Cooperative Coded Matrix Multiplication in Secrecy-Constrained Vehicular Networks

Ahmad Tanha, Mohammad Reza Deylam Salehi, Monolina Dutta, and Derya Malak
Communication Systems Department, EURECOM, Sophia Antipolis, France
{ahmad.tanha, reza.deylam-salehi, monolina.dutta, derya.malak}@eurecom.fr

***Abstract*—We propose a new class of polynomial codes, distributed structured PolyDot codes (DSPolyDot codes), for cooperative computation in vehicular edge computing networks, where computational tasks are abstracted by a matrix multiplication problem $\mathbf{A}^\intercal\mathbf{B}$, where $\mathbf{A}$ captures input data (features) and $\mathbf{B}$ the model parameters (weights), with entries over $\mathbb{F}_q$, where the matrices are stored in separate source nodes, e.g., the central cloud network and the base station. In cooperative vehicular networks, source data must be processed across distributed vehicles and roadside units that may not be fully trusted, motivating the need for information-theoretic secrecy in terms of bounded information leakage at each cooperative node. Leveraging the algebraic structure of matrix multiplication, we design a coded computation framework that ensures information-theoretic secrecy with bounded information leakage at each cooperative node while naturally tolerating stragglers and achieving the same communication and computation efficiency as state-of-the-art polynomial codes in the asymptotic regime. These make DSPolyDot codes well-suited for secrecy-constrained cooperative signal processing applications in vehicular wireless networks.**

***Index Terms*—Storage-communication-computation trade-offs, distributed matrix multiplication, vehicular networks, secrecy.**

I. Introduction

Modern vehicular communication systems increasingly rely on computationally intensive signal processing tasks such as edge and cloud computing [1]–[3], real-time control for connected and autonomous vehicles [4], sensing and perception [5], internet of vehicles [6], estimation and learning [7]–[11], and cooperative congestion control [12]. These tasks are heavily based on large-scale matrix computations that must be performed under stringent latency and reliability constraints, while operating over wireless links subject to mobility, fading, and intermittent connectivity. To meet these requirements, vehicular networks are increasingly adopting cooperative edge computing architectures, in which processing tasks are offloaded from the source nodes, such as the central cloud network and base stations, to nearby cooperative vehicle nodes, e.g., sensing cars, as illustrated in Figure 1. The resulting intermediate computations are then transmitted to the mobile end-user, e.g., the navigating car.

In vehicular networks, lacking direct access to source nodes necessitates distributing computational tasks across intermediate nodes, which are cooperative vehicles. In such dynamic wireless environments, mobility and channel variability introduce uncertainty in task offloading and result aggregation, leading to varying response times among cooperative vehicles and potential data confidentiality risks. Consequently, conventional distributed signal processing schemes become susceptible to mobility-induced stragglers and information leakage during the computation process at the intermediate nodes. To satisfy stringent secrecy requirements by ensuring that no vehicle's local observation uniquely identifies the underlying source matrices and also to alleviate end-to-end computational latency at the vehicle nodes, the encoded data from source nodes is transmitted to the vehicles for parallel processing, and the end-users then aggregate the resulting intermediate computations.

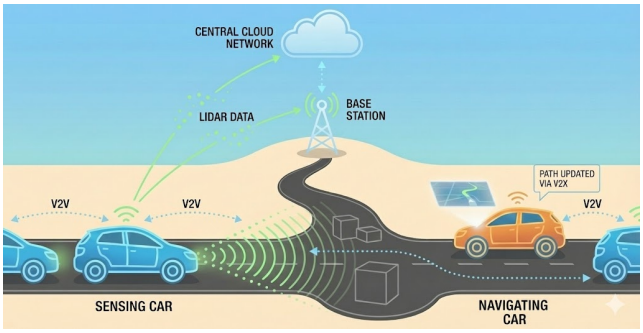

Fig. 1: A task-orchestrated vehicular edge computing architecture.

To mitigate the end-to-end computational latency caused by straggling nodes, several coded computation frameworks have been devised, such as convolutional coding [13], and more specifically, polynomial and Lagrange coding techniques for distributed matrix multiplication (DMM) [14]–[19]. Existing solutions primarily rely on linear encoding schemes, such as polynomial (Poly) codes [14], [16], [20] and PolyDot codes [19]. These methods reduce the computational complexity of DMM by exploiting linear polynomial constructions. However, these approaches are restricted to linear transformations of the source matrices. In contrast, incorporating structural or non-linear transformations that go beyond linear coding schemes can offer robustness against untrusted nodes [21], [22].

In this paper, by drawing on a recent source coding scheme for non-linear function computation, particularly bilinear functions, including dot products and matrix products [23], [24],

This research was partially supported by European Research Council ERC-StG Project SENSIBILITÉ under Grant 101077361, by the Huawei France-Funded Chair Toward Future Wireless Networks, and by the Program "PEPR Networks of the Future" of France 2030.

and combining it with the PolyDot code framework, we design novel *distributed structured PolyDot* codes — referred to as DSPolyDot codes — for DMM.

Our main contributions are summarized as follows.

