On the Capacity of Locally Decodable Codes

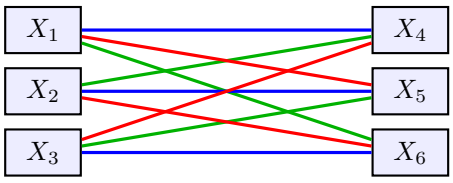
Hua Sun, *Member, IEEE,* and Syed Ali Jafar, *Fellow, IEEE*

***Abstract*—A locally decodable code (LDC) maps K source symbols, each of size L_w bits, to M coded symbols, each of size L_x bits, such that each source symbol can be decoded from $N \leq M$ coded symbols. A perfectly smooth LDC further requires that each coded symbol is uniformly accessed when we decode any one of the messages. The ratio L_w/L_x is called the symbol rate of an LDC. The highest possible symbol rate for a class of LDCs is called the capacity of that class. It is shown that given K, N, the maximum value of capacity of perfectly smooth LDCs, maximized over all code lengths M, is $C^* = N\left(1+1/N+1/N^2+\cdots+1/N^{K-1}\right)^{-1}$. Furthermore, given K, N, the minimum code length M for which the capacity of a perfectly smooth LDC is C^* is shown to be $M = N^K$. Both of these results generalize to a broader class of LDCs, called universal LDCs. The results are then translated into the context of $\text{PIR}_{\max}$, i.e., Private Information Retrieval subject to maximum (rather than average) download cost metric. It is shown that the minimum upload cost of capacity achieving $\text{PIR}_{\max}$ schemes is $(K-1)\log N$. The results also generalize to a variation of the PIR problem, known as Repudiative Information Retrieval (RIR).**

***Index Terms*—Capacity, locally decodable codes, private information retrieval.**

I. Introduction

A locally decodable code (LDC) with locality N is a mapping from K source symbols, $\mathcal{W} = \{W_1, W_2, \cdots, W_K\}$, each of size L_w bits, to M coded symbols, $\mathcal{X} = \{X_1, X_2, \cdots, X_M\}$, each of size L_x bits, such that for every source symbol W_k, there exists at least one subset of N coded symbols, $S \subset \mathcal{X}$, $|S| = N$, such that W_k can be recovered from the elements of S. Such a set S is called a decoding set for W_k. This basic definition is somewhat trivial, for example, any systematic code is locally decodable with locality $N = 1$. LDCs are useful primarily if they are capable of withstanding a significant fraction of corrupted coded symbols without losing their local decodability. An $(N, \delta, 1-\epsilon)$ LDC is guaranteed to have locality N and a randomized decoding algorithm that succeeds with probability at least $1-\epsilon$ when the fraction of corrupted coded symbols is at most δ. For this to be meaningful, there must be multiple decoding sets for each source symbol. Let $\mathcal{S}_k$ be the set of decoding sets for source symbol W_k, so that if $S \in \mathcal{S}_k$ then $S \subset \mathcal{X}$, $|S| = N$, and W_k is decodable from S. An LDC is said to be *perfectly smooth* if the coded symbols are uniformly distributed across decoding sets. Specifically, $\forall m_1, m_2 \in \{1, 2, \cdots, M\}$, and $\forall k \in \{1, 2, \cdots, K\}$, the number of decoding sets in $\mathcal{S}_k$ that contain X_{m_1}, must be equal to the number of decoding sets in $\mathcal{S}_k$ that contain X_{m_2}. If there are $|\mathcal{S}_k|$ decoding sets for W_k in a perfectly smooth LDC (SLDC) with locality N, then every coded symbol must appear in exactly $N|\mathcal{S}_k|/M$ of them. For such a code, at least one uncorrupted decoding set survives as long as the fraction of corrupted coded symbols, δ, is less than $1/N$. This is because each corrupted coded symbol can corrupt at most $N|\mathcal{S}_k|/M$ decoding sets in $\mathcal{S}_k$. If δM coded symbols are corrupted, then the number of decoding sets that are corrupted is no more than $\delta N|\mathcal{S}_k|$. So a decoding algorithm that randomly chooses one of the decoding sets must be successful with probability at least $1-\delta N$, provided that $\delta < 1/N$. Therefore, an SLDC is an $(N, \delta, 1-\delta N)$ LDC for any $\delta < 1/N$. By the same token, the minimum distance d of an SLDC, i.e., the minimum number of coded symbols that must be erased for a loss of data to occur, is at least M/N. Figure 1 shows an example of an SLDC with locality $N = 2$ that encodes $K = 3$ binary ($L_w = 1$) source symbols, W_1, W_2, W_3, into $M = 6$ binary ($L_x = 1$) coded symbols, $X_1, \cdots, X_6$. The decoding sets for W_1, W_2, W_3 are comprised of pairs of coded symbols connected by blue, red, and green edges, respectively. This is also a $(2, \delta, 1-2\delta)$ LDC for $\delta < 1/2$. So if $\delta = 1/3$, and any two coded symbols X_i, X_j are corrupted, then at least one of the three decoding sets remains uncorrupted for every source symbol, and a randomized decoder succeeds with probability at least $1-\delta N = 1/3$. The minimum distance of this code is $d = M/N = 3$ because, e.g., a loss of X_1, X_5, X_6 causes a loss of data (W_1 is lost).



$$X_1 = W_1, \quad X_4 = W_2 + W_3$$
$$X_2 = W_2, \quad X_5 = W_1 + W_2$$
$$X_3 = W_3, \quad X_6 = W_3 + W_1$$

Fig. 1. *An SLDC with locality $N = 2$ that encodes $K = 3$ binary ($L_w = 1$) source symbols, W_1, W_2, W_3, into $M = 6$ binary ($L_x = 1$) coded symbols, $X_1, \cdots, X_6$.*

LDCs were introduced in the year 2000 by Katz and

This work was supported in part by NSF Grants CCF-1317351, CCF-1617504, CNS-1731384, ONR Grant N00014-18-1-2057 and by ARO Grant W911NF1910344.

H. Sun is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203, USA (email: hua.sun@unt.edu).

S. A. Jafar is with the Center for Pervasive Communications and Computing, Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, CA 92697, USA (email: syed@uci.edu).

Trevisan in [1][1]. One of the motivations for studying LDCs comes from distributed storage applications. Coding is used in distributed storage systems to limit storage and decoding costs while providing resilience against failures of storage nodes and efficient repair when such failures occur. LDCs are especially effective for reducing the decoding cost in commonly encountered scenarios where multiple datasets are jointly encoded and only one of them needs to be retrieved. In particular, smoothness of LDCs is a desirable feature for distributed storage because it minimizes risk by spreading it evenly across storage nodes. Remarkably, LDCs play even more important roles in complexity theory [2], [3, Chapters 17, 18], data structures [4], [5], fault tolerant computation [6], multiparty computation [7] and private information retrieval (PIR) [8], [9], [10]. As such, understanding the fundamental limits of LDCs (especially the tradeoff between code length M and locality N) is recognized as a major open problem in theoretical computer science [7], whose answer could have a domino effect on a number of related problems. For further details on LDCs, we refer to the excellent tutorials in [11], [12] and references therein.

In this work we view this open problem through the lens of PIR. In its basic form [8], PIR is the problem of efficiently retrieving a desired message from a set of K messages that are replicated across N non-colluding databases, without disclosing any information about the identity of the desired message to any individual database. The strong connection between PIR and LDCs is evident from the example illustrated in Figure 1. In fact the example is derived from a PIR scheme with $K = 3$ messages, W_1, W_2, W_3, and two databases that store (X_1, X_2, X_3) and (X_4, X_5, X_6), respectively. The user randomly asks Database 1 for one of X_1, X_2 or X_3, and asks Database 2 for the other element of the decoding set for his desired message, which is also uniformly distributed over X_4, X_5, X_6, thus revealing no information to either database about which message is being retrieved. The upload cost for this PIR scheme is a 3-ary symbol per database. Interestingly, as shown in [13], the capacity of PIR subject to this upload cost is $1/2$, so the scheme shown in Figure 1 is optimal among all PIR schemes with the same upload constraint.

In particular, this work is motivated by recent capacity characterizations of PIR with various assumptions on message sets, storage, and upload costs [13], [14], [15], [16], [17], [18], [19]. The capacity of PIR, $C_{\text{PIR}}(N, K)$, is the maximum number of bits of desired message that can be retrieved per bit of total download from the N databases. Defining $R_s = L_w/L_x$ as the *symbol rate* of an LDC, the corresponding notion of capacity, $C_{\text{LDC}}(M, N, K)$, is the maximum symbol rate that is feasible for an LDC given the locality parameter N, the code length M, and the number of source symbols K. From this perspective, the fundamental tradeoff for SLDCs is expressed in terms of the 4 parameters: M, N, K, R_s. It is desirable for M, N to take smaller values, and for K, R_s to take larger values. The rate R_s is a critical part of this tradeoff. If we consider M, K as independently chosen natural numbers, then the range of values of N is between 1 and M, while the range of values of R_s is between $1/K$ and M/K. At one extreme, $N = 1$ forces $R_s = 1/K$. This is because $N = 1$ for an SLDC implies that *all* source symbols can be decoded from any single coded symbol. At the other extreme, $R_s = M/K$ forces $N = M$, because there is no redundancy, i.e., the total number of bits of all coded symbols is the same as the total number of bits of all source symbols.

In this paper we explore two particular aspects of the (M, N, K, R_s) tradeoff[2]. The first is the tradeoff between N, K, R_s for unconstrained M. In other words, we identify the capacity of an SLDC for arbitrary N, K and unconstrained code length M. Specifically we show that,

$$\begin{aligned} C^*(N, K) &\triangleq \max_{M \in \mathbb{N}} C_{\text{LDC}}(M, N, K) \\ &= N \left(1 + \frac{1}{N} + \cdots \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}\right)^{-1} \end{aligned} \tag{1}$$

The second aspect of the tradeoff that we characterize is the minimum codeword length M^* that is needed to achieve $C^\star(N, K)$ for arbitrary N, K. Specifically, we show that $M^* = N^K$. Remarkably, both results are shown not only for all SLDCs but also for a broader class of LDCs that we label *universal* LDCs (ULDCs). An LDC is *universal* if every coded symbol appears in at least one of the decoding sets of every source symbol. Mathematically, a ULDC is defined by the property that $\forall m \in \{1, 2, \cdots, M\}$, and $\forall k \in \{1, 2, \cdots, K\}$, there exists some $S \in \mathcal{S}_k$ such that $X_m \in S$. Clearly, every SLDC is a ULDC. However, not every ULDC is an SLDC. For example, the LDC that maps $K = 3$ binary source symbols W_1, W_2, W_3 to the $M = 4$ binary code symbols $W_1, W_2, W_3, W_2 + W_3$ with locality $N = 2$ and decoding sets $\mathcal{S}_1 = \{\{W_1, W_2\}, \{W_1, W_3\}, \{W_1, W_2 + W_3\}\}$, $\mathcal{S}_2 = \{\{W_1, W_2\}, \{W_2, W_3\}, \{W_3, W_2 + W_3\}\}$ and $\mathcal{S}_3 = \{\{W_1, W_3\}, \{W_2, W_3\}, \{W_2, W_2 + W_3\}\}$, is universal but not perfectly smooth. While less structured than SLDCs, evidently ULDCs retain all the structure needed for the two aspects of the tradeoff that are explored in this work.

