Conference Paper

On the Capacity of Private Monomial Computation

Author(s):
Yakimenka, Yauhen; Lin, Hsuan-Yin; Rosnes, Eirik

Publication Date:
2020-02-26

Permanent Link:

https://doi.org/10.3929/ethz-b-000402672 →

International Zurich Seminar on Information and Communication (IZS), February 26 – 28, 2020

On the Capacity of Private Monomial Computation

Yauhen Yakimenka, Hsuan-Yin Lin, and Eirik Rosnes
Simula UiB, N–5008 Bergen, Norway
Email: {yauhen, lin, eirikrosnes}@simula.no

***Abstract*—In this work, we consider private monomial computation (PMC) for replicated noncolluding databases. In PMC, a user wishes to privately retrieve an arbitrary multivariate monomial from a candidate set of monomials in f messages over a finite field $\mathbb{F}_q$, where $q = p^k$ is a power of a prime p and $k \geq 1$, replicated over n databases. We derive the PMC capacity under a technical condition on p and for asymptotically large q. The condition on p is satisfied, e.g., for large enough p. Also, we present a novel PMC scheme for arbitrary q that is capacity-achieving in the asymptotic case above. Moreover, we present formulas for the entropy of a multivariate monomial and for a set of monomials in uniformly distributed random variables over a finite field, which are used in the derivation of the capacity expression.**

I. Introduction

The concept of private computation (PC) was introduced independently by Sun and Jafar [1] and Mirmohseni and Maddah-Ali [2]. In PC, a user wishes to compute a function of the messages stored in a set of databases without revealing any information about the function to any of the databases. PC can be seen as a generalization of private information retrieval (PIR). In PIR, a user wants to retrieve a single message from the set of databases privately. Applications of PC include, in principle, all scenarios where insights about certain actions of the user should be kept private. One practical motivation for considering arbitrary functions is that of *algorithmic privacy*, as protecting the identity of an algorithm running in the cloud could be even more critical than data privacy in some scenarios. Not only could the algorithm be valuable, but also in some cases, parameters of the algorithm carry lifetime secrets such as biological information of individuals [2].

The capacity in the linear case, i.e., the computation of arbitrary linear combinations of the stored messages, has been settled for both replicated [1] and coded [3], [4] databases. In the coded databases scenario, the messages are encoded by a linear code before being distributed and stored in a set of databases. Interestingly, the capacity in the linear case is equal to the corresponding PIR capacity for both replicated and coded databases. The monomial case was recently considered in [5], [6]. However, the presented achievable schemes have a PC rate, defined here as the ratio between the *smallest* desired amount of information and the total amount of downloaded information, that in general is strictly lower than the best known converse bound for a finite number of messages. PC schemes in the coded case for arbitrary polynomials were considered by Karpuk and Raviv in [7], [8], and recently improved in [5] when the number of messages is small. The capacity of private polynomial computation for coded databases remains open.

In this work, we first derive formulas for the entropy of a multivariate monomial and a set of monomials in uniformly distributed random variables over a finite field. We then present a novel PC scheme for multivariate monomials in the messages stored in a set of replicated noncolluding databases. The key ingredient of the scheme is the use of discrete logarithms. The discrete logarithm in the multiplicative group of a finite field of order $q = p^k$ (p is a prime and $k \geq 1$) is a bijection to the integer ring of size $q-1$, mapping multiplication to addition. Hence, the discrete logarithm maps multivariate monomial retrieval to linear function retrieval, given that none of the messages is the zero element. The latter holds with probability approaching one as q becomes large. The corresponding PC rate in this limiting case is derived using the entropy formulas from the first part of the paper. When the candidate set of multivariate monomials is fixed (i.e., independent of q), the PC rate converges to the PIR capacity for any number of messages stored in the databases, under a technical condition on p and as q goes to infinity. The condition on p is satisfied, e.g., for large enough p. Also, the presented monomial computation scheme is capacity-achieving in this asymptotic case.

II. Preliminaries

A. General Definitions and Notation

Throughout the paper, vectors are denoted by bold font and matrices are written as sans-serif capitals.

We work with different algebraic structures: the ring of integers $\mathbb{Z}$, rings of residuals $\mathbb{Z}_m$ for integers $m > 1$, and finite fields $\mathbb{F}_q$, where $q = p^k$ is a power of a prime p and $k \geq 1$. Occasionally, $\mathcal{R}$ denotes any of these structures. We often use the connection between $\mathbb{Z}$ and $\mathbb{Z}_m$. In principle, any element in $\mathbb{Z}$ can be considered as an element of $\mathbb{Z}_m$, with correspondence of addition and multiplication. If an expression consists of both integers and elements of $\mathbb{Z}_m$, we assume all operations are over $\mathbb{Z}_m$. When we need to stress that an element is in $\mathbb{Z}_m$, we write $a^{\langle m \rangle} \in \mathbb{Z}_m$ for $a \in \mathbb{Z}$. The same notation is used for matrices, e.g., $\mathsf{A}^{\langle m \rangle}$ has entries $a_{ij}^{\langle m \rangle} \in \mathbb{Z}_m$ for $a_{ij} \in \mathbb{Z}$.