- *Integration of structured source coding into PolyDot codes*: We consider the distributed computing of general matrix products, where two source nodes generate length-n sequences of matrices $\{\mathbf{A}_i\}_{i\in[n]}$ and $\{\mathbf{B}_i\}_{i\in[n]}$ over finite fields, where $[n] \triangleq \{1,\dots,n\}$. The matrices are independently and identically distributed (i.i.d.) across n realizations, according to some joint probability distribution. Given $i \in [n]$, $\mathbf{A}_i$ and $\mathbf{B}_i$ may be correlated, reflecting statistical or structural dependence arising from shared sensing data, common model parameters, or temporally related observations in vehicular applications. However, there is no correlation between $\mathbf{A}_i$ and $\mathbf{A}_j$, and between $\mathbf{A}_i$ and $\mathbf{B}_j$ for $j \neq i$. This formulation enables the analysis of cooperative coded computation schemes. Building on [23], which employs the design of non-linear transformations of the source matrices $\mathbf{A}_i$ and $\mathbf{B}_i$, and by incorporating PolyDot codes [19], we devise novel DSPolyDot codes for DMM in a practical task-orchestrated vehicular edge computing model detailed in Section II, where each cooperative vehicle node has bounded storage reflecting a bounded computational capability.
- *Structural information-theoretic secrecy guarantees*: DSPolyDot codes leverage the computation structure of the two given source matrices $\mathbf{A}_i$ and $\mathbf{B}_i$, yielding structural secrecy with only bounded information leakage at the untrusted vehicle nodes, arising directly from our structured coding scheme (cf. Proposition 1). Moreover, it enables the navigating car (receiver) to reconstruct the desired matrix product $\mathbf{A}_i^\intercal\mathbf{B}_i$ without revealing the source matrices in their entirety and without imposing structural constraints on $\mathbf{A}_i$ and $\mathbf{B}_i$. On large scales, our model offers secrecy without incurring additional communication or computation costs compared to PolyDot codes [19].
- *Distributed and practical implementation*: DSPolyDot codes utilize the *distributed design* of the structured linear encoding scheme of [25], thus, naturally support *distributed implementations*, e.g., [3], [26] by separate non-linear operations on $\mathbf{A}_i$ and $\mathbf{B}_i$ directly at two distributed source nodes, each containing one matrix.

Organization. The rest of the paper is organized as follows. Section II outlines the DMM model and the construction details of the proposed DSPolyDot codes. The main results are then presented in Section III. Before concluding the paper in Section V, the numerical analysis is detailed in Section IV.

II. DSPolyDot Codes

We consider a distributed computing system with two distributed source nodes, N cooperative vehicles, and a receiver, as demonstrated in Figure 2. We assume large sequences of two source matrices $\{\mathbf{A}_i\}_{i\in[n]}, \{\mathbf{B}_i\}_{i\in[n]}$, where $\mathbf{A}_i, \mathbf{B}_i \in \mathbb{F}_q^{m_A\times m}$, $q \geq 2$, stored in source nodes I and II, respectively, where $i \in [n]$ represents the i-th realization. Each vehicle will construct a set of polynomials, utilizing Körner-Marton encoding [25], as will be detailed later on in this section, on the pre-processed subfunctions of submatrices of $\mathbf{A}_i$ and $\mathbf{B}_i$, transmitted by the source nodes. Exploiting a set of N distributed vehicles, each computing a function of the set of Körner-Marton encoded polynomials, the receiver aims to compute $\mathbf{A}_i^\intercal\mathbf{B}_i$ for each realization $i \in [n]$, where $\mathbf{A}_i^\intercal$ denotes the transpose of $\mathbf{A}_i$.

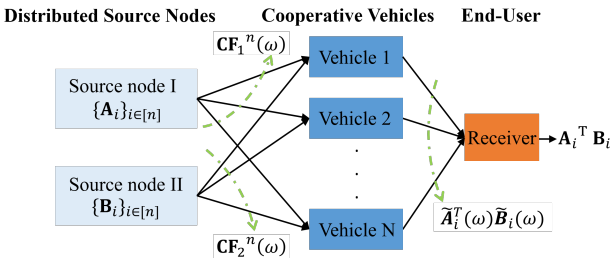

Fig. 2: The DMM framework considered in the current work. The length-n vectors $F_1^n(\omega), F_2^n(\omega)$ are encoded using a common linear encoding matrix $\mathbf{C}$, based on Körner-Marton encoding [25]. Worker $\omega \in \Omega$ then builds a set of polynomials $\mathbf{p}_i(\omega)$ with Körner-Marton decoding of the transmitted vectors, computes the desired output subfunction, and sends it to the receiver to recover the demanded $\mathbf{A}_i^\intercal\mathbf{B}_i$ for the i-th realization with polynomial interpolation.

We next detail the construction of our proposed DSPolyDot codes for DMM. The steps are implemented for each realization $i \in [n]$ whenever applicable.

a) Splitting source matrices: Source node I splits $\mathbf{A}_i$ into submatrices $\alpha_i(j,k)$ and source node II splits $\mathbf{B}_i$ into submatrices $\beta_i(j,k)$ for $j \in S_r$, $k \in S_c$, assuming that $S_r \triangleq \{0,\dots,s_r-1\}$, $S_c \triangleq \{0,\dots,s_c-1\}$, where s_r divides m_A and s_c divides m. Matrix $\mathbf{A}_i$ is then described as

$$\mathbf{A}_i \triangleq \begin{bmatrix} \alpha_i(0,0) & \dots & \alpha_i(0,s_c-1) \\ \vdots & \ddots & \vdots \\ \alpha_i(s_r-1,0) & \dots & \alpha_i(s_r-1,s_c-1) \end{bmatrix}, \quad (1)$$

where $\alpha_i(j,k) \in \mathbb{F}_q^{\frac{m_A}{s_r}\times\frac{m}{s_c}}$ and similarly $\beta_i(j,k) \in \mathbb{F}_q^{\frac{m_A}{s_r}\times\frac{m}{s_c}}$ for matrix $\mathbf{B}_i$, where $q \geq 2$.