For our final result, we apply the new insights from the study of fundamental limits of LDCs back to the problem of PIR. Recall that the rate of a PIR scheme is defined as $R_p = \frac{L_w}{ND}$, where L_w is the number of bits of each message, N is the number of databases, and D is the number of bits downloaded from each database. For most PIR capacity results [13], [16], [19], [20] the parameter D may be interpreted either as the *average* download per database or as the *maximum* download from any database (maximized across all databases and all queries), without changing the capacity. This is because the normalized downloads for almost all PIR

[1]In [1], Katz and Trevisan introduced $(N, \delta, 1 - \epsilon)$ LDCs and smooth LDCs (which include perfectly smooth LDCs as special cases). It is noted later in Section 3.2 of [2] that a perfectly smooth LDC produces an $(N, \delta, 1 - \delta N)$ LDC for every $\delta < 1/N$, and that for constant locality N (the setting considered in this work) all known constructions of LDCs and PIR schemes follow from the constructions of perfectly smooth LDCs.

[2]Prior work in theoretical computer science literature [1], [2], [11] typically explores a different regime where R_s is fixed ($R_s = 1$ is commonly assumed), and studies the tradeoff between the number of source symbols K and the number of coded symbols M for various values of locality parameter N (including scaling of N with K).

schemes are either already identical across databases or can be made identical by time-sharing across different permutations of databases. Exceptions include [15] which admits only the maximum download formulation and [14] which allows only the average download formulation. Reference [15] considers the capacity of PIR for fixed length messages, and relies on the maximum download formulation because averages are less meaningful over the finite horizon. Reference [14] on the other hand considers the minimum upload cost of a capacity achieving PIR scheme, and allows only the average download formulation because the PIR scheme is asymmetric and the usual approach of making the scheme symmetric with time-sharing arguments does not work (does not preserve the upload cost). When PIR is viewed in relation to LDCs, the natural interpretation of D is the maximum download across all databases and all queries,[3] which corresponds to L_x in the corresponding LDC setting. To make the distinction clear, we refer to PIR with the maximum download metric as $\text{PIR}_{\max}$, and PIR with the average download metric as PIR_{ave}. Using insights from LDCs, we determine the minimum upload cost needed to achieve the capacity of $\text{PIR}_{\max}$. Specifically, we show that the minimum upload for *any* capacity achieving $\text{PIR}_{\max}$ scheme, linear or non-linear, is $(K-1)\log N$ bits per database, i.e., the user must upload a q-ary symbol per database where q is at least N^{K-1}. Our result complements the result of [14] which shows that the minimum upload cost for capacity achieving PIR_{ave} schemes is also $(K-1)\log N$ bits per database, although the optimality in [14] is established only within a restricted class of decomposable (e.g., linear) schemes. Remarkably, while the capacity and minimum upload cost characterizations are identical for $\text{PIR}_{\max}$ and PIR_{ave}, the mapping between the corresponding PIR schemes turns out to be highly non-trivial. Furthermore, just as our results for SLDCs generalize to ULDCs, by the same token we show that both the capacity and the minimum upload cost are unaffected if the privacy constraint is relaxed in the $\text{PIR}_{\max}$ problem formulation from perfect privacy to a weaker *deniability* condition. Perfect privacy implies that the query to each database must not reveal any information about the user's desired message index. Deniability only implies that the query does not absolutely rule out any message from being the user's desired message, i.e., even if some messages are revealed by the query to be more likely to be the desired message than others, each message has a non-zero probability of being the desired message. Information retrieval under a deniability constraint is called *Repudiative* information retrieval (RIR) in [21]. Surprisingly, under the maximum download formulation, $\text{PIR}_{\max}$ and $\text{RIR}_{\max}$ have the same[4] capacity, and the same minimum upload cost.

Notation: For positive integers n_1, n_2, with $n_1 \leq n_2$, we use the notation $[n_1 : n_2]$ to represent the set $\{n_1, n_1+1, \cdots, n_2\}$. For a set A, $|A|$ denotes its cardinality and X_A represents the set $\{X_i, i \in A\}$. For two random variables X, Y, the notation $X \sim Y$ denotes that X and Y are identically distributed. If X and Y are sets of random variables, then the conditional entropy $H(X \mid Y)$ refers to the joint entropy of all the random variables in X, conditioned on all the random variables in Y.

II. Problem Statement and Preliminaries

A. Locally Decodable Codes (LDC)

Definition 1 (Set of Source Symbols, $\mathcal{W}$)**.** *Define $\mathcal{W} = \{W_1, \cdots, W_K\}$ as a set of K independent source symbols, each of size L_w bits,*

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \tag{2}$$

$$L_w = H(W_1) = \cdots = H(W_K). \tag{3}$$

Definition 2 (Set of Coded Symbols, $\mathcal{X}$)**.** *Define $\mathcal{X} = \{X_1, X_2, \cdots, X_M\}$ as a set of M coded symbols each of size L_x bits,*

$$L_x = H(X_1) = \cdots = H(X_M). \tag{4}$$

Note that L_x and L_w are not necessarily integer values. For example, if W_i are uniformly random 3-ary symbols, then $L_w = \log(3)$ bits. Furthermore, both L_w and L_x are allowed to take arbitrarily large values, since it is only their relative size that matters (see Definition 6). Indeed, in typical applications, such as distributed storage, each source symbol may represent a large dataset and each coded symbol may represent all data stored in one storage node. Measuring the size of each symbol by its entropy is especially meaningful for large symbols which can be optimally compressed.

Definition 3 (LDC $(\mathcal{C}, \mathcal{S}_{[1:K]})$)**.** *An LDC $(\mathcal{C}, \mathcal{S}_{[1:K]})$ with locality N is comprised of a mapping $\mathcal{C}$ from $(W_1, \cdots, W_K)$ to $(X_1, \cdots, X_M)$, and K non-empty sets $\mathcal{S}_k, k \in [1:K]$, called decoding supersets. Elements of the decoding superset $\mathcal{S}_k$ are called decoding sets of the source symbol W_k. Each decoding set of W_k is itself a set S containing N coded symbols from which W_k can be recovered.*

$$S \in \mathcal{S}_k \Rightarrow \begin{cases} S & \subset \mathcal{X}, \\ |S| & = N, \\ H(W_k \mid S) & = 0. \end{cases} \tag{5}$$

Definition 3 is useful only as a baseline upon which the definitions of more interesting types of LDCs can be built. The most interesting type of LDCs for our purpose are perfectly smooth LDCs, defined next.

Definition 4 (Perfectly Smooth LDC (SLDC))**.** *An LDC is said to be perfectly smooth if for all $k \in [1:K]$, a uniform choice of a decoding set from $\mathcal{S}_k$ implies that each coded symbol is equally likely to be in the chosen decoding set. Equivalently, $\forall m, m' \in [1:M]$ and $\forall k \in [1:K]$,*

$$|\{S \mid S \in \mathcal{S}_k, X_m \in S\}| = |\{S \mid S \in \mathcal{S}_k, X_{m'} \in S\}| \tag{6}$$

[3] Equivalently, the size of the download from each database n is fixed at the same constant value, D, for all queries and all databases, $n \in \{1, 2, \cdots, N\}$.

[4] Under the average download formulation, the capacity of PIR_{ave} is not the same as the capacity of RIR_{ave}. In particular, the capacity of RIR_{ave} is trivially seen to be 1 if the number of databases is $N > 1$. For example, let (i, j) be a random permutation of $(1, 2)$ generated privately by the user. The user downloads his desired message W_θ from Database i. With probability ϵ the user downloads a randomly chosen undesired message $W_{\theta'}$ from Database j. It is easy to verify that the scheme is valid for RIR, and that the rate achieved under the average download formulation with this scheme is $1/(1+\epsilon)$ which approaches 1 as $\epsilon \to 0$. If $N = 1$ then the capacity of RIR is $1/K$, same as PIR, under both average and maximum download formulations.

Thus, in an SLDC, every coded symbol appears in the same number of decoding sets for any given source symbol. While SLDCs are most commonly encountered in various applications of LDCs, it is useful to also define a broader class of LDCs, called universal LDCs.

Definition 5 (Universal LDC (ULDC))**.** *An LDC is said to be universal if every coded symbol $X_m, m \in [1:M]$ appears in at least one of the decoding sets of every source symbol $W_k, k \in [1:K]$.*

$$\forall m \in [1:M],\ \forall k \in [1:K],\ \exists S \in \mathcal{S}_k \text{ such that } X_m \in S. \quad (7)$$

Note that an SLDC is universal by definition.

Definition 6 (Symbol Rate and Capacity)**.** *The symbol rate of an LDC is defined as,*

$$R_s = \frac{L_w}{L_x}, \quad (8)$$

and the supremum of R_s values achievable within a class of LDCs is called the capacity of that class of LDCs.

For example, it may be of interest to find the capacity of the class of SLDCs for given values of locality parameter N, the number of source symbols K, and the code length M. Another important quantity of interest is the code rate of an LDC,

$$R_c = \frac{KL_w}{ML_x} \quad (9)$$

which measures the redundancy of the code. Note that $R_c = \frac{K}{M}R_s$.

B. Private Information Retrieval ($\text{PIR}_{\max}$)

Instead of repeating the definition of the PIR problem from, say [13], let us present it through the following definitions that are analogous to the corresponding notions in the context of LDCs. As much as possible we will use the same notation for corresponding quantities to make their relationship obvious.

Definition 7 (Set of Messages, $\mathcal{W}$)**.** *Define $\mathcal{W} = \{W_1, W_2, \cdots, W_K\}$ as the set of K independent messages, each of size L_w bits.*

$$H(W_1, \cdots, W_K) = H(W_1) + \cdots + H(W_K), \quad (10)$$
$$L_w = H(W_1) = \cdots = H(W_K). \quad (11)$$

Definition 8 (Sets of Answers, $\mathcal{X}$, $\mathcal{X}^{[1:N]}$, Upload Cost)**.** *Define sets $\mathcal{X}^{[n]} = \{X_1^{[n]}, X_2^{[n]}, \cdots, X_{M_n}^{[n]}\}$ containing all possible answers from Database n, $n \in [1:N]$, such that all answers have the same size, L_x.*

$$L_x = H(X_m^{[n]}), \qquad \forall n \in [1:N], m \in [1:M_n].$$

The upload cost for Database n, is defined to be $\log(M_n)$ for all $n \in [1:N]$. Furthermore, define

$$\mathcal{X} = \bigcup_{n \in [1:N]} \mathcal{X}^{[n]} \quad (12)$$

as the set of all answers.

Note that we assume all answers have the same size. Under 'maximum download' formulation of PIR, there is no loss of generality in this assumption because the rate of a PIR scheme is limited only by the largest possible download (answer) from any database for any query. If different possible answers have different lengths, then smaller answers can be padded with useless information to match the length of the biggest answer (maximum download).

