Private Polynomial Computation for Noncolluding Coded Databases

Sarah A. Obead[†], Hsuan-Yin Lin[‡], Eirik Rosnes[‡], and Jörg Kliewer[†]
[†]Helen and John C. Hartmann Department of Electrical and Computer Engineering
New Jersey Institute of Technology, Newark, New Jersey 07102, USA
[‡]Simula UiB, N–5008 Bergen, Norway

***Abstract*—We consider private polynomial computation (PPC) over noncolluding coded databases. In such a setting a user wishes to compute a multivariate polynomial of degree at most g over f variables (or messages) stored in multiple databases while revealing no information about the desired polynomial to the databases. We construct two novel PPC schemes, where the first is a generalization of our previous work in private linear computation for coded databases. In this scheme we consider Reed-Solomon coded databases with Lagrange encoding, which leverages ideas from recently proposed star-product private information retrieval and Lagrange coded computation. The second scheme considers the special case of coded databases with systematic Lagrange encoding. Both schemes yield improved rates compared to the best known schemes from the literature for a small number of messages, while in the asymptotic case the rates match.**

I. Introduction

The notion of private information retrieval (PIR) was introduced by Chor *et al.* in the computer science community [1]. The goal of PIR is to allow a user to privately access an arbitrary message stored in a set of databases, i.e., without revealing any information of the identity of the requested message to each database. The design of PIR protocols has focused on the case when multiple databases store the messages. This connects to the active and renowned research area of distributed storage systems (DSSs), where the messages are encoded by an $[n, k]$ linear code and then distributed and stored across n storage nodes. The study and design of efficient PIR protocols for coded DSSs have attracted a great deal of attention in recent years [2]–[6].

Private computation is a generalization of PIR that addresses the private computation for functions of the stored messages [7]–[13]. The scenario of noncolluding replicated databases for linear functions is considered in [7], [8] and referred to as private linear computation (PLC). The coded case is addressed in [10]–[13]. In particular, in [11], [12] we proposed a PLC scheme based on maximum distance separable (MDS) coded storage, where the obtained PLC capacity is equal to the MDS-coded PIR capacity in [4]. In [10], private polynomial computation (PPC) over t colluding and systematically coded databases is considered by generalizing the star-product PIR scheme of [3]. In that work, functions are computed that are polynomials of degree at most g, and a private computation rate equal to the best asymptotic PIR rate (when the number of messages tends to infinity) of MDS-coded storage is achieved for $g = t = 1$. An alternative PPC approach was recently proposed in [13] by employing Reed-Solomon (RS) coded databases with Lagrange encoding. For low code rates, the scheme improves on the private computation rate of [10].

In this work, we present two new approaches for PPC over coded databases by leveraging our previous works for PLC in [11], [12], ideas from star-product PIR [3], and Lagrange coded computation [14]. Our schemes apply to noncolluding RS-coded databases with Lagrange encoding. Compared to the scheme in [13], our first proposed PPC scheme yields a higher private computation rate when the number of messages is small. In addition, we construct a second PPC scheme for RS-coded databases with systematic Lagrange encoding that improves on the rate of the PPC scheme presented in [10]. In both cases, as the number of messages tends to infinity, the rate approaches those of [13] and [10], respectively. For the outer bound, we adopt our coded PLC capacity of [12, Thm. 2] since PPC can be seen as an extension of PLC.

II. Definitions and Problem Statement

A. Notation

We denote by $\mathbb{N}$ the set of all positive integers, $[a] \triangleq \{1, 2, \dots, a\}$, and $[a : b] \triangleq \{a, a + 1, \dots, b\}$ for $a, b \in \mathbb{N}$, $a \leq b$. A random variable is denoted by a capital Roman letter, e.g., X, while its realization is denoted by the corresponding small Roman letter, e.g., x. Vectors are boldfaced, e.g., $\boldsymbol{X}$ denotes a random vector and $\boldsymbol{x}$ denotes a deterministic vector, respectively. Random matrices are represented by bold sans serif letters, e.g., $\mathbfsf{X}$, where X represents its realization. In addition, sets are denoted by calligraphic uppercase letters, e.g., $\mathcal{X}$. $(\cdot)^\intercal$ denotes the transpose operator, $\mathsf{H}(X)$ represents the entropy of X, and $\mathsf{I}(X\,; Y)$ the mutual information between X and Y. The binomial coefficient of a over b, $a, b \in \{0\} \cup \mathbb{N}$, is denoted by $\binom{a}{b}$ where $\binom{a}{b} \triangleq 0$ if $a < b$. We use the customary code parameters $[n, k]$ to denote a code $\mathscr{C}$ over the finite field $\mathbb{F}_q$ of blocklength n and dimension k. The function $\chi(\boldsymbol{x})$ denotes the support of a vector $\boldsymbol{x}$, and the linear span of a set of vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_a\}$, $a \in \mathbb{N}$, is denoted by $\mathsf{span}\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_a\}$. A monomial $\boldsymbol{W}^{\boldsymbol{i}}$ in f variables $W^{(1)}, \dots, W^{(f)}$ with degree g is written as $\boldsymbol{W}^{\boldsymbol{i}} = (W^{(1)})^{i_1} (W^{(2)})^{i_2} \cdots (W^{(f)})^{i_f}$, where $\boldsymbol{i} \triangleq (i_1, \dots, i_f) \in (\{0\} \cup \mathbb{N})^f$ is the exponent vector with $\mathsf{wt}(\boldsymbol{i}) \triangleq \sum_{j=1}^{f} i_j = g$. Finally, a polynomial $\phi(\boldsymbol{W})$ of degree at most g is represented as $\phi(\boldsymbol{W}) = \sum_{\boldsymbol{i}:\mathsf{wt}(\boldsymbol{i}) \leq g} a_{\boldsymbol{i}} \boldsymbol{W}^{\boldsymbol{i}}$, $a_{\boldsymbol{i}} \in \mathbb{F}_q$. $\mathbb{F}_q[z]$ denotes the set of all univariate polynomials over $\mathbb{F}_q$ in the variable z. We denote by $\deg(\phi(z))$ the degree of a polynomial $\phi(z) \in \mathbb{F}_q[z]$.