b) Polynomial construction: Given the block representation in (1), to determine $\mathbf{A}_i^\intercal\mathbf{B}_i \in \mathbb{F}_q^{m\times m}$, inspired by PolyDot codes in [19], we define the following linear pre-processing functions at the source nodes for different coefficients x_ω across the vehicles $\omega \in \Omega \triangleq \{1,\cdots,N\}$.

$$\tilde{\mathbf{A}}_i(\omega) \triangleq \sum_{j=0}^{s_c-1}\sum_{k=0}^{s_r-1} \alpha_i(j,k)x_\omega^j x_\omega^{s_c k} \in \mathbb{F}_q^{\frac{m_A}{s_r}\times\frac{m}{s_c}}, \quad (2)$$

$$\tilde{\mathbf{B}}_i(\omega) \triangleq \sum_{j=0}^{s_r-1}\sum_{k=0}^{s_c-1} \beta_i(j,k)x_\omega^{s_c(s_r-1-j)} x_\omega^{s_c(2s_r-1)k} \in \mathbb{F}_q^{\frac{m_A}{s_r}\times\frac{m}{s_c}}.$$

To devise non-linear mappings of the constructed polynomials in (2) for DSPolyDot codes, each source node uses the following row-block representations to construct the cor-

responding submatrices of the polynomial evaluations:

$$\tilde{\mathbf{A}}_i(\omega) = \begin{bmatrix} \tilde{\mathbf{A}}_{i1}(\omega) \\ \tilde{\mathbf{A}}_{i2}(\omega) \end{bmatrix}, \quad \tilde{\mathbf{B}}_i(\omega) = \begin{bmatrix} \tilde{\mathbf{B}}_{i1}(\omega) \\ \tilde{\mathbf{B}}_{i2}(\omega) \end{bmatrix}, \quad \omega \in \Omega, \quad (3)$$

where $\tilde{\mathbf{A}}_{i1}(\omega), \tilde{\mathbf{A}}_{i2}(\omega), \tilde{\mathbf{B}}_{i1}(\omega), \tilde{\mathbf{B}}_{i2}(\omega) \in \mathbb{F}_q^{\frac{m_A}{2s_r} \times \frac{m}{s_c}}$.

c) Non-linear pre-processing: To devise novel polynomial codes that capture the computation structure for evaluating $\mathbf{A}_i^\intercal \mathbf{B}_i$, and following [23], source nodes I and II construct non-linear mappings $f_{1i}(\omega), f_{2i}(\omega) \in \mathbb{F}_q^{(\frac{m_A}{s_r} + \frac{m}{s_c}) \times \frac{m}{s_c}}$ of $\tilde{\mathbf{A}}_i(\omega)$ and $\tilde{\mathbf{B}}_i(\omega)$, for each $i \in [n]$, $\omega \in \Omega$, respectively as

$$f_{1i}(\omega) \triangleq \begin{bmatrix} \tilde{\mathbf{A}}_{i2}(\omega) \\ \tilde{\mathbf{A}}_{i1}(\omega) \\ \tilde{\mathbf{A}}_{i2}^\intercal(\omega)\tilde{\mathbf{A}}_{i1}(\omega) \end{bmatrix}, \; f_{2i}(\omega) \triangleq \begin{bmatrix} \tilde{\mathbf{B}}_{i1}(\omega) \\ \tilde{\mathbf{B}}_{i2}(\omega) \\ \tilde{\mathbf{B}}_{i1}^\intercal(\omega)\tilde{\mathbf{B}}_{i2}(\omega) \end{bmatrix}. \quad (4)$$

Each source node then reshapes the non-linear mappings $f_{1i}(\omega), f_{2i}(\omega)$, as the vector representations of the form

$$F_{1i}(\omega) \triangleq \big[f_{1i}(:,1) \; f_{1i}(:,2) \; \ldots \; f_{1i}(:,\tfrac{m}{s_c})\big]^\intercal,$$
$$F_{2i}(\omega) \triangleq \big[f_{2i}(:,1) \; f_{2i}(:,2) \; \ldots \; f_{2i}(:,\tfrac{m}{s_c})\big]^\intercal,$$

where $F_{1i}(\omega), F_{2i}(\omega) \in \mathbb{F}_q^{(\frac{m_A}{s_r} + \frac{m}{s_c})\frac{m}{s_c} \times 1}$.

The source nodes concatenate the above-mentioned vectors to build new vectors

$$F_1^n(\omega) \triangleq [F_{11}, F_{12}, \ldots, F_{1n}]^\intercal \in \mathbb{F}_q^{n(\frac{m_A}{s_r} + \frac{m}{s_c})\frac{m}{s_c} \times 1},$$
$$F_2^n(\omega) \triangleq [F_{21}, F_{22}, \ldots, F_{2n}]^\intercal \in \mathbb{F}_q^{n(\frac{m_A}{s_r} + \frac{m}{s_c})\frac{m}{s_c} \times 1}. \quad (5)$$

d) Körner-Marton encoding: Given two length-n source sequences X^n and Y^n, the Körner-Marton approach addresses the rate region for computing the sum of binary sources in the i.i.d. setting, provided that the joint distribution of sources is symmetric [25]. In this framework, both encoders apply the same linear mapping $\mathbf{C} \in \mathbb{F}_2^{k \times n}$ to their respective source sequences, and the decoder reconstructs the sum $Z^n = X^n \oplus Y^n$ by exploiting the linearity of the encoding. As shown in [25, Theorem 1], there exist such linear encoders each with rate $\frac{k}{n}$ arbitrarily close to $H(Z)$ for which Z^n can be recovered with small probability of error as $n \to \infty$, where $H(Z)$ denotes the entropy of the binary random variable $Z = X \oplus Y$. Extensions of the Körner-Marton framework to finite fields $\mathbb{F}_q$ with $q > 2$ can be found in, e.g., [23], [27].

