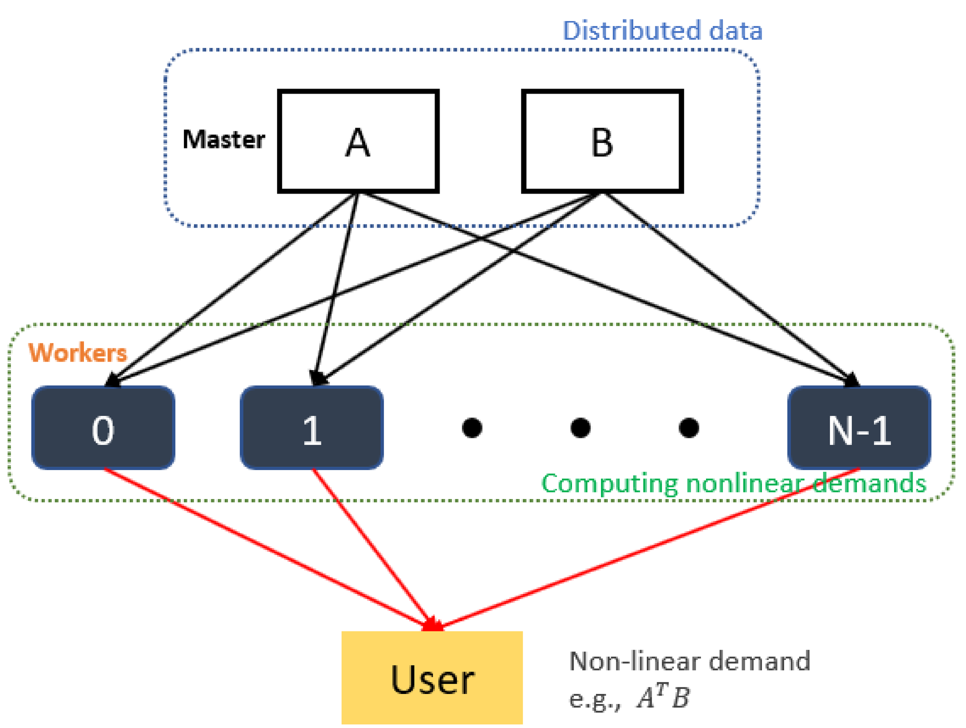
Structured Polynomial Codes

Mohammad Reza Deylam Salehi, Ahmad Tanha, Derya Malak

Communication Systems Department, EURECOM, Biot, Sophia Antipolis, France.

Emails: {deylam, tanha, malak}@eurecom.fr

Motivation



- A fundamental challenge: Balancing computation and communication complexity.

Related work

Distributed computing frameworks:

- MapReduce, Hadoop, Spark, TeraSort [1]

Channel coding approaches:

- Polynomial codes, Lagrange coded computing [2, 3]

Source coding approaches:

- Structured codes for modulo two sum computation in [4], and distributed matrix multiplication in [5]

Contributions

Novelty:

- Combining the benefits of structured coding and polynomial codes
- Elevating the Körner–Marton approach to the distributed matrix multiplication setting
- Incorporating a secure matrix multiplication design

Savings:

- Low complexity distributed encoding
- Communication costs (reduced by %50)
- Storage size (reduced by %50)

Future directions

Structured codes for

- n-matrix products
- privacy/security aspects
- tensor product computations

References

[1] Alkatheri et al. A comparative study of big data frameworks. *Int. Jour. Comp. Sci. IJCSIS*, 2019.