This work is supported by US NSF grant CNS-1526547 and the Research Council of Norway (grant 240985/F20).

978-1-5386-9291-2/19/$31.00 ©2019 IEEE

B. Preliminaries

Definition 1 (Star-product)**.** *Let $\mathscr{C}$ and $\mathscr{D}$ be two linear codes of length n over $\mathbb{F}_q$. The star-product (Hadamard product) of $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathscr{C}$ and $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathscr{D}$ is defined as $\boldsymbol{v} \star \boldsymbol{u} = (v_1 u_1, \ldots, v_n u_n) \in \mathbb{F}_q^n$. Further, the star-product of $\mathscr{C}$ and $\mathscr{D}$, denoted by $\mathscr{C} \star \mathscr{D}$, is defined by $\mathsf{span}\{\boldsymbol{v} \star \boldsymbol{u} : \boldsymbol{v} \in \mathscr{C}, \boldsymbol{u} \in \mathscr{D}\}$ and the g-fold star-product of $\mathscr{C}$ with itself is given by $\mathscr{C}^{\star g} = \mathsf{span}\{\boldsymbol{v}_1 \star \cdots \star \boldsymbol{v}_g : \boldsymbol{v}_i \in \mathscr{C}, i \in [g]\}$.*

Definition 2 (Reed-Solomon code)**.** *Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ be a vector of n distinct nonzero elements of $\mathbb{F}_q$. For $n \in \mathbb{N}$, $k \in [n]$, and $q > n$, the $[n, k]$ RS code (over $\mathbb{F}_q$) is defined as*

$$\mathcal{RS}_k(\boldsymbol{\alpha}) \triangleq \{(\phi(\alpha_1), \ldots, \phi(\alpha_n)) \colon \phi \in \mathbb{F}_q[z], \deg(\phi) < k\}. \quad (1)$$

It is well-known that RS codes are MDS codes that behave well under the star-product. We state the following proposition that was introduced in [3].

Proposition 1. *Let $\mathcal{RS}_k(\boldsymbol{\alpha})$ be a length-n RS code. Then, for $g \in \mathbb{N}$, the g-fold star-product of $\mathcal{RS}_k(\boldsymbol{\alpha})$ with itself is the RS code given by $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha}) = \mathcal{RS}_{\min\{g(k-1)+1, n\}}(\boldsymbol{\alpha})$.*

Let $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_k)$ be a vector of k distinct elements of $\mathbb{F}_q$. For a message vector $\boldsymbol{W} = (W_1, \ldots, W_k)$, let $\ell(z) \in \mathbb{F}_q[z]$ be a polynomial of degree at most $k-1$ such that $\ell(\gamma_i) = W_i$ for all $i \in [k]$. Using the Lagrange interpolation formula we present this polynomial as $\ell(z) = \sum_{i \in [k]} W_i \iota_i(z)$, where $\iota_i(z)$ is the Lagrange basis polynomial

$$\iota_i(z) = \prod_{t \in [k] \setminus \{i\}} \frac{z - \gamma_t}{\gamma_i - \gamma_t}.$$

It has been shown in [13] that Lagrange encoding is equivalent to the choice of a specific basis for an RS code. Thus, for encoding we choose the set of Lagrange basis polynomials as the code generating polynomials of (1) [14]. Thus, a generator matrix of $\mathcal{RS}_k(\boldsymbol{\alpha})$ is $\mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = (\iota_i(\alpha_j))$, $i \in [k]$, $j \in [n]$. Note that if we choose $\gamma_i = \alpha_i$ for $i \in [k]$, then the generator matrix $\mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ becomes systematic.

The set $\{\boldsymbol{W}^{\boldsymbol{i}} : \boldsymbol{i} \in (\{0\} \cup \mathbb{N})^f, 1 \leq \mathsf{wt}(\boldsymbol{i}) \leq g\}$ of all monomials in f variables of degree at most g has size

$$\mathsf{M}(f, g) \triangleq \sum_{h=1}^{g} \binom{h + f - 1}{h} = \binom{g + f}{g} - 1,$$

and the total number of polynomials in f variables of degree at most g generated with all possible distinct (up to scalar multiplication) $\mathsf{M}(f, g)$-dimensional coefficients vectors defined over $\mathbb{F}_q$ is equal to $\mu(f, g) \triangleq \frac{q^{\mathsf{M}(f,g)} - 1}{q - 1}$.

C. System Model

An RS-coded DSS is described as follows. The DSS stores in total f independent messages $\mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(f)}$, where each message $\mathbf{W}^{(m)} = \bigl(W_{i,j}^{(m)}\bigr)$, $m \in [f]$, is a random $\beta \times k$ matrix with some $\beta, k \in \mathbb{N}$, where each entry is chosen independently and uniformly at random from $\mathbb{F}_q$. Thus, $\mathsf{H}(\mathbf{W}^{(m)}) = \beta k \triangleq \mathsf{L}$, $\forall\, m \in [f]$ (in q-ary units).