Each source node then encodes vectors $F_1^n(\omega), F_2^n(\omega)$, given in (5), with a common linear encoder $\mathbf{C} \in \mathbb{F}_q^{k_n \times n'}$, where dimensions k_n and n' are expressed as

$$k_n \triangleq \max_{\omega \in \Omega} H_q\big(F_1^n(\omega) \oplus_q F_2^n(\omega)\big),$$
$$n' \triangleq n\big(\frac{m_A}{s_r} + \frac{m}{s_c}\big)\frac{m}{s_c}.$$

Each source node then transmits the Körner-Marton encoded data $\mathbf{C}F_1^n(\omega), \mathbf{C}F_2^n(\omega) \in \mathbb{F}_q^{k_n \times 1}$ to each vehicle $\omega \in \Omega$ at an operating rate of $\frac{k_n}{n'}$.

e) Körner–Marton decoding: Worker ω first recovers the modulo-q sum $F_1^n(\omega) \oplus_q F_2^n(\omega)$ by applying Körner–Marton decoding to the received sequence $\mathbf{C}F_1^n(\omega) \oplus_q \mathbf{C}F_2^n(\omega)$. The validity of this decoding follows from a generalization of Elias' lemma [28], which guarantees reliable recovery of $F_1^n(\omega) \oplus_q F_2^n(\omega)$ whenever $\frac{k_n}{n'} > H_q(F_1^n(\omega) \oplus_q F_2^n(\omega))$. Consequently, the decoder can reconstruct $F_{1i}(\omega) \oplus_q F_{2i}(\omega)$ in an asymptotically lossless manner. Next, from the recovered sum $F_{1i}(\omega) \oplus_q F_{2i}(\omega)$, vehicle ω computes the corresponding polynomial outputs $\mathbf{p}_i(\omega) \triangleq \{p_i^{(1)}(\omega), p_i^{(2)}(\omega), p_i^{(3)}(\omega)\}$, given by

$$p_i^{(1)}(\omega) \triangleq \tilde{\mathbf{A}}_{i2}(\omega) \oplus_q \tilde{\mathbf{B}}_{i1}(\omega) \in \mathbb{F}_q^{\frac{m_A}{2s_r} \times \frac{m}{s_c}}, \quad (6)$$
$$p_i^{(2)}(\omega) \triangleq \tilde{\mathbf{A}}_{i1}(\omega) \oplus_q \tilde{\mathbf{B}}_{i2}(\omega) \in \mathbb{F}_q^{\frac{m_A}{2s_r} \times \frac{m}{s_c}},$$
$$p_i^{(3)}(\omega) \triangleq \tilde{\mathbf{A}}_{i2}^\intercal(\omega)\tilde{\mathbf{A}}_{i1}(\omega) \oplus_q \tilde{\mathbf{B}}_{i1}^\intercal(\omega)\tilde{\mathbf{B}}_{i2}(\omega) \in \mathbb{F}_q^{\frac{m}{s_c} \times \frac{m}{s_c}}.$$

Here, $p_i^{(1)}(\omega)$ and $p_i^{(2)}(\omega)$ are linear polynomials in the submatrices defined in (3), while $p_i^{(3)}(\omega)$ is a parity polynomial obtained through non-linear processing as defined in (4). Although $p_i^{(3)}(\omega)$ involves non-linear terms, it remains linearly separable in $\tilde{\mathbf{A}}_i(\omega)$ and $\tilde{\mathbf{B}}_i(\omega)$, which enables the asymptotically lossless construction of $\mathbf{p}_i(\omega)$ using the Körner–Marton encoding.

f) Post-processing: After evaluating $\mathbf{p}_i(\omega)$, each vehicle $\omega \in \Omega$ performs a local post-processing step. Specifically, vehicle ω, using (2), (3), and (6), computes

$$p_i(\omega) \triangleq \big(p_i^{(1)}(\omega)\big)^\intercal \cdot p_i^{(2)}(\omega) \ominus_q p_i^{(3)}(\omega). \quad (7)$$

Under the relaxed condition $\tilde{\mathbf{A}}_i^\intercal(\omega)\tilde{\mathbf{B}}_i(\omega) = \tilde{\mathbf{B}}_i^\intercal(\omega)\tilde{\mathbf{A}}_i(\omega)$, the post-processing operation in (7) yields the desired partial matrix product. As a result, each vehicle $\omega \in \Omega$ obtains and transmits the computational output

$$p_i(\omega) = \tilde{\mathbf{A}}_i^\intercal(\omega)\tilde{\mathbf{B}}_i(\omega) = \sum_{(j,k,j',k') \in \mathcal{S}} \alpha_i^\intercal(j,k)\beta_i(j',k') \times x_\omega^{j + s_c(s_r - 1 - j' + k) + s_c(2s_r - 1)k'} \quad (8)$$

to the receiver, where $\mathcal{S} \triangleq S_c \times S_r \times S_r \times S_c$ denotes the Cartesian product of the corresponding index sets.

g) Receiver-side decoding: The receiver collects evaluations of $\tilde{\mathbf{A}}_i^\intercal(\omega)\tilde{\mathbf{B}}_i(\omega)$ from a subset of cooperative vehicles $\omega \in \Omega$. Once a sufficient number of such evaluations (equal to the recovery threshold N_r) is obtained, the receiver applies polynomial interpolation, following the approach in [19], to reconstruct the desired matrix product $\mathbf{A}_i^\intercal \mathbf{B}_i$ (cf. Proposition 3).