Any $a \in \mathbb{Z}$ can be viewed as $a^{\langle p \rangle} \in \mathbb{Z}_p = \mathbb{F}_p \subseteq \mathbb{F}_q$. Operations on such elements of $\mathbb{F}_q$ are modulo p, as p is the *characteristic* of $\mathbb{F}_q$, i.e., the minimum positive integer l such that $l \cdot \alpha = 0$ for all $\alpha \in \mathbb{F}_q$. Analogously, $\mathsf{A} \in \mathbb{Z}^{s \times t}$ can be viewed as $\mathsf{A}^{\langle p \rangle} \in \mathbb{F}_q^{s \times t}$. Note the difference between $\mathsf{A}^{\langle p \rangle} \in \mathbb{F}_q^{s \times t}$ and $\mathsf{A}^{\langle q \rangle} \in \mathbb{Z}_q^{s \times t}$ for $q = p^k$ and $k > 1$.

The *multiplicative group* $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ is cyclic (cf. [9, Thm. 2.18]), and it is possible to define a *discrete logarithm*

function[1] $\mathrm{dlog} : \mathbb{F}_q^* \to \mathbb{Z}_{q-1}$, which is an isomorphism between $(\mathbb{F}_q^*, \times)$ and $(\mathbb{Z}_{q-1}, +)$.

We write $[a] \triangleq \{1, \ldots, a\}$ for a positive integer a. The *greatest common divisor (gcd)* of $a_1, \ldots, a_s \in \mathbb{Z}$ is denoted by $\gcd(a_1, \ldots, a_s)$, with the convention $\gcd(0, \ldots, 0) \triangleq 0$ and $\gcd(a_1^{\langle m \rangle}, \ldots, a_s^{\langle m \rangle}, m) \triangleq \gcd(a_1, \ldots, a_s, m)$. We write $a \mid b$ when a divides b, and $a \nmid b$ otherwise. The binomial coefficient of a over b (both nonnegative integers) is denoted by $\binom{a}{b}$ where $\binom{a}{b} = 0$ if $a < b$. The transpose of A is denoted by $\mathsf{A}^\intercal$.

A $k \times k$ *minor* in $\mathcal{R}$ of a matrix $\mathsf{A} \in \mathcal{R}^{s \times t}$, for a positive integer k, is the determinant of a $k \times k$ submatrix of A obtained by removing $s-k$ rows and $t-k$ columns from A. The largest integer r such that there is a nonzero $r \times r$ minor of A is called the *rank* of A in $\mathcal{R}$ and denoted by $\mathrm{rank}_{\mathcal{R}}\, \mathsf{A}$. A matrix $\mathsf{A} \in \mathcal{R}^{s \times s}$ is invertible in $\mathcal{R}$ if and only if the determinant of A is invertible as an element of $\mathcal{R}$ (cf. [9, Thm. 2.1]).

For $\mathsf{A} \in \mathbb{Z}^{s \times t}$, we denote the gcd of all $k \times k$ minors of A by $g_k(\mathsf{A})$. If $\delta \in \mathbb{Z}$ is some minor of A, the corresponding minor of $\mathsf{A}^{\langle m \rangle}$ is $\delta^{\langle m \rangle}$. Hence, $\mathrm{rank}_{\mathbb{Z}_m}\, \mathsf{A} = \mathrm{rank}_{\mathbb{Z}}\, \mathsf{A}$ for all $m \nmid g_r(\mathsf{A})$, where $r = \mathrm{rank}_{\mathbb{Z}}\, \mathsf{A}$.[2] Also,

$$\mathrm{rank}_{\mathbb{F}_q}\, \mathsf{A} = \mathrm{rank}_{\mathbb{Z}}\, \mathsf{A} \quad \Leftrightarrow \quad p \nmid g_r(\mathsf{A}). \tag{1}$$

It is known [10, Cor. 1.13, Cor. 1.20] that there exists a unique diagonal matrix $\mathsf{D} = \mathrm{diag}(d_1, \ldots, d_{\min(s,t)}) \in \mathbb{Z}^{s \times t}$ called the *Smith normal form* of A, with the following properties.

1) $\mathsf{D} = \mathsf{PAQ}$ for some matrices $\mathsf{P} \in \mathbb{Z}^{s \times s}$ and $\mathsf{Q} \in \mathbb{Z}^{t \times t}$ invertible in $\mathbb{Z}$,
2) $d_i \mid d_{i+1}$ for $i \in [\min(s,t) - 1]$,
3) $d_1 d_2 \cdots d_i = g_i(\mathsf{A})$ for $i \in [\min(s,t)]$.

The diagonal elements $d_1, \ldots, d_{\min(s,t)}$ are *invariant factors*, and $d_i = 0$ if and only if $i > \mathrm{rank}_{\mathbb{Z}}\, \mathsf{A}$. While D is unique, the matrices P and Q are not unique in the general case. It is also important to mention that the Smith normal form is defined for matrices over *principal ideal domains (PIDs)*. For example, $\mathbb{Z}$ is a PID while $\mathbb{Z}_m$ is not (in general).

Random variables are labeled by capital roman letters and we write $X \sim Y$ to indicate that X and Y are identically distributed. Moreover, $X \sim \mathfrak{U}(\mathcal{S})$ means that X is uniformly distributed over the set $\mathcal{S}$. We use $\log$ to denote logarithm base-2, although most statements hold for an arbitrary constant base. We denote the entropy in bits and q-ary units by $\mathrm{H}(\cdot)$ and $\mathrm{H}_q(\cdot)$, respectively, and $\mathrm{I}(\cdot;\cdot)$ denotes mutual information. The binary entropy function is denoted by $h(\cdot)$.