Definition 9 (IR $(\mathcal{A}, \mathcal{S}_{[1:K]})$)**.** *An N-query Information Retrieval scheme is comprised of a mapping $\mathcal{A}$ from the set of messages $\mathcal{W}$ to the sets of answers $\mathcal{X}^{[1:N]}$, and K non-empty sets, $\mathcal{S}_k$, $k \in [1:K]$, called decoding supersets. Elements of the decoding supserset $\mathcal{S}_k$, are called decoding sets for the message W_k. Each decoding set for W_k is of the form $S = \{X_{q_1}^{[1]}, X_{q_2}^{[2]}, \cdots, X_{q_N}^{[N]}\}$ with $q_n \in [1:M_n], \forall n \in [1:N]$ such that*

$$S \in \mathcal{S}_k \Rightarrow H(W_k \mid S) = 0, \quad \forall k \in [1:K]. \quad \text{[Correctness]} \quad (13)$$

The parameter N is recognized as the number of databases. The elements of the decoding set, $X_{q_n}^{[n]}$ represent what is requested by the user from the n^{th} database, i.e., the query sent to Database n is q_n and the answer received from Database n is $X_{q_n}^{[n]}$. If the desired message is W_θ, then a decoding set is chosen from $\mathcal{S}_\theta$. Condition (13) is called the 'correctness' condition, because it guarantees that the message can be decoded correctly from the answers received from all N databases. Definition 9 is useful only as a baseline for introducing more interesting forms of information retrieval. The most interesting for our purpose is perfectly private information retrieval, or simply PIR.

Definition 10 (Perfectly Private Information Retrieval ($\text{PIR}_{\max}$))**.** *A PIR scheme is an N-query Information Retrieval scheme with a distribution defined on the elements of each decoding superset (so we have K distributions, one for each decoding superset), such that for all $n \in [1:N]$, and for all $k, k' \in [1:K]$ the conditional distribution of q_n given $S \in \mathcal{S}_k$ is identical to the conditional distribution of q_n given $S \in \mathcal{S}_{k'}$.*

$$\text{Prob}(q_n = q \mid S \in \mathcal{S}_k) = \text{Prob}(q_n = q \mid S \in \mathcal{S}_{k'}),$$
$$\forall k, k' \in [1:K], n \in [1:N], \forall q \in [1:M_n]. \quad (14)$$

Equation (14) ensures perfect privacy for the desired message index, because the query sent to any database has the same distribution regardless of the desired message index. It is useful to also define a broader class of N-query Information Retrieval schemes, called Repudiative Information Retrieval (RIR), which includes PIR as a special case.

Definition 11 (Repudiative Information Retrieval ($\text{RIR}_{\max}$))**.** *An RIR scheme is an N-query Information Retrieval scheme such that every possible answer from every database appears in at least one of the decoding sets of every $\mathcal{S}_k, k \in [1:K]$.*

$$\forall n \in [1:N], \forall m \in [1:M_n],\ \forall k \in [1:K],$$
$$\exists S \in \mathcal{S}_k \text{ such that } X_m^{[n]} \in S. \quad (15)$$

Definition 12 (Rate and Capacity)**.** *The rate of an N-query information retrieval scheme is defined as*

$$R = \frac{L_w}{NL_x} \tag{16}$$

and the supremum of R values for a class of information retrieval schemes is called the capacity of that class.

C. Connection between ULDCs and RIR$_{\max}$

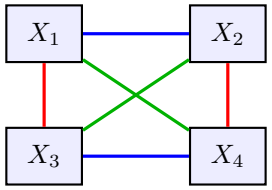
It is well known that LDCs and PIR schemes are closely related [10]. Comparing preceding definitions for LDCs with locality N and N-query information retrieval, it is evident that source symbols correspond to messages, coded symbols correspond to answers, code length corresponds to total upload cost, SLDCs correspond to PIR$_{\max}$, the relaxation to ULDCs correspond to the relaxation to RIR$_{\max}$, and the decoding sets, rates and capacity expressions for both settings are similar as well. However, a closer look also reveals clear differences. For example, answers are partitioned into $\mathcal{X}^{[n]}, n \in [1:N]$, whereas no such partitioning is invoked for coded symbols. While both SLDCs and PIR$_{\max}$ impose additional constraints on the decoding sets, the two constraints are not equivalent. These distinctions often do not matter much in practice, indeed most PIR$_{\max}$ schemes produce SLDCs and most constructions of SLDCs are obtained from PIR$_{\max}$ schemes. Nevertheless, the distinctions pose difficulties in translating theoretical results between the two problems. For our purpose, the precise connection[5] (obvious from the preceding definitions) that allows us to connect our results across the two settings is between ULDCs and RIR$_{\max}$, as stated below.

Observation 1. *The set of all answers $\mathcal{X}$ from an RIR$_{\max}$ scheme with message set $\mathcal{W}$, N databases, upload costs $\log(M_{[1:N]})$, decoding supersets $\mathcal{S}_{[1:K]}$ and rate R, constitutes a ULDC with set of source symbols $\mathcal{W}$, coded symbols $\mathcal{X}$, locality N, code length $M = \sum_{n\in[1:N]} M_n$, decoding supersets $\mathcal{S}_{[1:K]}$, and symbol rate $R_s = NR$.*

Given the translation from RIR$_{\max}$ to ULDCs, one might be interested in the other direction, i.e., the translation from ULDCs to RIR$_{\max}$, which is also possible, although in general less efficient. For example, by choosing the sets of answers $\mathcal{X}^{[n]}, n \in [1:N]$, to be *each* identical to the set of coded symbols $\mathcal{X}$ of a ULDC, an RIR$_{\max}$ scheme is trivially obtained. This is less efficient because of the *expansion* by the factor N, i.e., the value of $\sum_{n\in[1:N]} M_n$ for the resulting RIR$_{\max}$ scheme is N times larger than the code length M of the ULDC. Note that no such expansion occurs in the reverse direction. Interestingly, as illustrated in Figure 2 through an example, an expansion by a factor of N is *necessary* in some cases when translating a ULDC into an RIR$_{\max}$ scheme.

Note that since ULDCs and RIR$_{\max}$ are relaxations of SLDCs and PIR$_{\max}$, respectively, impossibility results (converse arguments) for ULDCs and RIR$_{\max}$ apply to SLDCs and PIR$_{\max}$ automatically, while achievable schemes for the SLDCs and PIR$_{\max}$ apply automatically to ULDCs and RIR$_{\max}$. These inclusions will be useful to prove our main results, presented in the next section.

[5]This may be viewed as an extension of the corresponding connections between SLDCs and PIR$_{\max}$ (e.g., see Section 3.2 of [2] and Lemma 7.2 of [11]).



$$W_1 = (a_1, a_2, a_3, a_4)$$
$$W_2 = (b_1, b_2, b_3, b_4)$$
$$W_3 = (c_1, c_2, c_3, c_4)$$
$$X_1 = (a_1, a_2, b_1, b_2, c_1, c_2)$$
$$X_2 = (a_3, a_4, b_1, b_3, c_1, c_3)$$
$$X_3 = (a_1, a_3, b_3, b_4, c_2, c_4)$$
$$X_4 = (a_2, a_4, b_2, b_4, c_3, c_4)$$

Fig. 2. *A ULDC (also an SLDC) with locality $N = 2$ that encodes $K = 3$ source symbols with $L_w = 4$ bits each, W_1, W_2, W_3, into $M = 6$ coded symbols, X_1, X_2, X_3, X_4, with $L_x = 6$ bits each. The decoding sets for W_1, W_2, W_3 are comprised of pairs of coded symbols connected by blue, red, and green edges, respectively. It is easy to see that the only RIR$_{\max}$ scheme that can be constructed from this ULDC is with answer sets $\{X_1, X_2, X_3, X_4\}$ replicated at the $N = 2$ databases. Therefore, the total number of answers is 8, $N = 2$ times the ULDC length, i.e., we have an expansion by a factor of $N = 2$.*

III. Main Results

A. Capacity Results

Our first set of results are capacity characterizations. Given K source symbols, code length M, and locality N, let $C_{\text{SLDC}}(N, K, M)$ and $C_{\text{ULDC}}(N, K, M)$ denote the capacity for the class of SLDCs and ULDCs respectively. Our first result characterizes the maximum possible capacity of a ULDC given the locality N and the number of source symbols K. The maximum is over all possible codeword lengths M.

Theorem 1.

$$\begin{aligned} C^*_{\textit{ULDC}}(N, K) &\triangleq \max_{M\in\mathbb{N}} C_{\textit{ULDC}}(N, K, M) \\ &= N\left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1}. \end{aligned} \tag{17}$$

The expression for $C^*_{\text{ULDC}}(N, K)$ is reminiscent of the capacity of PIR [13]. Indeed, since the capacity achieving PIR schemes in [13] naturally produce SLDCs, and all SLDCs are also ULDCs, the achievability argument is directly implied. However, since ULDCs are a more general class of objects than the LDCs produced by PIR schemes, the converse from [13] does not apply. Instead, a new combinatorial converse proof is presented for Theorem 1 in Section IV. As an immediate corollary, we settle the corresponding question for SLDCs as well.

Corollary 1.1.

$$\begin{aligned} C^*_{\textit{SLDC}}(N, K) &\triangleq \max_{M\in\mathbb{N}} C_{\textit{SLDC}}(N, K, M) \\ &= N\left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1}. \end{aligned} \tag{18}$$

The achievability argument for Corollary 1.1 follows from the capacity achieving PIR schemes in [13] (note that Corollary 2.1, to be presented in the next subsection, also contains

a capacity achieving SLDC). The converse follows from Theorem 1 as SLDCs are special cases of ULDCs.

As another corollary, the capacity of $\text{RIR}_{\max}$ is shown to be the same as the capacity of $\text{PIR}_{\max}$.

Corollary 1.2.

$$C_{\text{RIR}_{\max}}(N,K) = \left(1 + 1/N + 1/N^2 + \cdots + 1/N^{K-1}\right)^{-1} = C_{\text{PIR}_{\max}}(N,K) = C_{\text{PIR}_{ave}}(N,K). \quad (19)$$

The achievability for Corollary 1.2 follows because $\text{PIR}_{\max}$ schemes are special cases of $\text{RIR}_{\max}$ schemes and capacity achieving $\text{PIR}_{\max}$ schemes are available from [13]. The converse follows from Observation 1 and Theorem 1. That is, the rate of any $\text{RIR}_{\max}$ scheme must be no higher than $C_{\text{RIR}_{\max}}(N,K)$, otherwise by Observation 1 we will have a ULDC that has a rate higher than $C^*_{\text{ULDC}}(N,K)$, contradicting Theorem 1.

B. Optimal Code Length and Upload Cost Results

The next set of results concerns minimum code lengths and minimum upload costs. We first show that given N, K, the minimum code length M of ULDCs for which the capacity takes its maximum value (maximum over all M), is N^K.