Each message is encoded using an $[n, k]$ RS code as follows. Let $\boldsymbol{W}_i^{(m)} = \bigl(W_{i,1}^{(m)}, \ldots, W_{i,k}^{(m)}\bigr)$, $i \in [\beta]$, be a message vector corresponding to the i-th row of $\mathbf{W}^{(m)}$. Each $\boldsymbol{W}_i^{(m)}$ is encoded by an RS code $\mathcal{RS}_k(\boldsymbol{\alpha})$ with evaluation vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ over $\mathbb{F}_q$ into a length-n codeword $\boldsymbol{C}_i^{(m)}$ where $\boldsymbol{C}_i^{(m)} = \boldsymbol{W}_i^{(m)} \mathsf{G}_{\mathcal{RS}_k}(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \bigl(C_{i,1}^{(m)}, \ldots, C_{i,n}^{(m)}\bigr)$ and $C_{i,j}^{(m)} = \ell_i^{(m)}(\alpha_j)$, $j \in [n]$, where $\ell_i^{(m)}(z)$ is the Lagrange interpolation polynomial associated with the length-k message segment $\boldsymbol{W}_i^{(m)}$. The βf generated codewords $\boldsymbol{C}_i^{(m)}$ are then arranged in the array $\mathsf{C} = \bigl((\mathsf{C}^{(1)})^\mathsf{T} | \ldots | (\mathsf{C}^{(f)})^\mathsf{T}\bigr)^\mathsf{T}$ of dimensions $\beta f \times n$, where $\mathsf{C}^{(m)} = \bigl((\boldsymbol{C}_1^{(m)})^\mathsf{T} | \ldots | (\boldsymbol{C}_\beta^{(m)})^\mathsf{T}\bigr)^\mathsf{T}$. The code symbols $C_{1,j}^{(m)}, \ldots, C_{\beta,j}^{(m)}$, $m \in [f]$, for all f messages are stored on the j-th database, $j \in [n]$.

D. Private Polynomial Computation for RS-Coded DSSs

We consider the case of n noncolluding databases. A user wishes to privately compute exactly one polynomial out of μ *candidate* polynomial functions $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}$ from the RS-coded DSS while keeping the requested index private from each database. The polynomial function $\mathbf{X}^{(v)} = \bigl(\phi^{(v)}(\boldsymbol{W}_{i,j})\bigr)$, where $\boldsymbol{W}_{i,j} = (W_{i,j}^{(1)}, \ldots, W_{i,j}^{(f)})$, is a $\beta \times k$ random matrix for some polynomial $\phi^{(v)}$, where each $\phi^{(v)}(\boldsymbol{W}_{i,j}) \in \mathbb{F}_q$ is independent and distributed according to some probability mass function P_{X_v}. Thus, $\mathsf{H}(\mathbf{X}^{(v)}) = \mathsf{L}\,\mathsf{H}(X_v)$, $\forall\, v \in [\mu]$, and $\mathsf{H}(\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}) = \mathsf{L}\,\mathsf{H}(X_1, \ldots, X_\mu)$.

Consider an RS-coded DSS with n noncolluding databases storing f messages. The user wishes to retrieve the v-th polynomial function $\mathbf{X}^{(v)}$, $v \in [\mu]$, from the available information from queries $Q_j^{(v)}$ and answer strings $A_j^{(v)}$, $j \in [n]$. For a PPC protocol, the following conditions must be satisfied $\forall\, v \in [\mu]$,

[Privacy]

$$\mathsf{I}\bigl(v; Q_j^{(v)}, A_j^{(v)}, \mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(\mu)}\bigr) = 0, \ \forall\, j \in [n],$$

[Recovery]

$$\mathsf{H}\bigl(\mathbf{X}^{(v)} \bigm| A_1^{(v)}, \ldots, A_n^{(v)}, Q_1^{(v)}, \ldots, Q_n^{(v)}\bigr) = 0.$$

Definition 3 (PPC rate for RS-coded DSSs)**.** *The rate of a PPC scheme, denoted by* R*, is defined as* $\mathsf{R} = \mathsf{L}/\mathsf{D}$*, where* D *is the total required download cost.*[1]

Definition 4 (τ-sum)**.** *For $\tau \in [\mu]$, a sum $\phi^{(v_1)}(\boldsymbol{C}_{i_1,j}) + \cdots + \phi^{(v_\tau)}(\boldsymbol{C}_{i_\tau,j})$, where $\boldsymbol{C}_{i,j} = (C_{i,j}^{(1)}, \ldots, C_{i,j}^{(f)})$, $i \in [\beta]$, $j \in [n]$, of τ distinct candidate polynomial function evaluations is called a τ-sum for any $(i_1, \ldots, i_\tau) \in [\beta]^\tau$, and $\{v_1, \ldots, v_\tau\} \subseteq [\mu]$ determines the type of the τ-sum.*

III. A General PPC Scheme for RS-Coded DSSs With Lagrange Encoding

In the following we build a PPC scheme based on Lagrange encoding and our PLC scheme in [12]. Note that a polynomial can be written as a linear combination of monomials, and therefore any private monomial computation (PMC) scheme is a special case of PPC. Thus, a PPC scheme can be obtained from a PLC scheme by replacing independent messages with a monomial basis. We first discuss the PPC case in general and then provide an example for the special case of PMC.

[1] In order to compare with the PPC schemes from [10], [13], we use a slightly imprecise definition of the PPC rate. The exact information-theoretic PPC rate is defined as the ratio of the minimum desired polynomial function size $\mathsf{L}\min_{v \in [\mu]} \mathsf{H}(X_v)$ over the total required download cost D.

A. Lagrange Coded Computation

Lagrange coded computation [14] is a framework that can be applied to any function computation when the function of interest is a multivariate polynomial of the messages. We extend the application of this framework to PMC and PPC by utilizing the following argument.