The notation $O(\phi(x))$ stands for any function $\psi(x)$ in x such that $|\psi(x)/\phi(x)| < \mathsf{B}$ for all large enough x and some constant $\mathsf{B} > 0$ independent of x. Also, $o(\phi(x))$ represents any $\psi(x)$ such that $\lim_{x\to\infty} \psi(x)/\phi(x) = 0$. In particular, $O(1)$ is any bounded function and $o(1)$ is any function that converges to zero as $x \to \infty$.

B. Private Computation

Suppose we have n noncommunicating databases, each storing duplicated data: f messages subpacketized into λ parts, each part denoted as $X_i^{(j)} \in \mathbb{F}_q$ for $i \in [f]$ and $j \in [\lambda]$. The subpackets are considered mutually independent and uniformly drawn from $\mathbb{F}_q$. There are μ public functions $\varphi_1, \ldots, \varphi_\mu$, where $\varphi_i : \mathbb{F}_q^f \to \mathbb{F}_q$ for $i \in [\mu]$. The user randomly chooses a secret index $V \sim \mathfrak{U}([\mu])$ and wants to retrieve

$$\boldsymbol{F}_V = \left(\varphi_V(\boldsymbol{X}^{(1)}), \ldots, \varphi_V(\boldsymbol{X}^{(\lambda)})\right) \in \mathbb{F}_q^\lambda,$$

where $\boldsymbol{X}^{(j)} \triangleq (X_1^{(j)}, \ldots, X_f^{(j)})$, $j \in [\lambda]$, without revealing any information about V. To achieve that, the user and the databases employ the following scheme.

1) The user generates secret randomness R, computes queries $Q_j = Q_j(V, R)$, $j \in [n]$, and sends the j-th query to the j-th database.
2) Based on Q_j and all the messages, the j-th database computes the response $A_j = A_j\left(Q_j, \boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(\lambda)}\right)$ and sends it back to the user.
3) Using all available information, the user can recover $\boldsymbol{F}_V$.

Formally, we require the scheme to satisfy

$$\begin{aligned} &\text{Privacy:} && \mathrm{I}(V; Q_j) = 0, \text{ for all } j \in [n], \\ &\text{Recovery:} && \mathrm{H}(\boldsymbol{F}_V \mid V, R, A_1, \ldots, A_n) = 0. \end{aligned}$$

Definition 1. The *download rate* of a PC scheme over the field $\mathbb{F}_q$, referred to as the PC rate, is defined as

$$\mathsf{R} = \mathsf{R}(n, f, \mu, \{\varphi_i\}, \lambda, \{Q_j\}, \{A_j\}, q) \triangleq \frac{\min_{v \in [\mu]} \mathrm{H}(\boldsymbol{F}_v)}{\Delta},$$

where Δ is the expected total number of downloaded bits, referred to as the *download cost*. The supremum of all achievable rates for all choices of λ, $\{Q_j\}$, and $\{A_j\}$ is the *PC capacity* over $\mathbb{F}_q$, $\mathsf{C}_{\mathrm{PC}}(n, f, \mu, \{\varphi_i\}, q)$.

In case $\mu = f$ and $\varphi_i(x_1, \ldots, x_f) = x_i$ for $i \in [f]$, PC reduces to PIR with capacity $\mathsf{C}_{\mathrm{PIR}}(n, f) \triangleq (1 + 1/n + 1/n^2 + \cdots + 1/n^{f-1})^{-1}$ [11]. Note that $\mathsf{C}_{\mathrm{PIR}}$ is independent of q.

The case when $\varphi_1, \ldots, \varphi_\mu$ are linear functions described by a matrix of coefficients $\mathsf{A} \in \mathbb{F}_q^{\mu \times f}$ without zero rows, is referred to as private linear computation (PLC). Its capacity $\mathsf{C}_{\mathrm{PLC}}$ only depends on n and $r = \mathrm{rank}_{\mathbb{F}_q}\, \mathsf{A}$, and it holds that $\mathsf{C}_{\mathrm{PLC}}(n, r) = \mathsf{C}_{\mathrm{PIR}}(n, r)$ [1].[3]

In this work, we consider private monomial computation (PMC), i.e., the case when $\varphi_i(x_1, \ldots, x_f) = x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_f^{a_{if}}$, $i \in [\mu]$, where $a_{ij} \in \mathbb{Z}$. The monomials can be described by a matrix of degrees $\mathsf{A} = (a_{ij}) \in \mathbb{Z}^{\mu \times f}$, and we assume there are no constant functions, i.e., no zero rows in A. The capacity of PMC is denoted by $\mathsf{C}_{\mathrm{PMC}}(n, f, \mu, \mathsf{A}, q)$.

III. Entropies of Linear Functions and Monomials

Lemma 1. *Let $a \in \mathbb{Z}$ and $Y \sim \mathfrak{U}(\mathbb{Z}_m)$. Then,*

$$\mathrm{H}(aY) = \mathrm{H}(a^{\langle m \rangle} Y) = \log m - \log \gcd(a, m).$$

Proof: From the theory of linear congruences [12, Sec. 5, Thm. 1], the equation $ay = b$ has $d = \gcd(a, m)$ solutions in

[1] Strictly speaking, dlog requires fixing a particular generator of $\mathbb{F}_q^*$.

[2] In particular, the requirement $a \nmid b$ is satisfied if $a > b$.