Next, we evaluate the structural information-theoretic secrecy, computation, and communication functionality of the aforementioned system model.

III. Main Results

We formalize a structural secrecy property of DSPolyDot codes via a normalized information leakage as follows.

Definition 1 (Structural secrecy with normalized information leakage)**.** *Let* $\mathcal{M} \triangleq (\{\mathbf{A}_i\}_{i \in [n]}, \{\mathbf{B}_i\}_{i \in [n]})$ *and* $\mathcal{Z}_\omega$ *denotes the observation of vehicle* ω *in a coded DMM scheme. The scheme provides* structural information-theoretic secrecy via normalized information leakage ratio *if there exists a constant*

$\alpha \in [0,1)$, *independent of* n, *such that for every vehicle* ω *and every blocklength* n, *we have that*

$$\frac{I_q(\mathcal{M};\mathcal{Z}_\omega)}{H_q(\mathcal{M})} \;\leq\; \alpha\,. \tag{9}$$

Consequently, each vehicle's observation reveals at most an α*-fraction of the entire information in the source matrices.*

The following result demonstrates the robustness of DSPolyDot codes against information leakage at vehicles.

Proposition 1. (Structural secrecy of DSPolyDot codes.) *In the DSPolyDot-coded DMM framework described in Section II with an input sequence* $\mathcal{M}$ *that captures the length-n i.i.d. realizations of the source matrices* $\mathbf{A}_i$ *and* $\mathbf{B}_i$*, the information leakage from the source matrices to any vehicle* ω *is bounded by a constant fraction* $\alpha \ll 1$ *of the total source information* $H_q(\mathcal{M})$*, i.e., each vehicle learns at most an* α*-fraction of the total information in the source matrices.*

Proof. Given $\mathcal{M} = (\{\mathbf{A}_i\}_{i\in[n]}, \{\mathbf{B}_i\}_{i\in[n]})$, where each of $\mathbf{A}_i, \mathbf{B}_i \in \mathbb{F}_q^{m_A\times m}$ has $m_A m$ independently and uniformly distributed entries over $\mathbb{F}_q$, the total source entropy is $H_q(\mathcal{M}) = 2nm_Am$. Furthermore, the transmissions from the source nodes to vehicle $\omega \in \Omega$ are described by $\mathcal{Z}_\omega \triangleq \big(\mathbf{C}F_1^n(\omega),\, \mathbf{C}F_2^n(\omega)\big)$, where $\mathbf{C} \in \mathbb{F}_q^{k_n\times n'}$ is the Körner-Marton encoder with blocklength $n' \triangleq n\big(\frac{m_A}{s_r}+\frac{m}{s_c}\big)\frac{m}{s_c}$, where s_r and s_c divide m_A and m, respectively. Since each encoded vector $\mathbf{C}F_t^n(\omega)$ lies in $\mathbb{F}_q^{k_n}$, the observation $\mathcal{Z}_\omega$ takes values in a set of size at most q^{2k_n}, and hence $H_q(\mathcal{Z}_\omega) \leq 2k_n$. By the non-negativity of conditional entropy,

$$I_q(\mathcal{M};\mathcal{Z}_\omega) = H_q(\mathcal{Z}_\omega) - H_q(\mathcal{Z}_\omega \mid \mathcal{M}) \leq H_q(\mathcal{Z}_\omega) \leq 2k_n\,.$$

Considering the encoding matrix $\mathbf{C}$, $k_n \leq n'$, and therefore

$$\begin{aligned}\frac{I_q(\mathcal{M};\mathcal{Z}_\omega)}{H_q(\mathcal{M})} &\leq \frac{2k_n}{2nm_Am} \leq \frac{2n'}{2nm_Am} = \frac{n\big(\frac{m_A}{s_r}+\frac{m}{s_c}\big)\frac{m}{s_c}}{nm_Am}\\ &= \frac{1}{s_rs_c} + \frac{m}{m_A}\cdot\frac{1}{s_c^2} \triangleq \alpha\,,\end{aligned}$$

where we have that $\alpha \ll 1$ by considering the possible values for parameters s_r, s_c, m, m_A. □

On the other hand, the normalized information leakage for PolyDot codes [19] is equal to 1 because $\mathcal{Z}_\omega \triangleq \big(\tilde{\mathbf{A}}^n(\omega), \tilde{\mathbf{B}}^n(\omega)\big)$, where $\tilde{\mathbf{A}}^n(\omega), \tilde{\mathbf{B}}^n(\omega)$ are the vectorized forms of (2) for n realizations, and

$$I_q(\mathcal{M};\mathcal{Z}_\omega) = H_q(\mathcal{M}) - H_q(\mathcal{M} \mid \mathcal{Z}_\omega) = H_q(\mathcal{M})\,.$$

We next provide a cost analysis for computation and communication of the proposed framework shown in Figure 2, respectively.

A. Computation Cost of DSPolyDot Codes

In this part, we detail the complexity of computation at the source nodes as well as the vehicle nodes, respectively. Note that we ignore the bounded complexity terms imposed by reshape or concatenation operations.