Recall that $\ell_t^{(m)}(z)$, $t \in [\beta]$, $m \in [f]$, evaluated at γ_j results in an information symbol $W_{t,j}^{(m)}$ and when evaluated at α_j we obtain a code symbol $C_{t,j}^{(m)}$. Let $\boldsymbol{\ell}_t(z) = (\ell_t^{(1)}(z), \ldots, \ell_t^{(f)}(z))$ be a vector of f Lagrange interpolation polynomials associated with the messages $\boldsymbol{W}_t^{(1)}, \ldots, \boldsymbol{W}_t^{(f)}$. Now, given a multivariate polynomial function $\phi(\boldsymbol{W}_{t,j})$ of degree at most g, we introduce the composition function $\psi_t(z) = \phi(\boldsymbol{\ell}_t(z))$. Accordingly, evaluating $\psi_t(z)$ at any γ_j, $j \in [k]$, is equal to evaluating the polynomial function over the uncoded information symbols, i.e., $\phi(\boldsymbol{W}_{t,j})$ and similarly, evaluating $\psi_t(z)$ at α_j, $j \in [n]$, will result in the evaluation of the polynomial function over the coded symbols, i.e., $\phi(\boldsymbol{C}_{t,j})$. Since each Lagrange interpolation polynomial of $\boldsymbol{\ell}_t(z)$ is a polynomial of degree at most $k-1$, it follows that $\deg(\psi_t(z)) \leq g(k-1)$ and we require up to $g(k-1)+1$ coefficients to interpolate and determine the polynomial $\psi_t(z)$.

Note that $\psi_t(z)$ is a linear combination of monomials $z^i \in \mathbb{F}_q[z]$, $i \leq g(k-1)$, and the underlying code $\tilde{\mathscr{C}}$ for $(\psi_t(\alpha_1), \ldots, \psi_t(\alpha_n))$, referred to as the *decoding code*, is given by the g-fold star-product $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ of the storage code $\mathcal{RS}_k(\boldsymbol{\alpha})$ according to [13, Lem. 7]. This is due to the fact that the span of $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ is given by linear combinations of codewords in $\mathcal{RS}_k^{\star g}(\boldsymbol{\alpha})$ where each code symbol represents a monomial. With other words, to construct coded PPC schemes that retrieve polynomials of degree at most g, we require $g(k-1)+1 \leq n$ and $d_{\mathsf{min}}^{\tilde{\mathscr{C}}} \geq n - (g(k-1)+1)+1$, where $d_{\mathsf{min}}^{\tilde{\mathscr{C}}}$ denotes the minimum distance of $\tilde{\mathscr{C}}$, to be able to decode the computation correctly. It follows from Proposition 1 that $\tilde{\mathscr{C}} = \mathcal{RS}_{\tilde{k}}(\boldsymbol{\alpha})$ with dimension $\tilde{k} = \min\{g(k-1)+1, n\} = g(k-1)+1$ and $d_{\mathsf{min}}^{\tilde{\mathscr{C}}} = n - \tilde{k} + 1 = n - (g(k-1)+1)+1$.

B. PPC Achievable Rate Matrix

Similar to [12, Def. 3], where we introduce the notion of a PIR achievable rate matrix for the coded PLC problem, we provide the following definition for the PPC case.

Definition 5. *A $\nu \times n$ binary matrix $\mathsf{\Lambda}_{\kappa,\nu}$ is called a* PPC achievable rate matrix *for $(\mathscr{C}, \tilde{\mathscr{C}})$ if the following conditions are satisfied.*

1) *The Hamming weight of each column of $\mathsf{\Lambda}_{\kappa,\nu}$ is κ, and*
2) *for each matrix row $\boldsymbol{\lambda}_i$, $i \in [\nu]$, $\chi(\boldsymbol{\lambda}_i)$ is always an information set for $\tilde{\mathscr{C}}$.*

C. Redundancy Elimination

Here, we generalize the coded PLC scheme of [12] in terms of exploiting the dependency between the virtual messages. Since any polynomial is a linear function of the monomial basis of size $\mathsf{M}(f,g)$, a PPC scheme can be seen as a PLC scheme performed over a set of $\mathsf{M}(f,g)$ messages. Hence, the redundancy resulting from the linear dependencies between the virtual messages is also present for PPC and we can extend [12, Lem. 1] and [8, Lem. 1] to our scheme. To exploit the dependency between the virtual messages we adopt a similar sign assignment process to each queried symbol of the virtual monomial messages, based on the desired function index v as introduced in [8, Sec. IV.B]. This will result in a uniquely solvable equation system from the different τ-sum types given the side information available from all other databases. By obtaining such a system of equations in each round $\tau \in [\mu]$ of the protocol, the user can determine some of the answers offline.

Now, consider 1-sum types, where we download individual segments of each virtual message including f independent messages. For these types, the user can determine any polynomial from the f obtained message segments. Based on this insight we can state the following lemma.

Lemma 1. *Let $\mu \in [f : \mu(f,g)]$ be the number of candidate polynomials, including the f independent messages. For each query set, for all $v \in [\mu]$, each database $j \in [n]$, and based on the queried segments from the f independent messages, there are $\binom{\mu-f}{1}$ redundant 1-sum types out of all possible types $\binom{\mu}{1}$. On the other hand, for $\tau \in [2:\mu]$, there are $\binom{\max\{\mu-\mathsf{M}(f,g),0\}}{\tau}$ redundant τ-sum types out of $\binom{\mu}{\tau}$ types. The number of nonredundant τ-sum types with $\tau > 1$ is given by $\rho(\mu,\tau) \triangleq \binom{\mu}{\tau} - \binom{\max\{\mu-\mathsf{M}(f,g),0\}}{\tau}$.*

D. Achievable PPC Rate

Since $\tilde{\mathscr{C}}$ is an $[n, \tilde{k}]$ MDS code ($\mathscr{C}$ is an RS code), there always exists a PPC achievable rate matrix $\Lambda_{\kappa,\nu}$ with $(\kappa,\nu) = (\tilde{k}, n)$ for $(\mathscr{C}, \tilde{\mathscr{C}})$. Hence, using Lemma 1 we can prove the following theorem.