Theorem 2.

$$\min\{M \mid C_{ULDC}(N,K,M) = C^*_{ULDC}(N,K)\} = N^K. \quad (20)$$

For the converse, we prove that any capacity achieving ULDCs must have length $M \geq N^K$. The proof is presented in Section V. Since SLDCs are special cases of ULDCs, the converse also applies to SLDCs. For the achievability, we provide a construction of a capacity achieving SLDC with length $M = N^K$. The proof is presented in Section VI. Since every SLDC is also a ULDC, the achievability applies also to ULDCs. Thus, we immediately have the following corollary for SLDCs.

Corollary 2.1.

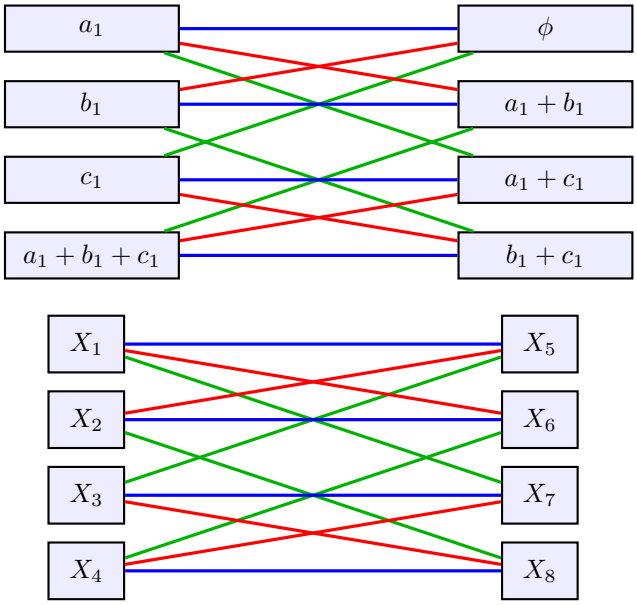
$$\min\{M \mid C_{SLDC}(N,K,M) = C^*_{SLDC}(N,K)\} = N^K. \quad (21)$$

Corollary 2.2. *The minimum upload cost of a capacity achieving $\text{RIR}_{\max}$ scheme with K messages and N databases is $(K-1)\log(N)$ per database.*

Corollary 2.3. *The minimum upload cost of a capacity achieving $\text{PIR}_{\max}$ scheme with K messages and N databases is $(K-1)\log(N)$ per database.*

The proofs of Corollaries 2.2 and 2.3 are presented in Section VII.

It is already known from [13] that the capacity of $\text{PIR}_{\max}$ is the same as the capacity of PIR_{ave}. Surprisingly, based on Corollary 2.3 and the results in [14], it turns out that the minimum upload cost for $\text{PIR}_{\max}$ is also the same as the minimum upload cost of PIR_{ave}. Note that any capacity achieving, upload optimal $\text{PIR}_{\max}$ scheme is also a capacity achieving, upload optimal PIR_{ave} scheme. However, the reverse direction is not true. This is evident from Figure 3 which shows capacity achieving and upload optimal schemes for both settings.



$$X_1 = (a_1, b_1, c_1, a_2+b_2, a_3+c_2, b_3+c_3, a_4+b_4+c_4)$$
$$X_2 = (a_6, b_6, c_4, a_5+b_5, a_8+c_3, b_8+c_2, a_7+b_7+c_1)$$
$$X_3 = (a_7, b_4, c_6, a_8+b_3, a_5+c_5, b_2+c_8, a_6+b_1+c_7)$$
$$X_4 = (a_4, b_7, c_7, a_3+b_8, a_2+c_8, b_5+c_5, a_1+b_6+c_6)$$
$$X_5 = (a_5, b_2, c_2, a_6+b_1, a_7+c_1, b_4+c_4, a_8+b_3+c_3)$$
$$X_6 = (a_2, b_5, c_3, a_1+b_6, a_4+c_4, b_7+c_1, a_3+b_8+c_2)$$
$$X_7 = (a_3, b_3, c_5, a_4+b_4, a_1+c_6, b_1+c_7, a_2+b_2+c_8)$$
$$X_8 = (a_8, b_8, c_8, a_7+b_7, a_6+c_7, b_6+c_6, a_5+b_5+c_5)$$

Fig. 3. *Shown at the top is a capacity achieving, upload optimal PIR_{ave} scheme for $K = 3$ messages, $N = 2$ databases from [14]. At the bottom is the corresponding capacity achieving, upload optimal $\text{PIR}_{\max}$ scheme from this work. The messages are denoted by $W_1 = a_{[1:L_w]}, W_2 = b_{[1:L_w]}, W_3 = c_{[1:L_w]}$, in both cases, with $L_w = 1$ for PIR_{ave} and $L_w = 8$ for $\text{PIR}_{\max}$. Nodes in the left column are all possible answers from Database 1, and the nodes in the right column are all possible answers from Database 2. In both cases, W_1 can be retrieved from pairs of nodes connected by blue edges, W_2 from red edges and W_3 from green edges.*

The PIR_{ave} scheme shown in Figure 3 uses message size $L_w = 1$ bit and achieves an average download of L_w from Database 1, and $\frac{3}{4}L_w = 3/4$ from Database 2, for total average download of $\frac{7}{4}L_w$, so its rate is $4/7$, the capacity for this setting. Note that this is because with probability $1/4$ nothing is downloaded from Database 2. However, the maximum download for this scheme is L_w per database which is not optimal. Therefore, using the answers from this scheme directly to produce an LDC would result in an LDC with $L_x = L_w$, which is not capacity achieving. On the other hand, the $\text{PIR}_{\max}$ scheme shown in Figure 3 uses message size $L_w = 8$ bits, and achieves constant, maximum, and average download of $\frac{7}{8}L_w = 7$ bits from each database, for a total download of $\frac{7}{4}L_w$, so its rate is also $4/7$, same as the capacity for this setting. This is a stronger capacity achieving scheme because not only is it capacity achieving and upload optimal for $\text{PIR}_{\max}$ but also it is capacity achieving and upload optimal for PIR_{ave}. Furthermore, the same scheme gives us a minimum length capacity achieving ULDC, a minimum length

capacity achieving SLDC, as well as a capacity achieving and upload optimal scheme for $\text{RIR}_{\max}$. Note that the upload optimal $\text{PIR}_{\max}$ scheme cannot be obtained simply from a time-sharing argument that symmetrizes the upload optimal PIR_{ave} scheme, because the time-sharing argument increases the upload cost. Instead, this powerful scheme, which gets even more sophisticated for larger number of messages and databases, is obtained by a special construction specified in Section VI.

IV. Converse Proof of Theorem 1

Let us start with a simple yet extremely useful lemma.

Lemma 1. *Let $S \in \mathcal{S}_k$ be an arbitrary decoding set of W_k. Consider an arbitrary subset of $[1:K]$, denoted by $\mathcal{J}$, such that $k \notin \mathcal{J}$. Then for any element X_s in S, we have*

$$\sum_{X_i \in S} H(X_i|W_{\mathcal{J}}) \geq L_w + H(X_s|W_{\{k\}\cup\mathcal{J}}), \quad \forall X_s \in S. \tag{22}$$

Proof:

$$\begin{aligned}
\sum_{X_i\in S} H(X_i|W_{\mathcal{J}}) &\geq H(S|W_{\mathcal{J}}) && (23)\\
&\overset{(a)}{=} H(S, W_k|W_{\mathcal{J}}) && (24)\\
&\overset{(2)}{=} H(W_k) + H(S|W_k, W_{\mathcal{J}}) && (25)\\
&\overset{(3)}{\geq} L_w + H(X_s|W_{\{k\}\cup\mathcal{J}}) && (26)
\end{aligned}$$

where (a) follows from the fact that S is a decoding set of W_k, so from S, we may decode W_k. The last step is due to the assumption that $X_s \in S$. ∎

Remark: Lemma 1 states that the amount of information contained in any decoding set of a source symbol is no less than the entropy of that source symbol plus the entropy of any coded symbol from the decoding set conditioned on that source symbol (i.e., interference about other source symbols).

The rest of the proof follows from invoking Lemma 1 for a carefully chosen sequence of decoding sets and a permutation of the K source symbols. Consider an arbitrary permutation of $[1:K]$, π such that $(1, 2, \cdots, K)$ is mapped to $(\pi_1, \pi_2, \cdots, \pi_K)$.

The decoding sets and coded symbols involved in the converse proof are constructed following a full N-ary tree with depth K (see Figure 4). At depth-k, $k \in [1:K]$, there are N^{k-1} decoding sets (not necessarily distinct) of the source symbol W_{π_k}. Specifically, we start from the root, where we pick an arbitrary coded symbol, X_{i_1}. Because the LDC is universal, X_{i_1} can be used to decode W_{π_1}, with another $N-1$ symbols (denoted as $X_{i_2}, \cdots, X_{i_N}$). These N symbols form the depth-1 nodes and this decoding set is denoted as $S^{[1]}_{\pi_1}$. The remaining procedure is similar, where for each node at depth-$(k-1)$, we find a decoding set of the source symbol W_{π_k} that contains it and these decoding sets appear at depth-k. Finally, at depth-K, we have N^{K-1} decoding sets of the source symbol W_{π_K}. When referring to a node in the full N-ary tree, we may use either the content (i.e., the entropy term) or the X_i value (called the node *label*).

Example 1. *To illustrate the construction of the full N-ary tree, we consider an example of a ULDC as shown in Figure 5. For one possible construction of the full binary tree, we set the permutation π as the identity permutation and pick X_1 as the root node. To find the depth-1 nodes, we pick any decoding set of W_1 that contains X_1, say $\{X_1, X_2\} \triangleq S^{[1]}_1$, so that the depth-1 nodes are $H(X_1|W_2, W_3)$ and $H(X_2|W_2, W_3)$. Next, we find the depth-2 nodes. Consider the two depth-1 nodes and for each of them, we pick any decoding set of W_2 that contains the coded symbol in the depth-1 node. For the first depth-1 node $H(X_1|W_2, W_3)$, we only have 1 decoding set that contains X_1 (note that there must exist one as the LDC is universal), so $S^{[1]}_2 = \{X_1, X_2\}$. For the second depth-1 node $H(X_2|W_2, W_3)$, we have 2 decoding sets that contain X_2 and we may choose either one, say we choose $\{X_2, X_3\} \triangleq S^{[2]}_2$. We have now found the 4 depth-2 nodes, as $H(X_1|W_3), H(X_2|W_3), H(X_2|W_3)$, and $H(X_3|W_3)$, where the first two nodes are from $S^{[1]}_2$ and the last two nodes are from $S^{[2]}_2$. Note that the nodes at the same depth are not necessarily distinct, e.g., X_2 appears twice[6] at depth-2. Finally, we consider the depth-K (depth-3) nodes. For each one of the depth-2 nodes, we find a decoding set of W_3 that contains it, e.g., $S^{[1]}_3 = \{X_1, X_3\}, S^{[2]}_3 = \{X_2, X_3\}, S^{[3]}_3 = \{X_2, X_4\}, S^{[4]}_3 = \{X_3, X_2\}$, then the depth-3 nodes are $H(X_1), H(X_3), H(X_2), H(X_3), H(X_2), H(X_4), H(X_3), H(X_2)$, where sequentially every 2 nodes form a decoding set of W_3. The construction of the full binary tree is now complete.*

Remark: From this example, it is clear that there are many different ways to generate the full N-ary tree (e.g., the permutation can be chosen arbitrarily, the root node can be chosen arbitrarily, and when there are multiple qualified decoding sets, any one may be chosen). Interestingly, the converse proof works for any realization of the full N-ary tree.