[3] In [1], the authors assume the messages are among the functions, e.g., $\varphi_i(x_1, \ldots, x_f) = x_i$ for $i \in [f]$. However, this is not required as we can define linearly independent functions as new variables and express other functions in these variables.

$\mathbb{Z}_m$ if $d \mid b$ and no solutions otherwise. Therefore, the random variable aY takes m/d different values from $\mathbb{Z}_m$ equiprobably, and the required statement follows. ∎

Lemma 2. *Let $\mathsf{A} \in \mathbb{Z}^{s\times t}$ be a fixed matrix whose invariant factors are $d_1, \dots, d_{\min(s,t)}$. Let $\boldsymbol{Y} = (Y_1, \dots, Y_t) \sim \mathfrak{U}(\mathbb{Z}_m^t)$, $r = \operatorname{rank}_{\mathbb{Z}} \mathsf{A}$, and $r' = \operatorname{rank}_{\mathbb{Z}_m} \mathsf{A}^{\langle m\rangle}$. Then,*

$$\mathrm{H}(\mathsf{A}\boldsymbol{Y}) = r\log m - \sum_{i=1}^{r} \log\gcd(d_i, m) \tag{2}$$

$$= r'\log m - \sum_{i=1}^{r'} \log\gcd(d_i, m). \tag{3}$$

Proof: Recall that, since $\boldsymbol{Y}$ is defined over $\mathbb{Z}_m^t$, the operations in $\mathsf{A}\boldsymbol{Y}$ are over $\mathbb{Z}_m$. In other words, $\mathsf{A}\boldsymbol{Y}$ is a shorthand for $\mathsf{A}^{\langle m\rangle}\boldsymbol{Y}$.

Let $\mathsf{D} = \mathsf{PAQ}$ be the Smith normal form of A, where both $\mathsf{P} \in \mathbb{Z}^{s\times s}$ and $\mathsf{Q} \in \mathbb{Z}^{t\times t}$ are invertible over $\mathbb{Z}$ (i.e., their determinants are ± 1) and $\mathsf{D} = \operatorname{diag}(d_1, \dots, d_r, 0, \dots, 0)$. After taking modulo m from both sides, we obtain $\mathsf{D}^{\langle m\rangle} = \mathsf{P}^{\langle m\rangle}\mathsf{A}^{\langle m\rangle}\mathsf{Q}^{\langle m\rangle}$, where $\mathsf{P}^{\langle m\rangle}$ and $\mathsf{Q}^{\langle m\rangle}$ are both invertible over $\mathbb{Z}_m$ (their determinants are ± 1 in $\mathbb{Z}_m$ too) and $\mathsf{D}^{\langle m\rangle} = \operatorname{diag}(d_1^{\langle m\rangle}, \dots, d_r^{\langle m\rangle}, 0, \dots, 0)$. Therefore,

$$\begin{aligned}\mathrm{H}(\mathsf{D}^{\langle m\rangle}\boldsymbol{Y}) &= \mathrm{H}(\mathsf{P}^{\langle m\rangle}(\mathsf{A}^{\langle m\rangle}\mathsf{Q}^{\langle m\rangle}\boldsymbol{Y})) = \mathrm{H}(\mathsf{A}^{\langle m\rangle}\mathsf{Q}^{\langle m\rangle}\boldsymbol{Y})\\ &= \mathrm{H}(\mathsf{A}^{\langle m\rangle}(\mathsf{Q}^{\langle m\rangle}\boldsymbol{Y})) = \mathrm{H}(\mathsf{A}^{\langle m\rangle}\boldsymbol{Y}) = \mathrm{H}(\mathsf{A}\boldsymbol{Y}),\end{aligned}$$

because $\mathsf{P}^{\langle m\rangle}$ and $\mathsf{Q}^{\langle m\rangle}$ are invertible over $\mathbb{Z}_m$, and multiplication from the left by an invertible matrix is a bijection. Thus, we can consider $\mathrm{H}(\mathsf{D}^{\langle m\rangle}\boldsymbol{Y})$ instead of $\mathrm{H}(\mathsf{A}\boldsymbol{Y})$. But $\mathsf{D}^{\langle m\rangle}\boldsymbol{Y} = (d_1^{\langle m\rangle}Y_1, \dots, d_r^{\langle m\rangle}Y_r, 0, \dots, 0)$ with mutually independent entries. Hence,

$$\begin{aligned}\mathrm{H}(\mathsf{D}^{\langle m\rangle}\boldsymbol{Y}) &= \sum_{i=1}^{r} \mathrm{H}(d_i^{\langle m\rangle}Y_i)\\ &\overset{\text{Lem. 1}}{=} r\log m - \sum_{i=1}^{r} \log\gcd(d_i, m).\end{aligned}$$

Finally, (3) holds because $m \mid d_i$ for $i > r'$ and hence $\gcd(d_i, m) = m$. ∎

Corollary 1. *In the setting of Lemma 2, $\mathrm{H}(\mathsf{A}\boldsymbol{Y}) = r\log m + O(1)$, as $m \to \infty$, where $r = \operatorname{rank}_{\mathbb{Z}} \mathsf{A}$.*

Proof: For all $m > d_r$ and all $i \in [\min(s,t)]$, it holds that $d_i^{\langle m\rangle} = d_i$. In this case, $r' = r$ and

$$\begin{aligned}\mathrm{H}(\mathsf{A}\boldsymbol{Y}) &= r\log m - \sum_{i=1}^{r} \log\gcd(d_i, m)\\ &\geq r\log m - \log\prod_{i=1}^{r} d_i = r\log m - \log g_r(\mathsf{A}).\end{aligned} \tag{4}$$

On the other hand,

$$\mathrm{H}(\mathsf{A}\boldsymbol{Y}) = r\log m - \sum_{i=1}^{r} \log\gcd(d_i, m) \leq r\log m. \tag{5}$$

We note that both (4) and (5) are attained for infinitely many values of m, e.g., for $m = ug_r(\mathsf{A})$ and $m = 1 + ug_r(\mathsf{A})$, respectively (for any positive integer u). In other words, $\mathrm{H}(\mathsf{A}\boldsymbol{Y})$ does not converge as $m \to \infty$.