Theorem 1. *Consider a DSS that uses an $[n,k]$ RS code $\mathscr{C}$ to store f messages over n noncolluding databases using Lagrange encoding. Let $\mu \in [f : \mu(f,g)]$ be the number of candidate polynomials to be computed of degree at most g, $g(k-1)+1 \leq n$, including the f independent messages. Then, the PPC rate*

$$\mathsf{R}_{\text{PPC}} = \frac{kn^{\mu-1}}{f\tilde{k}^{\mu} + \sum_{\tau=2}^{\mu} \rho(\mu,\tau)\tilde{k}^{\mu-\tau+1}\big(n-\tilde{k}\big)^{\tau-1}}$$

is achievable.

We remark that the PPC scheme requires the length of each message to be $\mathsf{L} = k \cdot \nu^{\mu}$. Note that our proposed scheme cannot readily be obtained using the concept of refinement and lifting of so-called one-shot schemes as introduced for PIR in [15], since this concept cannot readily be applied to the function computation case.

We now provide further insight into our proposed PPC scheme by considering the PMC scheme as a special case in which the candidate set is restricted to contain monomials.

E. Special Case: PMC Scheme

1) Candidate Monomials: As the rate of PMC is a decreasing function of the number of candidate monomial functions, we can limit ourselves to the set of monomials excluding *parallel* monomials, where we define a parallel monomial as a monomial resulting from raising another monomial to a positive integer power, i.e., to $\{\boldsymbol{W}^{\boldsymbol{i}} : \boldsymbol{i} \in (\{0\} \cup \mathbb{N})^f, 1 \leq \mathsf{wt}(\boldsymbol{i}) \leq g, \boldsymbol{i} \mid p, p \in \mathcal{P}_g\}$, where $\mathcal{P}_g$ denotes the set of prime numbers less or equal to g and $\boldsymbol{i} = (i_1, \ldots, i_f) \mid p$ means that all nonzero i_j, $j \in [f]$, are divisors of p. For example, for a bivariate monomial over the variables x and y of degree at most $g = 2$ the set of possible monomials is $\{x, y, xy, x^2, y^2\}$. Note that x^2 is a parallel monomial as it can be obtained by raising the monomial x to the power of 2. Thus, x^2 and y^2 are parallel monomials and can be excluded from the set of candidate monomials. Denote by $\mathcal{P} = \{p_1, \ldots, p_{|\mathcal{P}|}\}$ an arbitrary nonempty subset of $\mathcal{P}_g$. By applying the Legendre formula for counting the prime numbers less or equal to g, we obtain the number of nonparallel monomials as

$$\widetilde{\mathsf{M}}(f, g) = \binom{g+f}{g} - 1 + \sum_{\substack{\forall \mathcal{P} \subseteq \mathcal{P}_g : \mathcal{P} \neq \emptyset, \\ p_1 \cdots p_{|\mathcal{P}|} \leq g}} (-1)^{|\mathcal{P}|} \left[\binom{\left\lfloor \frac{g}{p_1 \cdots p_{|\mathcal{P}|}} \right\rfloor + f}{\left\lfloor \frac{g}{p_1 \cdots p_{|\mathcal{P}|}} \right\rfloor} - 1 \right],$$

where $\lfloor \cdot \rfloor$ denotes the floor function.

We illustrate the key concept of our proposed scheme in Theorem 1 with an example. Note that in all examples we assume that the *index preparation* step has been performed to keep the desired polynomial index private. We refer the readers to [12, Sec. IV-A] for details. Before we proceed with the example, given a $\nu \times n$ PPC achievable rate matrix $\Lambda_{\kappa,\nu}$, we define the notion of PPC interference matrices as follows.

Definition 6 ([12, Def. 5])**.** *For a given $\nu \times n$ PPC achievable rate matrix $\Lambda_{\kappa,\nu} = (\lambda_{u,j})$ for $(\mathscr{C}, \tilde{\mathscr{C}})$, we define the PPC interference matrices $\mathsf{A}_{\kappa \times n} = (a_{i,j})$ and $\mathsf{B}_{(\nu-\kappa) \times n} = (b_{i,j})$ for the code $\tilde{\mathscr{C}}$ with*

$$a_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 1, \forall j \in [n], i \in [\kappa], u \in [\nu],$$
$$b_{i,j} \triangleq u \text{ if } \lambda_{u,j} = 0, \forall j \in [n], i \in [\nu - \kappa], u \in [\nu].$$

Note that in Definition 6, for each $j \in [n]$, distinct values of $u \in [\nu]$ should be assigned for all i. Thus, the assignment is not unique in the sense that the order of the entries of each column of A and B can be permuted.