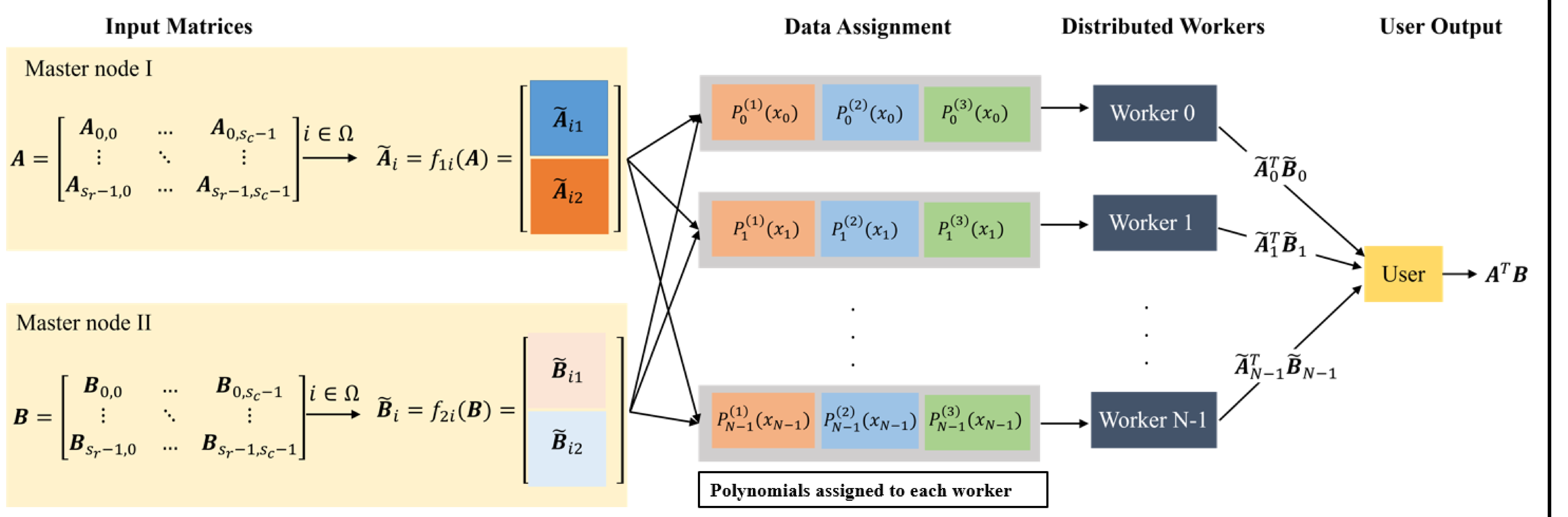
[2] López et al. Secure MatDot codes: a secure, distributed matrix multiplication scheme. In *ITW 2022*, Mumbai, India, 2022.

[3] Yu et al. Lagrange coded computing: Optimal design for resiliency, security, and privacy. In *Proc. Int. on AI and Stat.*, 2019.

[4] Körner and Marton. How to encode the modulo-two sum of binary sources. *IEEE Trans. Inf. Theory*, 1979.

[5] Malak. Distributed structured matrix multiplication. In *IEEE ISIT*, Athens, Greece, Jul. 2024.

A structured distributed matrix multiplication model



- ☐ *Each worker, using the assigned polynomials, calculates the product of sub-matrices* $\widetilde{\mathbf{A}}_i^\intercal\widetilde{\mathbf{B}}_i$.
- ☐ *Using* $\{\widetilde{\mathbf{A}}_i^\intercal\widetilde{\mathbf{B}}_i\}_i$ *from a subset of workers, the user decodes* $\mathbf{AB}$.
- ☐ *The user cannot decode* $\mathbf{A}$ *or* $\mathbf{B}$*, where the security of matrix multiplication is ensured by structured coding.*

Source coding for matrix multiplication [5]

Two *distributed sources*, $\mathbf{A} \in \mathbb{F}_q^{m\times 1}$ and $\mathbf{B} \in \mathbb{F}_q^{m\times 1}$:

- **Splitting of each source:**

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}^\intercal \in \mathbb{F}_q^{m\times 1}, \qquad \mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix} \in \mathbb{F}_q^{m\times 1}.$$

- **Nonlinear mapping from each source:**

$$\mathbf{X}_1 = g_1(\mathbf{A}) = \begin{bmatrix}\mathbf{A}_2\\ \mathbf{A}_1\\ \mathbf{A}_2^\intercal\mathbf{A}_1\end{bmatrix} \in \mathbb{F}_2^{(m+1)\times 1}, \qquad \mathbf{X}_2 = g_2(\mathbf{B}) = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\\ \mathbf{B}_1^\intercal\mathbf{B}_2\end{bmatrix} \in \mathbb{F}_2^{(m+1)\times 1}.$$