Finally, as $\log g_r(\mathsf{A})$ does not depend on m, we have

$$\mathrm{H}(\mathsf{A}\boldsymbol{Y}) = r\log m + O(1), \text{ as } m \to \infty.$$ ∎

Next, we present some results on entropies of monomials over finite fields. The key idea is to use the bijection of dlog and treat a special case of zero separately.

Lemma 3. *Let $a_1, \dots, a_t \in \mathbb{Z}$, $X_1, \dots, X_t \sim \mathfrak{U}(\mathbb{F}_q)$ be mutually independent, τ be the number of nonzeros among $a_1, \dots, a_t$, and $\pi = (1 - 1/q)^\tau$. Then,*

$$\mathrm{H}(X_1^{a_1}X_2^{a_2}\cdots X_t^{a_t}) = h(\pi) + \pi\log\frac{q-1}{\gcd(a_1, \dots, a_t, q-1)}.$$

Moreover, if not all $a_1, \dots, a_t$ are zeros,

$$\mathrm{H}_q(X_1^{a_1}X_2^{a_2}\cdots X_t^{a_t}) \xrightarrow[q\to\infty]{} 1.$$

Proof: If $a_i = 0$, the variable X_i is not present in the monomial. Hence, we can exclude such variables and assume $a_1, \dots, a_\tau \in \mathbb{Z} \setminus \{0\}$. Dropping zero arguments of the gcd above does not change its value either.

Let $M = X_1^{a_1}X_2^{a_2}\cdots X_\tau^{a_\tau}$. Define $Z = 0$ if $M = 0$ and $Z = 1$ otherwise. Then, $\pi = \mathbb{P}\{M \neq 0\} = \mathbb{P}\{Z = 1\}$ and

$$\begin{aligned}\mathrm{H}(M) &= \mathrm{H}(Z) + \mathrm{H}(M \mid Z) - \mathrm{H}(Z \mid M)\\ &= h(\pi) + \mathrm{H}(M \mid Z = 0)(1-\pi) + \mathrm{H}(M \mid Z = 1)\pi\\ &= h(\pi) + \pi\,\mathrm{H}(M \mid M \neq 0).\end{aligned}$$

Now, $M \neq 0$ if and only if none of $X_1, \dots, X_\tau$ is zero. In this case, all $X_1, \dots, X_\tau \in \mathbb{F}_q^*$ and we can define $Y_j = \operatorname{dlog} X_j \in \mathbb{Z}_{q-1}$ for $j \in [\tau]$ and $L' = \operatorname{dlog} M = a_1Y_1 + \cdots + a_\tau Y_\tau \in \mathbb{Z}_{q-1}$. Since dlog is bijective, $Y_1, \dots, Y_\tau \sim \mathfrak{U}(\mathbb{Z}_{q-1})$ and $\mathrm{H}(M \mid M \neq 0) = \mathrm{H}(L')$. By applying Lemma 2 with $m = q-1$, $s = 1$, $r = 1$, and $d_1 = \gcd(a_1, \dots, a_\tau)$, we get

$$\mathrm{H}(L') = \log\frac{q-1}{\gcd(a_1, \dots, a_\tau, q-1)}.$$

Further, as $q \to \infty$, $\pi \to 1$ and therefore $h(\pi) \to 0$. Additionally, $\gcd(a_1, \dots, a_\tau, q-1) \leq \min(|a_1|, \dots, |a_\tau|) = O(1)$, as $q \to \infty$. Finally,

$$\mathrm{H}_q(X_1^{a_1}X_2^{a_2}\cdots X_t^{a_t}) = \frac{\mathrm{H}(X_1^{a_1}X_2^{a_2}\cdots X_t^{a_t})}{\log q} \xrightarrow[q\to\infty]{} 1.$$ ∎

Theorem 1. *Let $\mathsf{A} \in \mathbb{Z}^{s\times t}$ be a fixed matrix of coefficients with rank $r = \operatorname{rank}_{\mathbb{Z}} \mathsf{A}$. Let $X_1, \dots, X_t \sim \mathfrak{U}(\mathbb{F}_q)$ be mutually independent. For $i \in [s]$, define $M_i = X_1^{a_{i1}}X_2^{a_{i2}}\cdots X_t^{a_{it}} \in \mathbb{F}_q$ and $\boldsymbol{M} = (M_1, \dots, M_s)$. Then,*

$$\mathrm{H}(\boldsymbol{M}) = r\log q + O(1), \text{ as } q \to \infty.$$

Proof: First, if there is a zero column in A, we can drop the corresponding variable, as it does not influence either the values of any of the monomials or $\operatorname{rank}_{\mathbb{Z}} \mathsf{A}$. Thus, for the remainder of the proof, we assume there are no zero columns

in A, and we also consider values of q large enough so that there are no zero columns in $\mathsf{A}^{\langle q-1\rangle}$ as well.