Proposition 2. (Computation cost per source node.) *The total computation cost of each source node, including the Körner-Marton linear encoding matrix, equals*

$$N\Big(m_Am + \frac{m^2m_A}{2s_rs_c^2}\Big)\,. \tag{10}$$

Proof. For each $\omega \in \Omega$, each source node first computes $\tilde{\mathbf{A}}_i(\omega)$ or $\tilde{\mathbf{B}}_i(\omega)$ (cf. (2)) with a complexity of $s_c \times s_r \times \frac{m_A}{s_r} \times \frac{m}{s_c}$. Each source node then computes one of the non-linear mappings f_{1i} and f_{2i}, each with a complexity of $\frac{m}{s_c}\times\frac{m_A}{2s_r}\times\frac{m}{s_c}$. Considering N vehicles and summing the aforementioned costs completes the proof. □

The encoding process in (6), incorporating the dimensions of polynomials, dictates to each vehicle a storage limit

$$M_{\text{StPolyDot}} = \frac{m_Am}{s_rs_c} + \frac{m^2}{s_c^2}\,. \tag{11}$$

The proposed DSPolyDot codes require storage no greater than that of PolyDot codes, given as $M_{\text{PolyDot}} = \frac{2m_Am}{s_rs_c}$ in [19], when $\frac{m}{s_c} \leq \frac{m_A}{s_r}$.

We next characterize the minimum number of vehicles needed to successfully evaluate $\mathbf{A}^\intercal\mathbf{B}$ at the receiver, known as the *recovery threshold*, for DSPolyDot codes.

Proposition 3. *The recovery threshold of DSPolyDot codes for the proposed DMM framework is expressed as*

$$N_r = s_c^2(2s_r-1)\,. \tag{12}$$

Proof. Exploiting (6)-(8), the degree of $p_i(\omega) = \tilde{\mathbf{A}}_i^\intercal\tilde{\mathbf{B}}_i$ is $\deg(p_i(\omega)) = (s_c-1)+s_c(s_r-1+s_r-1)+s_c(2s_r-1)(s_c-1)$. The recovery threshold is then obtained as (12). □

The recovery threshold of DSPolyDot codes for the proposed DMM scheme is equivalent to that of PolyDot codes specified in [19], allowing us to have a fair comparison.

Proposition 4. (Computation cost of a vehicle node.) *The computation cost of the vehicle* $\omega \in \Omega$ *is equal to*

$$\frac{m_Am}{s_rs_c} + \frac{m_Am^2}{2s_rs_c^2} + \frac{2m^2}{s_c^2}\,. \tag{13}$$

Proof. Each vehicle $\omega \in \Omega$ evaluates the summation of f_{1i} and f_{2i} with a cost of $\frac{m_Am}{s_rs_c} + \frac{m^2}{s_c^2}$ to build the polynomials $p_i^{(1)}(\omega)$, $p_i^{(2)}(\omega)$, and $p_i^{(3)}(\omega)$. Each vehicle then computes $p_i(\omega)$ (cf. (7), (8)), exploiting the dimensions of each computational term, with a cost of $\frac{m}{s_c}\times\frac{m_A}{2s_r}\times\frac{m}{s_c}$ for multiplying $\big(p_i^{(1)}(\omega)\big)^\intercal$ and $p_i^{(2)}(\omega)$), followed by the addition of the terms $\big(p_i^{(1)}(\omega)\big)^\intercal p_i^{(2)}(\omega) \in \mathbb{F}_q^{\frac{m}{s_c}\times\frac{m}{s_c}}$ and $p_i^{(3)}(\omega) \in \mathbb{F}_q^{\frac{m}{s_c}\times\frac{m}{s_c}}$ with a complexity of $\frac{m^2}{s_c^2}$. Summing all the costs, the total computation cost per vehicle equals (13). □

B. Communication Cost of DSPolyDot Codes

We here determine the communication complexity from the source node to the vehicle nodes as well as from the vehicles to the receiver, respectively, as follows.

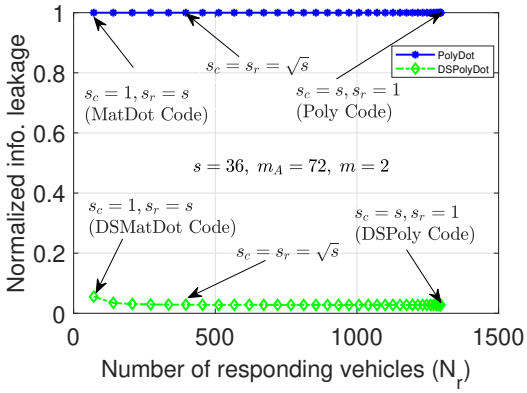

Fig. 3: The normalized information leakage at each vehicle node.

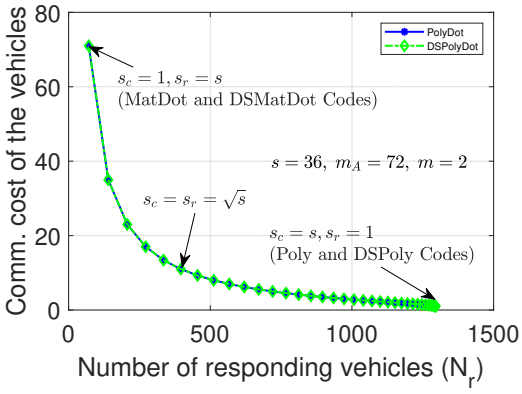

Fig. 4: Communication cost of vehicles for computing an element of $\mathbf{A}^\intercal\mathbf{B}$ (normalized: # symbols/m^2).