Example 1. *Consider two messages $\mathsf{W}^{(1)}$ and $\mathsf{W}^{(2)}$ that are stored in a noncolluding DSS using a $[4, 2]$ RS code $\mathscr{C}$. Suppose that the user wishes to obtain a monomial function $\mathsf{X}^{(v)}$ from the candidate set $\{\mathsf{W}^{(1)}, \mathsf{W}^{(2)}, \mathsf{W}^{(1)} \star \mathsf{W}^{(2)}\}$ of monomial functions, i.e., $\mu = \widetilde{\mathsf{M}}(2, 2) = 3$. We have $\tilde{k} = g(k-1)+1 = 3$ and*

$$\Lambda_{3,4} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

is a valid PPC achievable rate matrix for $(\mathscr{C}, \tilde{\mathscr{C}})$. From $\Lambda_{3,4}$ we further obtain the interference matrices

$$\mathsf{A}_{3 \times 4} = \begin{pmatrix} 1 & 1 & 1 & 2 \\ 2 & 2 & 3 & 3 \\ 3 & 4 & 4 & 4 \end{pmatrix} \text{ and } \mathsf{B}_{1 \times 4} = \begin{pmatrix} 4 & 3 & 2 & 1 \end{pmatrix}.$$

TABLE I
QUERY SETS FOR A $[4, 2]$ RS-CODED DSS WITH LAGRANGE ENCODING STORING $f = 2$ MESSAGES AND WHERE THE FIRST $(v = 1)$ MONOMIAL IS PRIVATELY COMPUTED FOR $g = 2$ AND $\mu = 3$.

j	1	2	3	4
$Q_j^{(1)}(\mathcal{D}; 1)$	$x_{1:9,1}, x_{10:18,1}, x_{19:27,1}$	$x_{1:9,2}, x_{10:18,2}, x_{28:36,2}$	$x_{1:9,3}, x_{19:27,3}, x_{28:36,3}$	$x_{10:18,2}, x_{19:27,3}, x_{28:36,4}$
$Q_j^{(1)}(\mathcal{U}; 1)$	$y_{1:9,1}, y_{10:18,1}, y_{19:27,1}$	$y_{1:9,2}, y_{10:18,2}, y_{28:36,2}$	$y_{1:9,3}, y_{19:27,3}, y_{28:36,3}$	$y_{10:18,2}, y_{19:27,3}, y_{28:36,4}$
$Q_j^{(1)}(\mathcal{D}; 2)$	$x_{37:39,1} + y_{28:30,1}$ $x_{40:42,1} + z_{28:30,1}$ $x_{43:45,1} + y_{31:33,1}$ $x_{46:48,1} + z_{31:33,1}$ $x_{49:51,1} + y_{34:36,1}$ $x_{52:54,1} + z_{34:36,1}$	$x_{37:39,2} + y_{19:21,2}$ $x_{40:42,2} + z_{19:21,2}$ $x_{43:45,2} + y_{22:24,2}$ $x_{46:48,2} + z_{22:24,2}$ $x_{55:57,2} + y_{25:27,2}$ $x_{58:60,2} + z_{25:27,2}$	$x_{37:39,3} + y_{10:12,3}$ $x_{40:42,3} + z_{10:12,3}$ $x_{49:51,3} + y_{13:15,3}$ $x_{52:54,3} + z_{13:15,3}$ $x_{55:57,3} + y_{16:18,3}$ $x_{58:60,3} + z_{16:18,3}$	$x_{43:45,4} + y_{1:3,4}$ $x_{46:48,4} + z_{1:3,4}$ $x_{49:51,4} + y_{4:6,4}$ $x_{52:54,4} + z_{4:6,4}$ $x_{55:57,4} + y_{7:9,4}$ $x_{58:60,4} + z_{7:9,4}$
$Q_j^{(1)}(\mathcal{U}; 2)$	$y_{40:42,1} + z_{37:39,1}$ $y_{46:48,1} + z_{43:45,1}$ $y_{52:54,1} + z_{49:51,1}$	$y_{40:42,2} + z_{37:39,2}$ $y_{46:48,2} + z_{43:45,2}$ $y_{58:60,2} + z_{55:57,2}$	$y_{40:42,3} + z_{37:39,3}$ $y_{52:54,3} + z_{49:51,3}$ $y_{58:60,3} + z_{55:57,3}$	$y_{46:48,4} + z_{43:45,4}$ $y_{52:54,4} + z_{49:51,4}$ $y_{58:60,4} + z_{55:57,4}$
$Q_j^{(1)}(\mathcal{D}; 3)$	$x_{61,1} + y_{58,1} + z_{55,1}$ $x_{62,1} + y_{59,1} + z_{56,1}$ $x_{63,1} + y_{60,1} + z_{57,1}$	$x_{61,2} + y_{52,2} + z_{49,2}$ $x_{62,2} + y_{53,2} + z_{50,2}$ $x_{64,2} + y_{54,2} + z_{51,2}$	$x_{61,3} + y_{46,3} + z_{43,3}$ $x_{63,3} + y_{47,3} + z_{44,3}$ $x_{64,3} + y_{48,3} + z_{45,3}$	$x_{62,4} + y_{40,4} + z_{37,4}$ $x_{63,4} + y_{41,4} + z_{38,4}$ $x_{64,4} + y_{42,4} + z_{39,4}$

We simplify notation by letting $x_{t,j} = C_{t,j}^{(1)}$, $y_{t,j} = C_{t,j}^{(2)}$, and $z_{t,j} = C_{t,j}^{(1)} \cdot C_{t,j}^{(2)}$ for all $t \in [\beta]$, $j \in [n]$, where $\beta = \nu^\mu = 64$. Let the desired monomial function index be $v = 1$. The construction of the query sets is briefly presented in the following steps.[2]

Initialization* (*Round* $\tau = 1$)*: *We start with $\tau = 1$ to generate query sets for each database j holding $\kappa^\mu = 27$ distinct instances of $x_{t,j}$. By message symmetry this also applies to $y_{t,j}$ and $z_{t,j}$.*