Define $Z=0$ if $X_1X_2\cdots X_t=0$ and $Z=1$ otherwise. It holds that $\pi=\mathbb{P}\{Z=1\}=(1-1/q)^t$. Moreover, $Z=0$ if and only if any of the monomials $M_1,\dots,M_s$ is zero. Hence, $\mathrm{H}(Z\mid \boldsymbol{M})=0$ and we have

$$\begin{aligned}\mathrm{H}(\boldsymbol{M}) &= \mathrm{H}(Z)+\mathrm{H}(\boldsymbol{M}\mid Z)-\mathrm{H}(Z\mid \boldsymbol{M})\\ &= h(\pi)+(1-\pi)\,\mathrm{H}(\boldsymbol{M}\mid Z=0)+\pi\,\mathrm{H}(\boldsymbol{M}\mid Z=1).\end{aligned}$$

Next, $Z=1$ if and only if none of $X_1,\dots,X_t$ is zero, i.e., all $X_1,\dots,X_t\in\mathbb{F}_q^*$. In this case, we can define $Y_j=\operatorname{dlog}X_j\in\mathbb{Z}_{q-1}$, for $j\in[t]$, $L'_i=\operatorname{dlog}M_i=a_{i1}Y_1+\cdots+a_{it}Y_t\in\mathbb{Z}_{q-1}$, for $i\in[s]$, and $\boldsymbol{L}'=(L'_1,\dots,L'_s)$. Then, $\mathrm{H}(\boldsymbol{L}')=\mathrm{H}(\boldsymbol{M}\mid Z=1)$ and

$$\begin{aligned}&|\,\mathrm{H}(\boldsymbol{M})-\mathrm{H}(\boldsymbol{L}')| = |\,\mathrm{H}(\boldsymbol{M})-\mathrm{H}(\boldsymbol{M}\mid Z=1)|\\ &= |h(\pi)+(1-\pi)\,\mathrm{H}(\boldsymbol{M}|Z=0)+(\pi-1)\,\mathrm{H}(\boldsymbol{M}\mid Z=1)|\\ &\le h(\pi)+(1-\pi)|\,\mathrm{H}(\boldsymbol{M}|Z=0)-\mathrm{H}(\boldsymbol{M}|Z=1)|\\ &\le h(\pi)+s(1-\pi)\log q=o(1),\ \text{as } q\to\infty.\end{aligned}$$

From Corollary 1 with $m=q-1$, we have $\mathrm{H}(\boldsymbol{L}')=r\log(q-1)+O(1)=r\log q+O(1)$, as $q\to\infty$. Finally,

$$\mathrm{H}(\boldsymbol{M})=\mathrm{H}(\boldsymbol{L}')+o(1)=r\log q+O(1),\ \text{as } q\to\infty.\quad\blacksquare$$

Corollary 2. *In the setting of Theorem 1, consider $q=p^k$ with $p\nmid g_r(\mathsf{A})$. Then,*

$$|\mathrm{H}_q(\boldsymbol{M})-\mathrm{H}_q(\boldsymbol{L})|=o(1),\ \textit{as } q\to\infty,$$

where $L_i=a_{i1}X_1+\cdots+a_{it}X_t\in\mathbb{F}_q$ for $i\in[s]$, and $\boldsymbol{L}=(L_1,\dots,L_s)$.[4]

Proof: As A defines a linear transformation of a vector space over $\mathbb{F}_q$, $\mathrm{H}(\boldsymbol{L})=\operatorname{rank}_{\mathbb{F}_q}\mathsf{A}\cdot\log q$. From (1) and since $p\nmid g_r(\mathsf{A})$, we obtain $\operatorname{rank}_{\mathbb{F}_q}\mathsf{A}=\operatorname{rank}_{\mathbb{Z}}\mathsf{A}=r$. Next, from Theorem 1, as $q\to\infty$,

$$|\mathrm{H}_q(\boldsymbol{M})-\mathrm{H}_q(\boldsymbol{L})|=\frac{|\mathrm{H}(\boldsymbol{M})-\mathrm{H}(\boldsymbol{L})|}{\log q}=\frac{O(1)}{\log q}=o(1).\quad\blacksquare$$

Note that we do not require p to be either fixed or infinitely large. However, all primes $p>g_r(\mathsf{A})$ satisfy the requirement $p\nmid g_r(\mathsf{A})$. Corollary 2 states that the entropy of any fixed set of monomials is equal to the entropy of the corresponding set of linear functions (i.e., defined by the same matrix A), both over $\mathbb{F}_q$, when $p\nmid g_r(\mathsf{A})$ and as q approaches infinity. Moreover, this also holds for conditional entropies consisting of various sets of monomials because they can be expressed as a difference of two unconditional entropies. This key observation is further used in Section IV-B.

IV. Achievable Scheme

A. Sun–Jafar Scheme for Private Linear Computation

We build our PMC achievable scheme based on the Sun–Jafar scheme for PLC ([1, Alg. 1], referred to as *PC* there). Due to lack of space, we do not present their scheme in all details and refer the reader to [1] for a full description and analysis. Here, we briefly repeat the facts (in our notation) essential for further discussion.