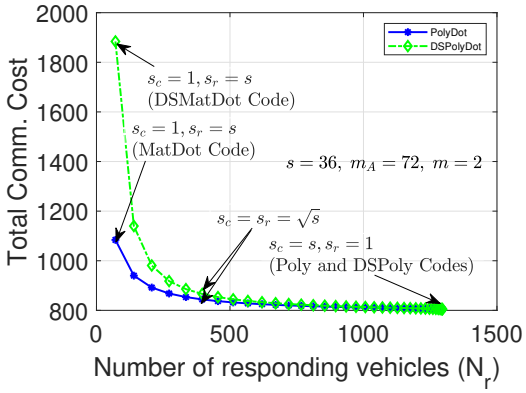

Fig. 5: Total communication cost (from sources to the vehicles and from vehicles to the receiver).

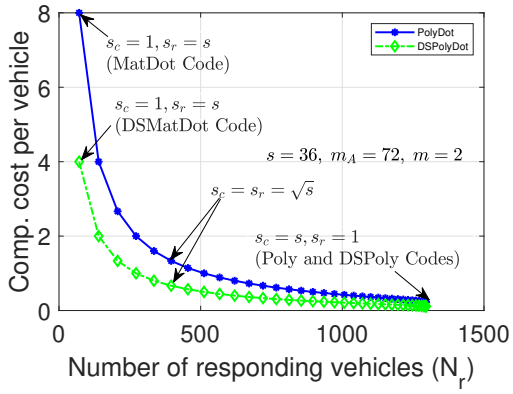

Fig. 6: Computation cost per vehicle for computing $\mathbf{A}^\intercal\mathbf{B}$ (normalized: # operations/N).

Proposition 5. (Communication cost per source node.) *The total communication cost of each source node for transmitting* $\mathbf{C}F_{1i}(\omega), \mathbf{C}F_{2i}(\omega)$ *to all vehicles* $\omega \in \Omega$ *is given by*

$$H_q\big(F_{1i}(\omega) \oplus_q F_{2i}(\omega)\big) = N\Big(\frac{m_A m}{s_r s_c} + \frac{m^2}{s_c^2}\Big)\,. \tag{14}$$

Proof. Exploiting the dimensions of each polynomial in (6), the communication cost of each source node for transmitting $\mathbf{C}F_{1i}(\omega), \mathbf{C}F_{2i}(\omega)$ to N vehicles equals (14). □

Proposition 6. (Communication cost per vehicle node.) *The communication cost of vehicle* $\omega \in \Omega$ *is*

$$H_q(p_i(\omega)) = H_q(\tilde{\mathbf{A}}_i^\intercal(\omega)\tilde{\mathbf{B}}_i(\omega)) = \frac{m^2}{s_c^2}\,. \tag{15}$$

Proof. Employing (8), vehicle $\omega \in \Omega$ transmits $p_i(\omega) \in \mathbb{F}_q^{\frac{m}{s_c}\times\frac{m}{s_c}}$ to the receiver with a cost equivalent to (15). □

Given N_r responding (active) vehicles, the total communication cost of vehicles is given as

$$\frac{m^2}{s_c^2} \times N_r = m^2(2s_r - 1)\,. \tag{16}$$

We next present the numerical results for the proposed DMM framework with DSPolyDot codes and compare them to the state-of-the-art.

IV. Numerical Evaluations of DSPolyDot Codes

We here provide numerical evaluations for DSPolyDot codes in a task-orchestrated vehicular edge computing framework and contrast them with the prior works [14], [19]. All numerical simulations assume that the elements of each source matrix are independently and uniformly distributed over $\mathbb{F}_q$.

As shown in Figure 3, the proposed framework provides intrinsic *secrecy guarantees* by bounding information leakage at cooperative vehicule nodes, which is vital in vehicular networks with sensor faults, unreliable links, and node failures.

In Figure 4, utilizing (16), we numerically evaluate the total communication cost of vehicles for computing an element of $\mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_q^{m\times m}$, i.e., the total number of transmitted symbols normalized by m^2. The curves for PolyDot and DSPolyDot codes overlap because in both approaches vehicle $\omega \in \Omega$ computes and then transmits $\tilde{\mathbf{A}}_i^\intercal\tilde{\mathbf{B}}_i$. Furthermore, for the total communication cost (from source nodes to vehicles and vehicles to the receiver), as shown in Figure 5, the proposed approach incurs a modest increase in cost when only

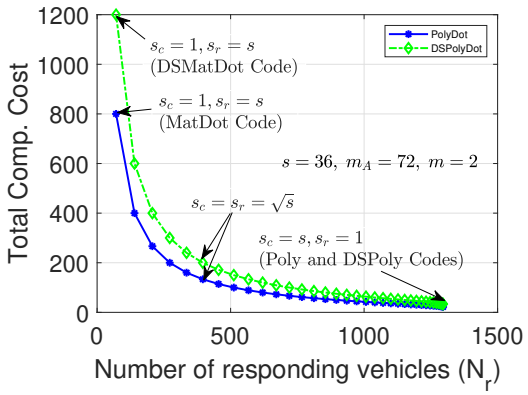

Fig. 7: Total computation cost (sources-vehicles-receiver).

a small number of vehicles are active, given the same set of parameters. This modest increase in communication cost is essential to guarantee secrecy. However, as the number of responding vehicles increases, the performance converges to that of PolyDot codes [19].

In Figure 6, exploiting (13), we illustrate the computation cost per vehicle for computing $\mathbf{A}^\intercal\mathbf{B}$, i.e., the total number of operations normalized by N, which demonstrates savings for a limited number of responding vehicles, however, asymptotically converging to that of PolyDot codes [19]. Finally, Figure 7 illustrates the total computation cost of the proposed DMM framework, including contributions from the source nodes, vehicles, and the receiver. When only a small number of vehicles participate, the proposed approach incurs a bounded increase in total computation cost. As the number of responding vehicles increases, the computation cost decreases and asymptotically converges to that of PolyDot codes [19].