Following Rounds (*$\tau \in [2 : 3]$*)**:** *Using the interference matrices $\mathsf{A}_{3 \times 4}$ and $\mathsf{B}_{1 \times 4}$ for the exploitation of side information for the j-th database, $j \in [n]$, we generate the desired query sets $Q_j^{(1)}(\mathcal{D}; \tau)$ by querying a number of new symbols of the desired monomial jointly combined with symbols from other monomials queried in the previous round from database $i \neq j$. Next, the undesired query sets $Q_j^{(1)}(\mathcal{U}; \tau)$ (if $\tau = 2$) are generated by enforcing message symmetry. We make the final modification to the query sets by removing all redundant 1-sum types from the first round (see Lemma 1) and update the query sets. This translates to removing the queries for $z_{t,j}$, since they can be generated offline by the user given $x_{t,j}$ and $y_{t,j}$. The resulting query sets are shown in Table I, where $u_{a:b,j} \triangleq (u_{a,j}, \ldots, u_{b,j})$ for $u = x, y, z$. The PMC rate of the scheme is equal to $\frac{k\nu^\mu}{\mathsf{D}} = \frac{2 \times 4^3}{3 \times 4 \times 28} = 0.3810$.*

IV. PPC Scheme for RS-Coded DSSs With Systematic Lagrange Encoding

In this section, we consider the case of RS-coded DSSs with systematic Lagrange encoding and first adapt the concept of a PPC achievable rate matrix from Definition 5 to this scenario by extending [6, Def. 14]. In contrast to the PPC scheme in Section III, the basic idea is to utilize the systematic part of the RS code to recover the requested function.

Definition 7. *A $\nu \times n$ binary matrix $\Lambda_{\kappa,\nu}^{\mathsf{S}}$ is called a* PPC systematic achievable rate matrix *for $(\mathscr{C}, \tilde{\mathscr{C}})$ if the following conditions are satisfied.*

1) *$\Lambda_{\kappa,\nu}^{\mathsf{S}}$ is a κ-column regular matrix, and*
2) *there are exactly κ rows $\{\boldsymbol{\lambda}_i\}_{i \in [\kappa]}$ and $\nu - \kappa$ rows $\{\boldsymbol{\lambda}_{i+\kappa}\}_{i \in [\nu-\kappa]}$ of $\Lambda_{\kappa,\nu}^{\mathsf{S}}$ such that $\forall i \in [\kappa]$, $\chi(\boldsymbol{\lambda}_i)$*

[2] With some abuse of notation, the generated queries are sets containing their answers, and vectors should be considered as the union of their entries.

TABLE II
QUERY SETS FOR A $[4,2]$ RS-CODED DSS WITH SYSTEMATIC LAGRANGE ENCODING STORING $f=2$ MESSAGES AND WHERE THE FIRST $(v=1)$ MONOMIAL IS PRIVATELY COMPUTED FOR $g=2$ AND $\mu=3$.

j	1	2	3	4
$Q_j^{(1)}(\mathcal{D};1)$	$x_{1:4,1}$, $x_{9:12,1}$	$x_{5:8,2}$, $x_{9:12,2}$	$x_{1:4,3}$, $x_{5:8,3}$	$x_{1:4,4}$, $x_{5:8,4}$
$Q_j^{(1)}(\mathcal{U};1)$	$y_{1:4,1}$, $y_{9:12,1}$	$y_{5:8,2}$, $y_{9:12,2}$	$y_{1:4,3}$, $y_{5:8,3}$	$y_{1:4,4}$, $y_{5:8,4}$
$Q_j^{(1)}(\mathcal{D};2)$	$x_{13:14,1}+y_{5:6,1}$ $x_{15:16,1}+z_{5:6,1}$ $x_{21:22,1}+y_{7:8,1}$ $x_{23:24,1}+z_{7:8,1}$	$x_{17:18,2}+y_{1:2,2}$ $x_{19:20,2}+z_{1:2,2}$ $x_{21:22,2}+y_{3:4,2}$ $x_{23:24,2}+z_{3:4,2}$	$x_{13:14,3}+y_{9:10,3}$ $x_{15:16,3}+z_{9:10,3}$ $x_{17:18,3}+y_{11:12,3}$ $x_{19:20,3}+z_{11:12,3}$	$x_{13:14,4}+y_{9:10,4}$ $x_{15:16,4}+z_{9:10,4}$ $x_{17:18,4}+y_{11:12,4}$ $x_{19:20,4}+z_{11:12,4}$
$Q_j^{(1)}(\mathcal{U};2)$	$y_{15:16,1}+z_{13:14,1}$ $y_{23:24,1}+z_{21:22,1}$	$y_{19:20,2}+z_{17:18,2}$ $y_{23:24,2}+z_{21:22,2}$	$y_{15:16,3}+z_{13:14,3}$ $y_{19:20,3}+z_{17:18,3}$	$y_{15:16,4}+z_{13:14,4}$ $y_{19:20,4}+z_{17:18,4}$
$Q_j^{(1)}(\mathcal{D};3)$	$x_{25,1}+y_{19,1}+z_{17,1}$ $x_{27,1}+y_{20,1}+z_{18,1}$	$x_{26,2}+y_{15,2}+z_{13,2}$ $x_{27,2}+y_{16,2}+z_{14,2}$	$x_{25,3}+y_{23,3}+z_{21,3}$ $x_{26,3}+y_{24,3}+z_{22,3}$	$x_{25,4}+y_{23,4}+z_{21,4}$ $x_{26,4}+y_{24,4}+z_{22,4}$

contains an information set for $\tilde{\mathscr{C}}$ *and* $\forall\, i \in [\nu-\kappa]$, $\chi(\boldsymbol{\lambda}_{i+\kappa}) = [k]$.

Using Lemma 1, the following theorem follows since it can be proved that a PPC systematic achievable rate matrix $\Lambda^{\mathsf{S}}_{\kappa,\nu}$ with $(\kappa,\nu) = \bigl(k, k+\min\{k, n-\tilde{k}\}\bigr)$ always exists.