The Sun–Jafar scheme uses $\lambda=n^\mu$ subpackets. From each of the n databases, the user downloads symbols in μ blocks. The b-th block, $b\in[\mu]$, of each database consists of $(n-1)^{b-1}\binom{\mu}{b}$ symbols, and each symbol is a linear combination (using only coefficients ±1) of b judiciously chosen pieces $\varphi_u(\boldsymbol{X}^{(j)})$ for different values of $u\in[\mu]$ and $j\in[\lambda]$. Since all φ_u are linear combinations, each symbol the user downloads is some linear combination of $\{X_i^{(j)}\}$. The user's randomized queries define which linear combinations the databases will reply with. The queries enforce symmetry across databases and function evaluation symmetry within symbols downloaded from each database. This ensures privacy of the user.

A crucial observation is that $(n-1)^{b-1}\binom{\mu-r}{b}$ of the symbols in block b of each database are redundant based on side information downloaded from other databases. More precisely, these redundant symbols are linear combinations of other symbols in block b from the same database as well as symbols downloaded from other databases. Hence, they need not to be downloaded, as the user can reconstruct them offline. This preserves the user's privacy while reducing the download cost to the value corresponding to the PLC capacity. A distinctive property of the Sun–Jafar scheme is that it is oblivious to the coefficients of the linear functions φ_v. It is only the number of them, μ, that matters. Furthermore, the scheme can be used for PIR if $\mu=f$ and the linear functions are the messages, i.e., $\varphi_i(x_1,\dots,x_f)=x_i$ for $i\in[f]$. In this case, there are no redundant symbols in any block.

B. Private Monomial Computation

Let $\lambda=n^\mu$ and suppose that none of $\{X_i^{(j)}\}$ equals zero. Then we can construct a *multiplicative* scheme by substituting each linear combination of $\{\varphi_v\}$ in the Sun–Jafar scheme with a corresponding multiplicative combination. For example, if at some step the user downloads the symbol $\varphi_1(\boldsymbol{X}^{(j_1)})+\varphi_2(\boldsymbol{X}^{(j_2)})-\varphi_3(\boldsymbol{X}^{(j_3)})$, $j_1,j_2,j_3\in[\lambda]$, then the corresponding multiplicative combination is $\varphi_1(\boldsymbol{X}^{(j_1)})\varphi_2(\boldsymbol{X}^{(j_2)})\big(\varphi_3(\boldsymbol{X}^{(j_3)})\big)^{-1}$, where the functions φ_v now denote the corresponding monomials. Since there are no zeros among $\{X_i^{(j)}\}$, all operations are valid and ensure correct reconstruction of the monomial of interest. Moreover, from Corollary 2, when $p\nmid g_r(\mathsf{A})$ and as $q\to\infty$, the entropies of all the symbols as well as the entropy of each block b conditioned on the side information received from other databases converge to those of the Sun–Jafar scheme. This means that in the multiplicative scheme above, a database can also encode the whole b-th block into no more than $(n-1)^{b-1}\left(\binom{\mu}{b}-\binom{\mu-r}{b}\right)$ q-ary symbols, resulting in the same download cost as in the Sun–Jafar scheme. Since there is only a finite number of entropies involved, we can satisfy the requirement on p from Corollary 2 for all of them simultaneously, e.g., by requiring p to be large enough (but not necessarily approaching infinity).

[4] In contrast to Lemma 2 and Corollary 1, $\boldsymbol{L}$ is defined over the field.

Now, in case any of $\{X_i^{(j)}\}$ equals zero, we can ignore dependencies between the monomials and run a PIR scheme, for example, the same Sun–Jafar scheme in PIR mode for μ messages. Altogether, our scheme is as follows.

Algorithm 1: PMC Scheme

1 **if** *there are no zeros among* $\{X_i^{(j)}\}$ *and* $\mu > r$ **then**
2 Each database replies according to the multiplicative scheme.
3 **else**
4 Each database replies according to the Sun–Jafar scheme in PIR mode oblivious to the dependencies between the monomials.

Note that the queries of both schemes need to be uploaded since the user does not know if there are zeros among $\{X_i^{(j)}\}$. Moreover, the user can determine which scheme is used (Line 2 or Line 4) from (r, μ) and the size of the responses (the size is smaller for the multiplicative scheme provided $r < \mu$).

We note that privacy of the user in the suggested PMC scheme is inherited from the privacy of the Sun–Jafar scheme.

Theorem 2. *For PMC with n databases, f messages, and μ monomials defined by a degree matrix $\mathsf{A} \in \mathbb{Z}^{\mu \times f}$ of rank $r = \operatorname{rank}_{\mathbb{Z}} \mathsf{A}$, for $p \nmid g_r(\mathsf{A})$ and as $q \to \infty$, the PMC capacity converges to that of PIR: $\mathrm{C}_{\mathrm{PMC}}(n, f, \mu, \mathsf{A}, q) \to \mathrm{C}_{\mathrm{PIR}}(n, r)$.*

Proof: First, we show that the PC rate $\mathrm{C}_{\mathrm{PIR}}(n, r)$ is achievable by Algorithm 1. For Line 2, for $p \nmid g_r(\mathsf{A})$ and as $q \to \infty$, the download cost measured in q-ary units converges to $n^\mu / \mathrm{C}_{\mathrm{PLC}}(n, r) = n^\mu / \mathrm{C}_{\mathrm{PIR}}(n, r)$. The download cost at Line 4 is $n^\mu / \mathrm{C}_{\mathrm{PIR}}(n, \mu)$.