For the converse proof, we start from the N^{K-1} decoding sets of the source symbol W_{π_K} at depth-K and repeatedly apply Lemma 1 as we ascend the tree, and stop when we reach the root.

$$\begin{aligned}
N^K L_x &= \sum_{n=1}^{N^{K-1}} \sum_{X_i \in S^{[n]}_{\pi_K}} H(X_i) && (27)\\
&\overset{(22)}{\geq} N^{K-1} L_w + \sum_{n=1}^{N^{K-2}} \sum_{X_i \in S^{[n]}_{\pi_{K-1}}} H(X_i|W_{\pi_K}) && (28)\\
&\overset{(22)}{\geq} N^{K-1} L_w + N^{K-2} L_w \\
&\quad + \sum_{n=1}^{N^{K-3}} \sum_{X_i \in S^{[n]}_{\pi_{K-2}}} H(X_i|W_{\pi_{K-1:K}}) && (29)\\
&\geq \cdots && (30)
\end{aligned}$$

[6]However, for any ULDC to achieve the capacity, the nodes from the same depth must be distinct. We refer to the proof of Theorem 2 for the justification of this distinctness property. Therefore, it follows that this ULDC does not achieve the capacity, verified by noting that the symbol rate is $R = L_w/L_x = 1$ while the capacity is $C^*_{\text{ULDC}}(N = 2, K = 2) = 4/3$.

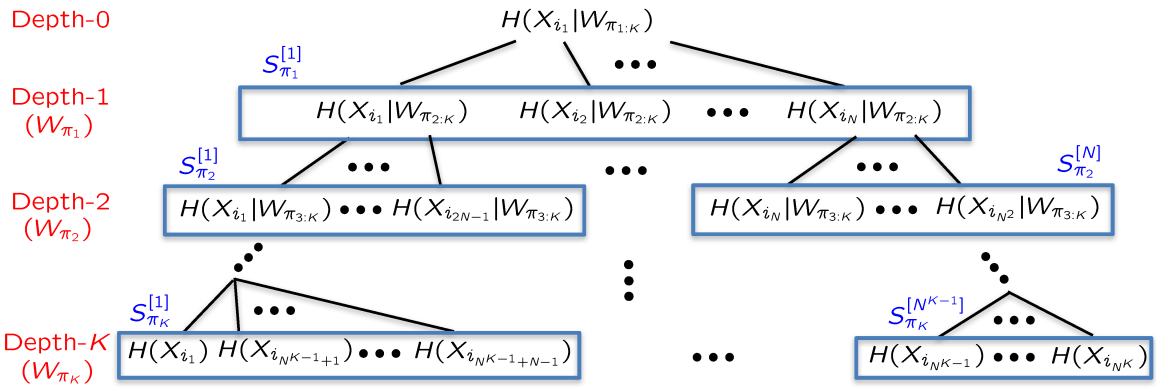


Fig. 4. The full N-ary tree with depth K containing all coded symbols and decoding sets that appear in the converse proof. The indices of coded symbols are labelled lexicographically from the root to the leaf nodes (they are not necessarily distinct).

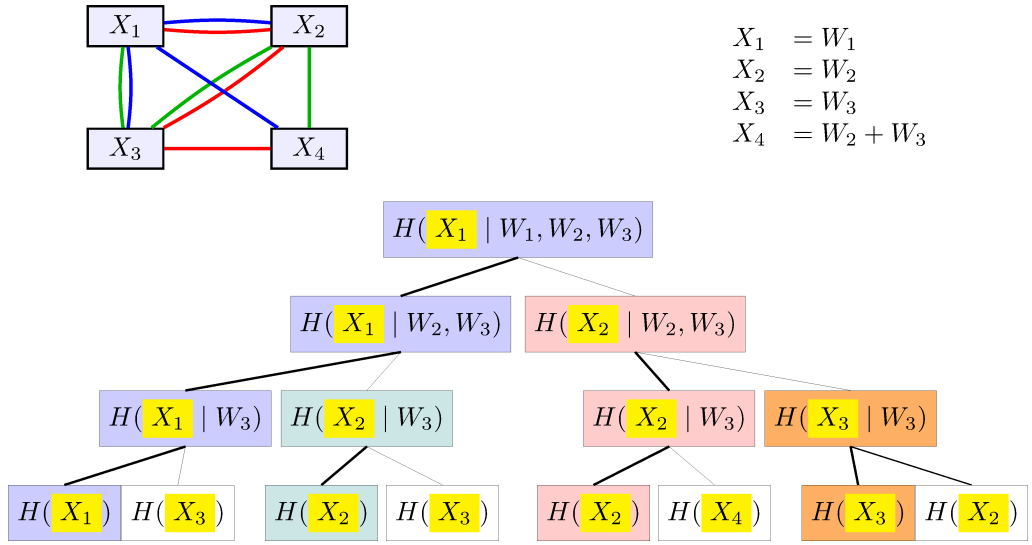


Fig. 5. *Shown at the top of the figure is a ULDC with locality $N = 2$ that codes $K = 3$ binary source symbols, W_1, W_2, W_3, into $M = 4$ binary coded symbols, X_1, X_2, X_3, X_4. The decoding sets for W_1, W_2, W_3 are shown as pairs of coded symbols connected by blue, red, and green edges, respectively. At the bottom of the figure is one possible N-ary tree for this ULDC. Node labels are the X_i values highlighted in yellow.*

$$\begin{aligned}
&\overset{(22)}{\geq} && N^{K-1}L_w + \cdots + NL_w \\
&&&+ \sum_{X_i \in S_{\pi_1}^{[1]}} H(X_i|W_{\pi_{2:K}}) && (31)\\
&\overset{(22)}{\geq} && N^{K-1}L_w + \cdots + NL_w + L_w \\
&&&+ H(X_{i_1}|W_{\pi_{1:K}}) && (32)\\
&\geq && (N^{K-1} + \cdots + N + 1)L_w && (33)
\end{aligned}$$

We obtain the final rate bound by rearranging terms.

$$R_s = \frac{L_w}{L_x} \leq N\left(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}}\right)^{-1}. \quad (34)$$

■

V. Proof of Theorem 2: Converse

We show that a capacity achieving ULDC has length at least N^K. For a set $\mathcal{K} \subset [1:K]$, denote its complement set as $\overline{\mathcal{K}}$ (i.e., the set of elements that are not in $\mathcal{K}$). We start by defining when two coded symbols contain the same information about a source symbol set.

Definition 13 (Same information)**.** *We say that two coded symbols X_{i_1}, X_{i_2} contain the same information about a set of source symbols $W_{\mathcal{K}}$ if $H(X_{i_1}|X_{i_2}, W_{\overline{\mathcal{K}}}) = H(X_{i_2}|X_{i_1}, W_{\overline{\mathcal{K}}}) = 0$ and denote it as $X_{i_1} \overset{W_{\mathcal{K}}}{\simeq} X_{i_2}$.*

By definition, the same information operation is symmetric, i.e., if $X_{i_1} \overset{W_{\mathcal{K}}}{\simeq} X_{i_2}$, then $X_{i_2} \overset{W_{\mathcal{K}}}{\simeq} X_{i_1}$. Interestingly, the same information operation is also transitive. This is proved in the following lemma.

Lemma 2 (Transitivity of same information)**.** *If $X_{i_1} \overset{W_{\mathcal{K}}}{\simeq} X_{i_2}$ and $X_{i_2} \overset{W_{\mathcal{K}}}{\simeq} X_{i_3}$, then $X_{i_1} \overset{W_{\mathcal{K}}}{\simeq} X_{i_3}$.*

Proof: We show that $H(X_{i_1}|X_{i_3}, W_{\overline{\mathcal{K}}}) = 0$, and the proof

of $H(X_{i_3}|X_{i_1}, W_{\overline{\mathcal{K}}}) = 0$ follows by symmetry.

$$\begin{aligned} & H(X_{i_1}|X_{i_3}, W_{\overline{\mathcal{K}}}) \\ = \; & H(X_{i_1}|X_{i_2}, X_{i_3}, W_{\overline{\mathcal{K}}}) + I(X_{i_1}; X_{i_2}|X_{i_3}, W_{\overline{\mathcal{K}}}) \quad (35) \\ = \; & H(X_{i_1}|X_{i_2}, X_{i_3}, W_{\overline{\mathcal{K}}}) + H(X_{i_2}|X_{i_3}, W_{\overline{\mathcal{K}}}) \\ & - H(X_{i_2}|X_{i_1}, X_{i_3}, W_{\overline{\mathcal{K}}}) \quad (36) \\ = \; & 0 \quad (37) \end{aligned}$$

where in (36), the first term is zero because $X_{i_1} \overset{W_{\mathcal{K}}}{\simeq} X_{i_2}$ (i.e., $H(X_{i_1}|X_{i_2}, W_{\overline{\mathcal{K}}}) = 0$) and adding conditioning can not increase entropy and the last two terms are zero because $X_{i_2} \overset{W_{\mathcal{K}}}{\simeq} X_{i_3}$. ■

Similarly, we define when two coded symbols contain distinct information about a single source symbol.

Definition 14 (Distinct information)**.** *We say that two coded symbols X_{i_1}, X_{i_2} contain distinct information about the source symbol $W_k, k \in [1:K]$ if $H(X_{i_1}|X_{i_2}, W_{\overline{k}}) = H(X_{i_1}|W_{\overline{k}})$ and denote it as $X_{i_1} \overset{W_k}{\perp} X_{i_2}$.*

Next we distill properties of capacity achieving ULDCs.

Lemma 3 (Properties of capacity achieving ULDC)**.** *For capacity achieving ULDCs, we have*

1) *(Non-zero entropy property) $\forall i \in [1:M]$, $\forall k \in [1:K]$, $H(X_i|W_{\overline{k}}) \neq 0$.*
2) *For an arbitrary decoding set of $W_k, k \in [1:K]$, $S \in \mathcal{S}_k$,*
   a) *(Same interference property) $\forall i_1, i_2 \in S, \forall k' \neq k$, $X_{i_1} \overset{W_{k'}}{\simeq} X_{i_2}$.*
   b) *(Distinct desired information property) $\forall i_1, i_2 \in S$, $X_{i_1} \overset{W_k}{\perp} X_{i_2}$.*
   c) *(Independence of coded symbols) $\forall i_1, i_2 \in S$, $H(X_{i_1}|X_{i_2}) = H(X_{i_1})$.*
3) *(Incompatibility of same and distinct information) There do not exist coded symbols X_{i_1}, X_{i_2} and source symbol W_k such that $X_{i_1} \overset{W_k}{\simeq} X_{i_2}$ and $X_{i_1} \overset{W_k}{\perp} X_{i_2}$.*

The proof of Lemma 3 is deferred to Section V-C.