V. Conclusion

We introduced a new class of polynomial codes for DMM (DSPolyDot codes) tailored to cooperative vehicular edge computing architectures involving vehicles and roadside units. By exploiting structured source coding and the algebraic structure of matrix multiplication via a non-linear parity polynomial, DSPolyDot codes generalize existing polynomial coding schemes to support cooperative computation under secrecy constraints. The proposed framework guarantees information-theoretic secrecy with bounded information leakage at individual cooperative nodes, while remaining robust to mobility-induced stragglers in vehicular networks. Compared to existing polynomial coding approaches [14], [19], these properties are achieved without increasing the asymptotic communication and computation overheads, making DSPolyDot codes well-suited for secrecy-constrained and straggler-resilient cooperative computation in vehicular networks.

Acknowledgments

Figure 1 is built via Gemini image creation tools (Banana).

References

[1] H. L. Nguyen Thi *et al.*, "Coded distributed computing for vehicular edge computing with dual-function radar communication," in *Proc., IEEE Veh. Technol. Conf. (VTC)*, Hong Kong, Hong Kong, Oct. 2023.
[2] J. Shamsi, M. A. Khojaye, and M. A. Qasmi, "Data-intensive cloud computing: Requirements, expectations, challenges, and solutions," *J. Grid Comput.*, vol. 11, no. 2, pp. 281–310, Jun. 2013.
[3] A. Toshniwal *et al.*, "Storm@twitter," in *Proc., ACM SIGMOD Int. Conf. Management of Data*. ACM, 2014, pp. 147–156.
[4] H. Park, M. Cho, Y. Jang, and J.-W. Choi, "Latency analysis of in-vehicle network for advanced driver assistance system," in *Proc., IEEE Veh. Technol. Conf. (VTC)*, Washington, DC, USA, Oct. 2024, pp. 1–7.
[5] T. Huang *et al.*, "Vehicle-to-everything cooperative perception for autonomous driving," *Proc., IEEE*, vol. 113, no. 5, pp. 443–477, 2025.
[6] X. Miao *et al.*, "Secure satellite–vehicle communications with randomly distributed vehicles on different roads," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 291–303, 2024.
[7] F. Xiao and D. Slock, "Performance analysis of hyperparameter optimization in sparse bayesian learning via Stein's unbiased risk estimator," in *Proc., 33rd Eur. Signal Process. Conf. (EUSIPCO)*, 2025.
[8] Z. Zhao, C. Forsch, L. Cottatellucci, and D. Slock, "Distributed iterative ML and message passing for grant-free cell-free massive MIMO systems," in *Proc., IEEE Middle East Conf. Commun. Netw.*, 2025.
[9] H. Yang, T. Ding, and X. Yuan, "Federated learning with lossy distributed source coding: Analysis and optimization," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4561–4576, May 2023.
[10] R. Tandon *et al.*, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc., Int. Conf. Machine Learning*. Sydney, Australia: PMLR, Aug. 2017, pp. 3368–3376.
[11] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.
[12] M. Sepulcre, J. Gozalvez, J. Härri, and H. Hartenstein, "Contextual communications congestion control for cooperative vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 385–389, 2011.
[13] A. B. Das, A. Ramamoorthy, and N. Vaswani, "Efficient and robust distributed matrix computations via convolutional coding," *IEEE Trans. Inf. Theory.*, vol. 67, no. 9, pp. 6266–6282, Jul. 2021.
[14] Q. Yu *et al.*, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," in *Proc., Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 4403–4413.
[15] ——, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc., Int. Conf. Art. Intellig. and Stats.* Naha, Okinawa, Japan: PMLR, Apr. 2019, pp. 1215–1225.
[16] ——, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1920–1933, Jan. 2020.
[17] M. Aliasgari, O. Simeone, and J. Kliewer, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2722–2734, Feb. 2020.
[18] M. Soleymani, H. Mahdavifar, and A. S. Avestimehr, "Analog Lagrange coded computing," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 283–295, Feb. 2021.
[19] S. Dutta *et al.*, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 278–301, Jul. 2019.
[20] A. M. Subramaniam, A. Heidarzadeh, and K. R. Narayanan, "Collaborative decoding of polynomial codes for distributed computation," in *Proc., IEEE Inf. Theory Wksh.*, Visby, Sweden, Aug. 2019, pp. 1–5.
[21] B. Karp, O. Amrani, and O. Keren, "Nonlinear product codes for reliability and security," in *Proc., IEEE Int. Verif. Secur. Workshop (IVSW)*, Rhodes, Greece, Jul. 2019, pp. 13–18.
[22] A. Nikkhah, M. Shoushtari, B. Akhbari, and W. K. Harrison, "Secrecy coding for the binary symmetric wiretap channel via linear programming," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 2450–2463, 2025.
[23] D. Malak, "Distributed structured matrix multiplication," in *Proc., IEEE ISIT*, Athens, Greece, Jul. 2024.
[24] A. Tanha, M. R. Deylam-Salehi, and D. Malak, "Structured coded matrix multiplication," in *Recent Results, IEEE Commun. Theory Wksh. (CTW)*, Venezia, Italy, May 2025.
[25] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.
[26] Y. Yang, P. Grover, and S. Kar, "Computing linear transformations with unreliable components," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3729–3756, Apr. 2017.
[27] V. Lalitha *et al.*, "Linear coding schemes for the distributed computation of subspaces," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 678–690, Mar. 2013.
[28] R. G. Gallager, *Information Theory and Reliable Communication*. Wiley: New York, Jan. 1968, vol. 588.