The probability that none of $\{X_i^{(j)}\}$ equals zero is $\pi = (1-1/q)^{n^\mu f} \to 1$, as $q \to \infty$. Therefore, the average download cost of Algorithm 1 becomes

$$n^\mu \left(\frac{\pi}{\mathrm{C}_{\mathrm{PIR}}(n, r)} + \frac{1-\pi}{\mathrm{C}_{\mathrm{PIR}}(n, \mu)} \right) \xrightarrow[q \to \infty]{} \frac{n^\mu}{\mathrm{C}_{\mathrm{PIR}}(n, r)}.$$

On the other hand, from Lemma 3, it follows that

$$\min_{v \in [\mu]} \mathrm{H}_q(\boldsymbol{F}_v) = n^\mu \cdot \min_{v \in [\mu]} \mathrm{H}_q(\varphi_v(\boldsymbol{X}^{(1)})) \xrightarrow[q \to \infty]{} n^\mu.$$

Altogether, we have that the download rate of our PMC scheme converges to the PIR capacity for r messages.

It remains to prove the converse, i.e., showing that $\mathrm{C}_{\mathrm{PIR}}(n, r)$ is an upper (or outer) bound on the PC capacity. For that, we consider the general converse in [6, Thm. 1] and show that, for $q \to \infty$ and provided $p \nmid g_r(\mathsf{A})$, the upper bounds from [6, Thm. 1] coincide for the monomial and linear cases with the same matrix A. Note that [6, Thm. 1] gives $\mu!$ upper bounds on the PC capacity (according to the number of permutations of μ functions). For the linear case, the outer bounds in [6, Thm. 1] reduce to $\mathrm{C}_{\mathrm{PIR}}(n, r)$, independent of q. In general, for a fixed permutation, the bound depends on $\min_{v \in [\mu]} \mathrm{H}_q(\varphi_v(\boldsymbol{X}^{(1)}))$ and joint entropies of different subsets of function evaluations. Then, it follows from the key observation in Section III that this bound is coinciding for the monomial and linear cases as $q \to \infty$, provided $p \nmid g_r(\mathsf{A})$ (details omitted for brevity). ∎

Corollary 3. *In the setting of Theorem 2, the scheme in Algorithm 1 is capacity-achieving for $p \nmid g_r(\mathsf{A})$ and as $q \to \infty$.*

Note that we prove that the scheme in Algorithm 1 is capacity-achieving only for asymptotic q and provided $p \nmid g_r(\mathsf{A})$. As an example, take $\mu = f = 2$, $n = 2$, $\varphi_1(x_1, x_2) = x_1^2 x_2$, and $\varphi_2(x_1, x_2) = x_1 x_2^2$. Then the asymptotic PC rate of Corollary 3 is $\mathrm{C}_{\mathrm{PIR}}(2, 2) = 2/3$, since $r = \operatorname{rank}_{\mathbb{Z}} \mathsf{A} = 2$. On the other hand, the PC capacity C_{PC} for two arbitrary functions for any finite field is known [1, Sec. VII, Eq. (82)]. For this example, $\mathrm{C}_{\mathrm{PC}} = 2\,\mathrm{H} / (\mathrm{H}(X_1^2 X_2, X_1 X_2^2) + \mathrm{H})$, where $\mathrm{H} \triangleq \mathrm{H}(X_1^2 X_2) = \mathrm{H}(X_1 X_2^2)$ and the superscripts on the X's have been suppressed for brevity. Finally, Algorithm 1 defaults to PIR mode and achieves the PC rate $2\,\mathrm{H}/3$, which can be shown to be smaller than C_{PC} for any finite q.

V. Conclusion

We derived the PMC capacity for replicated noncolluding databases, by considering the case of an arbitrary large field and under a technical condition on the size p of the base field, which is satisfied, e.g., for p large enough. A PMC scheme that is capacity-achieving in the above asymptotic case was also outlined. Furthermore, we presented formulas for the entropy of a multivariate monomial and for a set of monomials in uniformly distributed random variables over a finite field.

Acknowledgment

The authors would like to thank Srimathi Varadharajan and Alessandro Melloni for useful discussions.

References

[1] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.
[2] M. Mirmohseni and M. A. Maddah-Ali, "Private function retrieval," in *Proc. Iran Workshop Commun. Inf. Theory (IWCIT)*, Tehran, Iran, Apr. 25–26, 2018, pp. 1–6.
[3] S. A. Obead and J. Kliewer, "Achievable rate of private function retrieval from MDS coded databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2117–2121.
[4] 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, pp. 813–820.
[5] ——, "Private polynomial computation for noncolluding coded databases," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1677–1681.
[6] ——, "On the capacity of private nonlinear computation for replicated databases," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 25–28, 2019, pp. 1–5.
[7] D. Karpuk, "Private computation of systematically encoded data with colluding servers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, CO, USA, Jun. 17–22, 2018, pp. 2112–2116.
[8] N. Raviv and D. A. Karpuk, "Private polynomial computation from Lagrange encoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1672–1676.
[9] N. Jacobson, *Basic Algebra I*, 2nd ed. Freeman and Company, 1985.
[10] C. Norman, *Finitely Generated Abelian Groups and Similarity of Matrices over a Field*. Springer Science & Business Media, 2012.
[11] 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.
[12] U. Dudley, *Elementary Number Theory*, 2nd ed. Freeman and Company, 1978.