Remark: The idea of using properties on same interference and distinct information has appeared previously in [14], albeit within a restricted class of decomposable (e.g., linear) schemes. Here we develop them in the information theoretic sense (that works for any non-linear schemes). Further we treat same and distinct information as general mathematical operators and establish the transitivity of same information and incompatibility of same and distinct information.

Equipped with the definitions and lemmas presented above, we are now ready for the proof, i.e., any capacity achieving ULDC must have length $M \geq N^K$. The proof idea is to consider a full N-ary tree (refer to Figure 4) that contains N^K coded symbols and show that these coded symbols must be all distinct (so the length $M \geq N^K$). To this end, we show that if any two coded symbols are the same, then the ULDC can not achieve the capacity (as some properties established in Lemma 3 are violated). To illustrate the idea in a simpler setting, let us start from an example with $N = 2, K = 3$.

A. Example: $N = 2, K = 3$

We redraw the full binary tree with depth 3 in Figure 6, when the permutation is the identity permutation. There are $N^K = 8$ coded symbols (leaf nodes) involved, i.e., $X_{i_1}, \cdots, X_{i_8}$, and we show that they are all distinct, i.e., $X_{i_j} \neq X_{i_l}, \forall j, l \in [1:8], j \neq l$. This is proved by contradiction, i.e., if $X_{i_j} = X_{i_l}$, then the ULDC violates some property that must be satisfied by capacity achieving ULDCs.

We have 3 cases for the 2 leaf nodes X_{i_j}, X_{i_l}.

1) X_{i_j}, X_{i_l} are siblings (i.e., X_{i_j}, X_{i_l} have the same parent). For example, X_{i_1} and X_{i_5} are siblings. Now if $X_{i_1} = X_{i_5}$, we have $H(X_{i_1}|X_{i_5}) = 0$. Noting that X_{i_1}, X_{i_5} form a decoding set of W_3, we apply the independence property of coded symbols (Property 2.(c)), and obtain $H(X_{i_1}) = H(X_{i_1}|X_{i_5}) = 0$, which contradicts the fact that $H(X_{i_1}) = L_x \neq 0$ (as the code is capacity achieving). Therefore X_{i_1}, X_{i_5} must be distinct.
2) X_{i_j}, X_{i_l} are descendants of the same node from depth-1 (i.e., the same depth-1 node is reached from X_{i_j}, X_{i_l} by proceeding from child to parent). For example, the leaf nodes X_{i_5} and X_{i_6} are descendants of the same depth-1 node with label X_{i_1}. As $\{X_{i_1}, X_{i_5}\}$ can be used to decode W_3, we apply the same interference property to obtain that X_{i_1}, X_{i_5} contain the same information about W_2, i.e.,
$$\{X_{i_1}, X_{i_5}\} \in \mathcal{S}_3 \overset{\text{Property 2.(a)}}{\Longrightarrow} X_{i_1} \overset{W_2}{\simeq} X_{i_5}. \quad (38)$$
Similarly, $\{X_{i_3}, X_{i_6}\}$ can be used to decode W_3 so that they contain the same information about W_2,
$$\{X_{i_6}, X_{i_3}\} \in \mathcal{S}_3 \overset{\text{Property 2.(a)}}{\Longrightarrow} X_{i_6} \overset{W_2}{\simeq} X_{i_3}. \quad (39)$$
Now suppose $X_{i_5} = X_{i_6}$. Applying the transitivity of the same information operation, we have that X_{i_1}, X_{i_3} must contain the same information about W_2.
$$X_{i_1} \overset{W_2}{\simeq} X_{i_5}, X_{i_5} \overset{W_2}{\simeq} X_{i_3} \overset{\text{Lemma 2}}{\Longrightarrow} X_{i_1} \overset{W_2}{\simeq} X_{i_3}. \quad (40)$$
However, $\{X_{i_1}, X_{i_3}\}$ can be used to decode W_2, so from the distinct desired information property (Property 2.(b)), they must contain distinct information about W_2.
$$\{X_{i_1}, X_{i_3}\} \in \mathcal{S}_2 \overset{\text{Property 2.(b)}}{\Longrightarrow} X_{i_1} \overset{W_2}{\perp} X_{i_3}. \quad (41)$$
Finally, we arrive at the contradiction by invoking the incompatibility property of same and distinct information (Property 3).
$$X_{i_1} \overset{W_2}{\simeq} X_{i_3}, X_{i_1} \overset{W_2}{\perp} X_{i_3} \overset{\text{Property 3}}{\Longrightarrow} \text{Contradiction}. \quad (42)$$
Therefore we conclude that X_{i_5} and X_{i_6} must be distinct. The proof for other choices of X_{i_j}, X_{i_l} is similar.
3) X_{i_j}, X_{i_l} are descendants of the same node from depth-0. For example, the leaf nodes X_{i_6} and X_{i_8} are descendants of the same depth-0 node with label X_{i_1}. The remaining proof is similar to the one above, where we trace X_{i_6} to X_{i_1} (and X_{i_8} to X_{i_2}) using decoding constraints of W_2, W_3 and argue that they must contain the same information about W_1. Then if $X_{i_6} = X_{i_8}$, X_{i_1} and X_{i_2} must

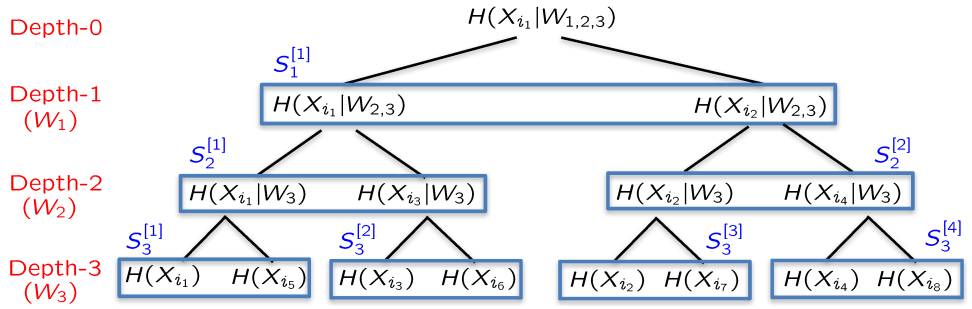


Fig. 6. The full binary tree with locality $N = 2$ and $K = 3$ messages.

contain the same information about W_1, contradicting the fact that they must contain distinct information about W_1 (as X_{i_1} and X_{i_2} form a decoding set of W_1).

$$\begin{aligned}
&\{X_{i_6}, X_{i_3}\} \in \mathcal{S}_3 && \xRightarrow{\text{Property 2.(a)}} && X_{i_6} \overset{W_1}{\simeq} X_{i_3} && (43)\\
&\{X_{i_3}, X_{i_1}\} \in \mathcal{S}_2 && \xRightarrow{\text{Property 2.(a)}} && X_{i_3} \overset{W_1}{\simeq} X_{i_1} && (44)\\
& && \xRightarrow{\text{Lemma 2}} && X_{i_6} \overset{W_1}{\simeq} X_{i_1} && (45)\\
& && \xRightarrow{\text{(Symmetrically)}} && X_{i_8} \overset{W_1}{\simeq} X_{i_3} && (46)\\
&\text{Suppose } X_{i_6} = X_{i_8} && \xRightarrow{\text{Lemma 2}} && X_{i_1} \overset{W_1}{\simeq} X_{i_3} && (47)\\
&\{X_{i_1}, X_{i_3}\} \in \mathcal{S}_1 && \xRightarrow{\text{Property 2.(b)}} && X_{i_1} \overset{W_1}{\perp} X_{i_3} && (48)\\
& && \xRightarrow{\text{Property 3}} && \text{Contradiction} && (49)
\end{aligned}$$

The proof for other choices of X_{i_j}, X_{i_l} is similar.

The proof for the 3 cases is now complete. To sum up, any two coded symbols can not be the same, i.e., all $N^K = 8$ coded symbols are all distinct, so the code length for any capacity achieving ULDC must satisfy $M \geq N^K = 8$. The converse proof with $N = 2, K = 3$ is thus complete.

B. General Proof for Arbitrary N, K

The general proof for arbitrary N, K is a simple generalization of that presented in the previous section. Consider a full N-ary tree with depth K (refer to Figure 4), root node label X_{i_1} and permutation π. There are N^K coded symbols that appear as the leaf nodes. We show that they are all distinct.

To set up the proof by contradiction, let us assume there exist two coded symbols $X_j, X_{j'}$ such that $X_j = X_{j'}$. We have two cases.

1) $X_j, X_{j'}$ are siblings. In this case if $X_j = X_{j'}$, then $H(X_j|X_{j'}) = 0$. However, as $X_j, X_{j'}$ are siblings, they belong to a decoding set of W_{π_K}. Applying the independence property of coded symbols (Property 2.(c)), we have $H(X_j) = H(X_j|X_{j'}) = 0$, which contradicts the fact that $H(X_j) = L_x \neq 0$ (as the code is capacity achieving). Therefore $X_j, X_{j'}$ must be distinct.
2) $X_j, X_{j'}$ are descendants of the same node (denoted as X_{j*}) from depth-$k, k \in [0 : K-2]$. We find the path from X_j to X_{j*} (by moving from chid to parent recursively). The path passes $K-k+1$ nodes (one each from depth-$k', k' \in [k : K]$).

$$X_j - X_{j_1} - X_{j_2} \cdots - X_{j_l} - X_{\tilde{j}} - \cdots - X_{\tilde{j}} - X_{j*}. \quad (50)$$

Note that due to the construction of the full N-ary tree, the coded symbol in the parent node is always equal to the coded symbol in the leftmost child node. The nodes that appear in the path are initially distinct but after some steps, the node (nodes) that appear in the path will be equal to $X_{\tilde{j}}$ (which might be the same as X_{j*} if $X_{\tilde{j}}$ is the leftmost child of X_{j*}). Any two distinct adjacent nodes in the path belong to a decoding set of some source symbol $W_{k'}, k' \in [k+2 : K]$. Applying the same interference property to each such pair of nodes, we have

$$\begin{aligned}
&X_j \overset{W_{k+1}}{\simeq} X_{j_1}, X_{j_1} \overset{W_{k+1}}{\simeq} X_{j_2}, \cdots, X_{j_l} \overset{W_{k+1}}{\simeq} X_{\tilde{j}}\\
&\xRightarrow{\text{Lemma 2}} X_j \overset{W_{k+1}}{\simeq} X_{\tilde{j}} \quad (51)
\end{aligned}$$

Symmetrically, we consider the path from $X_{j'}$ to X_{j*},

$$X_{j'} - X_{j'_1} - X_{j'_2} \cdots - X_{j'_{l'}} - X_{\tilde{j'}} - \cdots - X_{\tilde{j'}} - X_{j*}. \quad (52)$$

Similarly, we apply the same interference property to distinct adjacent nodes in the path as they belong to a decoding set of some source symbol other than W_{k+1}.