- **Linear encoding:** Sources use a common encoder, compute and send $\mathbf{C}\mathbf{X}_j^n \in \mathbb{F}_2^{(m+1)\times k}$ [4].
- **Decoding:** Exploiting [4], the sum rate needed for the user to recover the vector sequence

$$\mathbf{Z}^n = \mathbf{X}_1^n \oplus_2 \mathbf{X}_2^n \in \mathbb{F}_2^{(m+1)\times n}$$

with a vanishing error probability is determined as

$$R_{\text{KM}}^{\Sigma} = 2H(\mathbf{X}_1 \oplus_2 \mathbf{X}_2) = 2H(\mathbf{U}, \mathbf{V}, W),$$

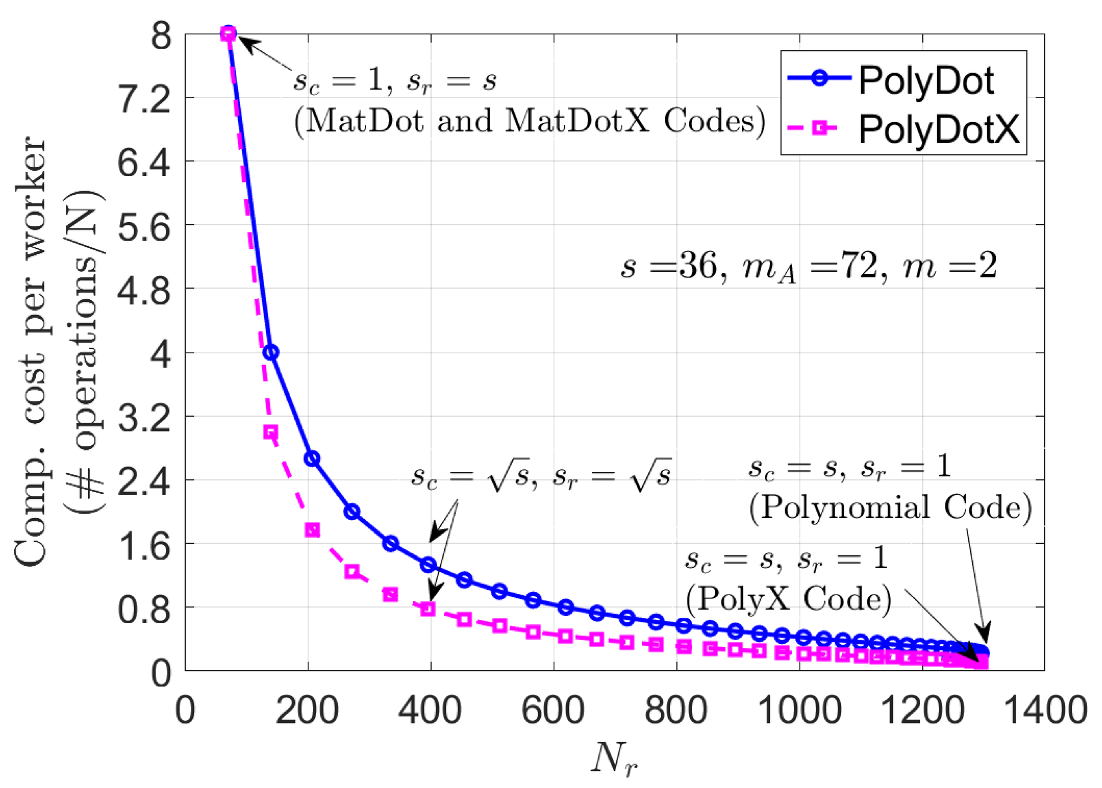
where the following vectors can be computed in a fully distributed manner:

$$\mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1 \in \mathbb{F}_q^{m/2\times 1}, \qquad \mathbf{V} = \mathbf{A}_1 \oplus_q \mathbf{B}_2 \in \mathbb{F}_q^{m/2\times 1}, \qquad W = \mathbf{A}_2^\intercal\mathbf{A}_1 \oplus_q \mathbf{B}_1^\intercal\mathbf{B}_2 \in \mathbb{F}_q.$$

The user can recover the desired inner product using $\mathbf{U}$, $\mathbf{V}$, and W.

Performance results

For $s_c \gg m$, the upper bound of computation cost per worker approaches $1 + \frac{1}{2s}$.



The total communication cost is reduced by %50 compared to the PolyDot model.