Theorem 2. *Consider a DSS that uses an* $[n,k]$ *RS code* $\mathscr{C}$ *to store* f *messages over* n *noncolluding databases using systematic Lagrange encoding. Let* $\mu \in [f : \mu(f,g)]$ *be the number of candidate polynomials to be computed of degree at most* g, $g(k-1)+1 \leq n$, *including the* f *independent messages. Then, the PPC rate*

$$\mathsf{R}^{\mathsf{S}}_{\mathsf{PPC}} = \frac{\nu^{\mu}}{n\Bigl[fk^{\mu-1} + \sum_{\tau=2}^{\mu} \rho(\mu,\tau) k^{\mu-\tau} \bigl(\nu-k\bigr)^{\tau-1}\Bigr]},$$

with $\nu = k + \min\{k, n-\tilde{k}\}$, *is achievable.*

Example 2. *Consider the same scenario as in Example 1 where* $n=4$, $k=2$, *and* $\tilde{k}=3$. *It follows that* $\nu = k + \min\{k, n-\tilde{k}\} = 3$ *and*

$$\Lambda^{\mathsf{S}}_{2,3} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

is a valid PPC systematic achievable rate matrix. We further obtain (by adapting Definition 6 correspondingly)

$$\mathsf{A}^{\mathsf{S}}_{2\times 4} = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 3 & 3 & 2 & 2 \end{pmatrix} \text{ and } \mathsf{B}^{\mathsf{S}}_{1\times 4} = \begin{pmatrix} 2 & 1 & 3 & 3 \end{pmatrix}$$

from $\Lambda^{\mathsf{S}}_{2,3}$. *The resulting query sets are shown in Table II for* $\mu=3$, *where* $u_{a:b,j} \triangleq (u_{a,j},\dots,u_{b,j})$ *for* $u = x,y,z$, *and the PMC rate* $\frac{k\nu^{\mu}}{\mathsf{D}} = \frac{2\times 3^3}{2\times 4\times 15} = 0.45$ *is achievable.*

V. NUMERICAL RESULTS

In Fig. 1, we compare the PPC rates of Theorems 1 and 2 to those of the schemes from [10], [13] for $n=5$, $k=2$, and $g=2$. The proposed schemes show improved performance for a low number of messages f. Observe that the curves converge to the rates from [10], [13] as the number of messages f grows. In fact, it can easily be seen from the rate expressions of Theorems 1 and 2 that this is always the case (details omitted for brevity). For comparison, we also plot the PMC rate when parallell monomials are excluded (magenta and purple lines).

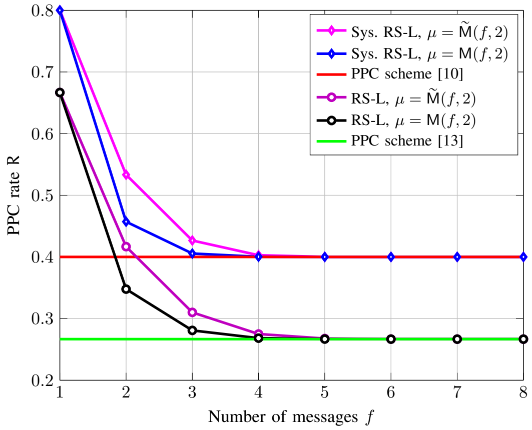

Fig. 1. Achievable PPC rates as a function of the number of messages f for $n=5$, $k=2$, and $g=2$.

VI. CONVERSE BOUND

Since RS codes are MDS codes and PPC can be seen as an extension of PLC, we can adapt the coded PLC capacity of [12, Thm. 2] to be an outer bound to the PPC rate. However, for an infinite number of messages the PPC rates of our proposed schemes, as for the schemes of [10], [13], do not approach this outer bound, and it is still unknown whether the PLC capacity can be achieved by a coded PPC scheme.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th IEEE Symp. Found. Comp. Sci.*, Milwaukee, WI, USA, Oct. 23–25, 1995, pp. 41–50.

[2] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[3] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geom.*, vol. 1, no. 1, pp. 647–664, Nov. 2017.

[4] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[5] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[6] S. Kumar, H.-Y. Lin, E. Rosnes, and A. Graell i Amat, "Achieving maximum distance separable private information retrieval capacity with linear codes," 2019, to app. in *IEEE Trans. Inf. Theory*.

[7] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *Proc. Iran Workshop Commun. Inf. Theory*, Tehran, Iran, Apr. 2018.

[8] H. Sun and S. A. Jafar, "The capacity of private computation," 2019, to app. in *IEEE Trans. Inf. Theory*.

[9] Z. Chen, Z. Wang, and S. Jafar, "The asymptotic capacity of private search," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2122–2126.

[10] D. Karpuk, "Private computation of systematically encoded data with colluding servers," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2112–2116.

[11] S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from MDS coded databases," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2117–2121.

[12] S. A. Obead, H.-Y. Lin, E. Rosnes, and J. Kliewer, "Capacity of private linear computation for coded databases," in *Proc. 56th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2–5, 2018.

[13] N. Raviv and D. A. Karpuk, "Private polynomial computation from Lagrange encoding," Dec. 2018, arXiv:1812.04142v2 [cs.IT].

[14] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and A. S. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, vol. 89, Naha, Okinawa, Japan, Apr. 16–18, 2019, pp. 1215–1225.

[15] R. G. L. D'Oliveira and S. El Rouayheb, "Lifting private information retrieval from two to any number of messages," in *Proc. IEEE Int. Symp. Inf. Theory*, Vail, CO, USA, Jun. 17–22, 2018, pp. 1744–1748.