$$\begin{aligned}
&X_{j'} \overset{W_{k+1}}{\simeq} X_{j'_1}, X_{j'_1} \overset{W_{k+1}}{\simeq} X_{j'_2}, \cdots, X_{j'_{l'}} \overset{W_{k+1}}{\simeq} X_{\tilde{j'}}\\
&\xRightarrow{\text{Lemma 2}} X_{j'} \overset{W_{k+1}}{\simeq} X_{\tilde{j'}} \quad (53)
\end{aligned}$$

Now if $X_j = X_{j'}$, then

$$X_j \overset{W_{k+1}}{\simeq} X_{\tilde{j}}, X_j \overset{W_{k+1}}{\simeq} X_{\tilde{j'}} \xRightarrow{\text{Lemma 2}} X_{\tilde{j}} \overset{W_{k+1}}{\simeq} X_{\tilde{j'}} \quad (54)$$

However, this contradicts the fact that $X_{\tilde{j}}, X_{\tilde{j'}}$ belong to a decoding set of the source symbol W_{k+1} (as the two paths overlap at node X_{j*}).

$$X_{\tilde{j}} \overset{W_{k+1}}{\perp} X_{\tilde{j'}}, X_{\tilde{j}} \overset{W_{k+1}}{\simeq} X_{\tilde{j'}} \xRightarrow{\text{Property 3}} \text{Contradiction.} \quad (55)$$

Therefore, $X_j = X_{j*}$ can not hold and we have N^K distinct coded symbols, i.e., $M \geq N^K$. The proof is thus complete. ■

Remark: Comparing our minimum length proof of capacity achieving ULDC (and the upload cost proof of $\text{PIR}_{\max}$) to the upload cost proof of PIR_{ave} [14], we have an additional non-zero entropy property (Property 1 in Lemma 3) that allows the proof to work for all non-linear schemes (whereas the result of [14] is limited to a restricted class of decomposable schemes).

C. Proof of Lemma 3

Let us prove the properties listed in Lemma 3 one at a time.

1) Proof of Property 1: To set up the proof by contradiction, let us assume, for some $i_1 \in [1:M], k \in [1:K]$,

$$H(X_{i_1}|W_{\overline{k}}) = 0. \tag{56}$$

Consider a full N-ary tree (see Figure 4) with root node label X_{i_1} and permutation π such that $\pi_1 = k$. Thus $W_{\pi_{2:K}} = W_{\overline{k}}$. For a capacity achieving ULDC, all the inequalities from (28) to (33) must be equalities. Replacing (32) and (33) with equalities, we have

$$\begin{aligned} L_w &= \sum_{X_i \in S_{\pi_1}^{[1]}} H(X_i|W_{\overline{k}}) && (57)\\ &= H(X_{i_1}|W_{\overline{k}}) + H(X_{i_2}|W_{\overline{k}}) + \cdots \\ &\quad + H(X_{i_{N-1}}|W_{\overline{k}}) + H(X_{i_N}|W_{\overline{k}}) && (58)\\ &= H(X_{i_2}|W_{\overline{k}}) + \cdots + H(X_{i_N}|W_{\overline{k}}) && (59)\end{aligned}$$

where in (59), we used our assumption (56). Because the sum of $N-1$ non-negative terms is equal to L_w, we must have at least one term, say corresponding to X_{i*}, that is not less than $\frac{L_w}{N-1}$.

$$H(X_{i*}|W_{\overline{k}}) \geq \frac{L_w}{N-1}. \tag{60}$$

Because the code is universal, there exists a decoding set $S \in \mathcal{S}_j$ of message $W_j, j \neq k$ that contains X_{i*}.

$$\begin{aligned} NL_x &= \sum_{X_i \in S} H(X_i) && (61)\\ &\overset{(22)}{\geq} L_w + H(X_{i^*}|W_j) && (62)\\ &\geq L_w + H(X_{i^*}|W_{\overline{k}}) && (63)\end{aligned}$$

Plugging in the capacity achieving condition, $L_x = C^*_{\text{ULDC}}(N,K)^{-1}L_w$, we have

$$\begin{aligned} H(X_{i^*}|W_{\overline{k}}) &\leq L_w(NC^*_{\text{ULDC}}(N,K)^{-1} - 1) && (64)\\ &= \left(\frac{1}{N} + \frac{1}{N^2} + \cdots + \frac{1}{N^{K-1}}\right) L_w && (65)\\ &< \frac{1/N}{1-1/N} L_w = \frac{L_w}{N-1} && (66)\end{aligned}$$

But (60) and (66) contradict each other. The contradiction completes the proof of Property 1. ∎

2) Proof of Property 2: First let us prove (a), that $\forall X_{i_1}, X_{i_2} \in S \in \mathcal{S}_k$ and $\forall k' \neq k$, $X_{i_1} \overset{W_{k'}}{\simeq} X_{i_2}$. For this purpose, let us consider a full N-ary tree (see Figure 4) where the root has label X_{i_1}, the permutation π satisfies $\pi_K = k$, and X_{i_1}, X_{i_2} appear at depth-K in decoding set S. Consider the step from depth-K to depth-$(K-1)$ of the converse proof (i.e., (28)). As we assume the ULDC achieves the capacity, the following equality must hold (refer to (26)).

$$\begin{aligned} \sum_{X_i \in S_{\pi_K}} H(X_i) &= \sum_{X_i \in S} H(X_i) && (67)\\ &= L_w + H(X_{i_1} \mid W_k) && (68)\\ &= L_w + H(S \mid W_k) && (69)\end{aligned}$$

In (69) we used (26), which must also be an equality for a capacity achieving ULDC. From (68) and (69) we must have

$$\begin{aligned} & H(X_{i_1}, X_{i_2}|W_k) = H(X_{i_1}|W_k) && (70)\\ \Rightarrow\quad & H(X_{i_2}|X_{i_1}, W_k) = 0 && (71)\\ \Rightarrow\quad & H(X_{i_2}|X_{i_1}, W_{\overline{k'}}) = 0, k' \neq k. && (72)\end{aligned}$$

By symmetry, we can similarly prove $H(X_{i_1}|X_{i_2}, W_{\overline{k'}}) = 0$ so that $X_{i_1} \overset{W_{k'}}{\simeq} X_{i_2}$ and we have proved Property $2(a)$.

To prove Property $2(b)$, we consider a full N-ary tree (see Figure 4) where the root has label X_{i_1}, the permutation π satisfies $\pi_1 = k$ (such that $\pi_{2:K} = \overline{k}$), and the label X_{i_2} appears at depth-1. Consider the step from depth-1 to depth-0 of the converse proof (i.e., (32)). As the ULDC achieves the capacity, the following equality must hold (refer to (23)).

$$\begin{aligned} & H(X_{i_1}|W_{\overline{k}}) + H(X_{i_2}|W_{\overline{k}}) = H(X_{i_1}, X_{i_2}|W_{\overline{k}}) && (73)\\ \Rightarrow\quad & H(X_{i_1}|X_{i_2}, W_{\overline{k}}) = H(X_{i_1}|W_{\overline{k}}) && (74)\end{aligned}$$

Therefore we have proved Property $2(b)$, that $X_{i_1} \overset{W_k}{\perp} X_{i_2}$ holds.

To prove Property $2(c)$, we consider a full N-ary tree (see Figure 4) where the root label is X_{i_1}, the permutation π satisfies $\pi_K = k$, and the label X_{i_2} appears at depth-K. Consider the step from depth-K to depth-$(K-1)$ of the converse proof (i.e., (28)). As we assume the ULDC achieves the capacity, the following equality must hold (refer to (23)).

$$\begin{aligned} & H(X_{i_1}) + H(X_{i_2}) = H(X_{i_1}, X_{i_2}) && (75)\\ \Rightarrow\quad & H(X_{i_1}|X_{i_2}) = H(X_{i_1}) && (76)\end{aligned}$$

Therefore the desired claim is proved. ∎

3) Proof of Property 3:

$$\begin{aligned} & X_{i_1} \overset{W_k}{\simeq} X_{i_2} \;\Rightarrow\; H(X_{i_1}|X_{i_2}, W_{\overline{k}}) = 0 && (77)\\ & X_{i_1} \overset{W_k}{\perp} X_{i_2} \;\Rightarrow\; H(X_{i_1}|X_{i_2}, W_{\overline{k}}) = H(X_{i_1}|W_{\overline{k}}) && (78)\\ \Rightarrow\quad & H(X_{i_1}|W_{\overline{k}}) = 0 && (79)\end{aligned}$$

which contradicts the non-zero entropy property (Property 1). So same and distinct information conditions can not be simultaneously satisfied and the proof is complete. ∎

VI. Proof of Theorem 2: Achievability

In this section, we present the construction of a capacity achieving SLDC with length $M = N^K$. Before proceeding to the general proof, we first consider two examples.

A. Example 1: $N = 2, K = 2$

When $N = 2, K = 2$, the capacity is $C^*_{\text{ULDC}}(N=2, K=2) = \frac{L_w}{L_x} = 2(1+\frac{1}{2})^{-1} = \frac{4}{3}$. We present an SLDC with length 4, where each source symbol is comprised of $L_w = 4$ bits and each coded symbol has $L_x = 3$ bits.

Denote $W_1 = (a_1, a_2, a_3, a_4), W_2 = (b_1, b_2, b_3, b_4)$, where a_i, b_j are i.i.d. uniform bits. The code is as follows.

X_1	X_2	X_3	X_4
$\emptyset$	a_1	$a_1 + b_1$	b_1
a_2	$\emptyset$	b_2	$a_2 + b_2$
b_3	$a_3 + b_3$	a_3	$\emptyset$
$a_4 + b_4$	b_4	$\emptyset$	a_4

(80)

We have 2 decoding sets for each source symbol.

$$\mathcal{S}_1 = \{\{X_1, X_2\}, \{X_3, X_4\}\} \tag{81}$$
$$\mathcal{S}_2 = \{\{X_1, X_4\}, \{X_2, X_3\}\} \tag{82}$$

Correctness is easy to verify (i.e., from any decoding in $\mathcal{S}_k$, we can decode W_k). Perfect smoothness is also easily verified, as each coded symbol appears once and only once in the decoding sets for any message.

Inspecting the code in (80), we see that each row forms a feasible sub-code and the rows are some permutations of each other (note however, this is a highly-structured permutation that preserves the same upload cost and is particularly distinct from time-sharing). This is in fact the key idea of our SLDC and we will further develop it in the following example and in the general proof.

B. Example 2: $N = 3, K = 3$

When $N = 3, K = 3$, the capacity is $C^*_{\text{ULDC}}(N = 3, K = 3) = \frac{L_w}{L_x} = 3(1 + \frac{1}{3} + \frac{1}{3^2})^{-1} = \frac{27}{13} = \frac{54}{26}$. We present an SLDC with length 27, where each source symbol is comprised of $L_w = 54$ bits and each coded symbol has $L_x = 26$ bits.

Each source symbol is divided into 27 sub-source-symbols and each sub-source-symbol has 2 bits. Denote W_1 as the collection of $(a_1^{(\gamma_1,\gamma_2,\gamma_3)}, a_2^{(\gamma_1,\gamma_2,\gamma_3)})$ for all $\gamma_1, \gamma_2, \gamma_3$, where $\gamma_1, \gamma_2, \gamma_3 \in [0:2]$ are indices for sub-source-symbol. Similarly, W_2 is the collection of $(b_1^{(\gamma_1,\gamma_2,\gamma_3)}, b_2^{(\gamma_1,\gamma_2,\gamma_3)})$ and W_3 is the collection of $(c_1^{(\gamma_1,\gamma_2,\gamma_3)}, c_2^{(\gamma_1,\gamma_2,\gamma_3)})$. a_i, b_j, c_l are i.i.d. uniform bits. $a_0^{(\gamma_1,\gamma_2,\gamma_3)}, b_0^{(\gamma_1,\gamma_2,\gamma_3)}, c_0^{(\gamma_1,\gamma_2,\gamma_3)}$ are set to 0.

To simplify the notation, we denote the $N^K = 27$ coded symbols as X_{p_1,p_2,p_3} where $p_i \in [0:2], i \in [1:3]$. These 27 coded symbols are divided into 3 groups depending on the value of $p_1 + p_2 + p_3$, so that x_{p_1,p_2,p_3} belongs to Group $p_1+p_2+p_3$ (modulo 3), and each group has 9 coded symbols.

Each coded symbol is similarly comprised of 27 sub-coded-symbols, denoted as $X^{(\gamma_1,\gamma_2,\gamma_3)}_{p_1,p_2,p_3}$. When there will be no confusion from the context, we simply denote $X^{(\gamma_1,\gamma_2,\gamma_3)}_{p_1,p_2,p_3}$ as x_{p_1,p_2,p_3}. To determine the value of x_{p_1,p_2,p_3}, we use $p_k + \gamma_k$ as the bit sub-script for the $(\gamma_1, \gamma_2, \gamma_3)$ sub-source-symbol of $W_k, k \in [1:3]$ and take the sum of all 3 bits, i.e., $x_{p_1,p_2,p_3} = a^{(\gamma_1,\gamma_2,\gamma_3)}_{p_1+\gamma_1} + b^{(\gamma_1,\gamma_2,\gamma_3)}_{p_2+\gamma_2} + c^{(\gamma_1,\gamma_2,\gamma_3)}_{p_3+\gamma_3}$. For example, the symbol denoted as $x_{0,1,2} = a_{\gamma_1} + b_{1+\gamma_2} + c_{2+\gamma_3}$, is comprised of 27 sub-coded-symbols corresponding to all 27 values of $(\gamma_1, \gamma_2, \gamma_3) \in [0:2]^3$, such as $a_1 + b_0 + c_1$ when $(\gamma_1, \gamma_2, \gamma_3) = (1, 2, 2)$. All these symbols belong to Group 0 because $p_1 + p_2 + p_3 = 0 + 1 + 2 = 0 \bmod 3$.

The decoding constraints are as follows (easy to verify from the table above).

$$\begin{aligned}&\text{From } x_{p_1,p_2,p_3}, x_{p_1+1,p_2,p_3}, x_{p_1+2,p_2,p_3},\\ &\text{we can decode } a_{p_1}, a_{p_1+1}, a_{p_1+2}.\end{aligned} \tag{84}$$

$$\begin{aligned}&\text{From } x_{p_1,p_2,p_3}, x_{p_1,p_2+1,p_3}, x_{p_1,p_2+2,p_3},\\ &\text{we can decode } b_{p_2}, b_{p_2+1}, b_{p_2+2}.\end{aligned} \tag{85}$$

$$\begin{aligned}&\text{From } x_{p_1,p_2,p_3}, x_{p_1,p_2,p_3+1}, x_{p_1,p_2,p_3+2},\\ &\text{we can decode } c_{p_3}, c_{p_3+1}, c_{p_3+2}.\end{aligned} \tag{86}$$

That is, if we pick one coded symbol from each group such that their subscripts only differ in the k^{th} digit, then we can decode W_k. Further, this claim remains valid for any realization of $(\gamma_1, \gamma_2, \gamma_3)$. As a result, for each source symbol, we have 9 decoding sets and each coded symbol appears once and only once in the decoding sets, leading to correctness and perfect smoothness.

Finally, we note that each coded symbol contains 26 bits, although it contains 27 sub-coded-symbols (each sub-coded-symbol is one equation, thus at most 1 bit). This follows from the observation that for any p_1, p_2, p_3, there exists one and only one realization of $(\gamma_1, \gamma_2, \gamma_3)$ such that $p_i + \gamma_i = 0$ (modulo 3), $\forall i \in [1:3]$, $X^{(\gamma_1,\gamma_2,\gamma_3)}_{p_1,p_2,p_3} = a_0 + b_0 + c_0 = 0$ and nothing needs to be stored. For all other cases, the sub-coded-symbol is 1 bit. Therefore, $L_x = 26$ and the SLDC achieves the capacity.

C. General Proof for Arbitrary N, K

The general proof follows from the ideas presented in previous sections. For any N, K, the capacity is $C^*_{\text{ULDC}}(N, K) = \frac{L_w}{L_x} = N(1 + \frac{1}{N} + \cdots + \frac{1}{N^{K-1}})^{-1} = \frac{N^K(N-1)}{N^K-1}$. We present an SLDC with length $M = N^K$, where each source symbol is comprised of $L_w = N^K(N-1)$ bits and each coded symbol has $L_x = N^K - 1$ bits.

Each source symbol is divided into N^K sub-source-symbols and each sub-source-symbol has $N-1$ bits. Define $\vec{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_K)$.

$$\begin{aligned}W_k &= (W_k^{(0,0,\cdots,0)}, W_k^{(0,0,\cdots,1)}, \cdots, W_k^{(N-1,N-1,\cdots,N-1)}),\\ &\quad \forall k \in [1:K].\end{aligned} \tag{87}$$

$$\begin{aligned}W_k^{\vec{\gamma}} &= (W_{k,0}^{\vec{\gamma}}, W_{k,1}^{\vec{\gamma}}, W_{k,2}^{\vec{\gamma}}, \cdots, W_{k,N-1}^{\vec{\gamma}}),\\ &\quad \forall i \in [1:K], \forall \gamma_i \in [0:N-1].\end{aligned} \tag{88}$$

$$W_{k,0}^{\vec{\gamma}} \triangleq 0. \tag{89}$$

Define $\vec{p} = (p_1, p_2, \cdots, p_K)$. The N^K coded symbols are denoted as $X_{\vec{p}}$, where $i \in [1:K], p_i \in [0:N-1]$. These N^K coded symbols are divided into N groups depending on the value of $\sum_{i=1}^K p_i$ (modulo N), so that $X_{\vec{p}}$ belongs to Group $\sum_{i=1}^K p_i$ (modulo N) and each group has N^{K-1} coded symbols.

$$\forall n \in [0:N-1],\ \text{Group } n = \left\{ X_{\vec{p}} : \sum_{i=1}^K p_i \ (\text{modulo } N) \ = n \right\}. \tag{90}$$

Each coded symbol is similarly comprised of N^K sub-coded-symbols and each sub-coded-symbol is designed as follows.

$$X_{\vec{p}} = (X_{\vec{p}}^{(0,0,\cdots,0)}, X_{\vec{p}}^{(0,0,\cdots,1)}, \cdots, X_{\vec{p}}^{(N-1,N-1,\cdots,N-1)}), \forall \vec{p} \tag{91}$$

$$X_{\vec{p}}^{\vec{\gamma}} = W_{1,p_1+\gamma_1}^{\vec{\gamma}} + W_{2,p_2+\gamma_2}^{\vec{\gamma}} + \cdots + W_{K,p_K+\gamma_K}^{\vec{\gamma}}, \forall \vec{\gamma} \tag{92}$$

For each message, we have N^{K-1} decoding sets. For given $p_1, \cdots, p_{k-1}, p_{k+1}, \cdots, p_K$, define $p_k^* = N - (p_1 + \cdots + p_{k-1} + p_{k+1} + \cdots + p_K)$ (modulo N). The subscripts below are understood modulo N.

$$\begin{aligned}&\forall k \in [1:K], \forall i \in [1:k-1] \cup [k+1:K],\\ &\forall p_i \in [0:N-1],\end{aligned} \tag{93}$$

Group 0	Group 1	Group 2
$x_{0,0,0}=a_{\gamma_1}+b_{\gamma_2}+c_{\gamma_3}$	$x_{0,0,1}=a_{\gamma_1}+b_{\gamma_2}+c_{1+\gamma_3}$	$x_{0,0,2}=a_{\gamma_1}+b_{\gamma_2}+c_{2+\gamma_3}$
$x_{1,1,1}=a_{1+\gamma_1}+b_{1+\gamma_2}+c_{1+\gamma_3}$	$x_{0,1,0}=a_{\gamma_1}+b_{1+\gamma_2}+c_{\gamma_3}$	$x_{0,2,0}=a_{\gamma_1}+b_{2+\gamma_2}+c_{\gamma_3}$
$x_{2,2,2}=a_{2+\gamma_1}+b_{2+\gamma_2}+c_{2+\gamma_3}$	$x_{1,0,0}=a_{1+\gamma_1}+b_{\gamma_2}+c_{\gamma_3}$	$x_{2,0,0}=a_{2+\gamma_1}+b_{\gamma_2}+c_{\gamma_3}$
$x_{0,1,2}=a_{\gamma_1}+b_{1+\gamma_2}+c_{2+\gamma_3}$	$x_{0,2,2}=a_{\gamma_1}+b_{2+\gamma_2}+c_{2+\gamma_3}$	$x_{0,1,1}=a_{\gamma_1}+b_{1+\gamma_2}+c_{1+\gamma_3}$
$x_{0,2,1}=a_{\gamma_1}+b_{2+\gamma_2}+c_{1+\gamma_3}$	$x_{2,0,2}=a_{2+\gamma_1}+b_{\gamma_2}+c_{2+\gamma_3}$	$x_{1,0,1}=a_{1+\gamma_1}+b_{\gamma_2}+c_{1+\gamma_3}$
$x_{1,0,2}=a_{1+\gamma_1}+b_{\gamma_2}+c_{2+\gamma_3}$	$x_{2,2,0}=a_{2+\gamma_1}+b_{2+\gamma_2}+c_{\gamma_3}$	$x_{1,1,0}=a_{1+\gamma_1}+b_{1+\gamma_2}+c_{\gamma_3}$
$x_{2,0,1}=a_{2+\gamma_1}+b_{\gamma_2}+c_{1+\gamma_3}$	$x_{1,1,2}=a_{1+\gamma_1}+b_{1+\gamma_2}+c_{2+\gamma_3}$	$x_{2,2,1}=a_{2+\gamma_1}+b_{2+\gamma_2}+c_{1+\gamma_3}$
$x_{1,2,0}=a_{1+\gamma_1}+b_{2+\gamma_2}+c_{\gamma_3}$	$x_{1,2,1}=a_{1+\gamma_1}+b_{2+\gamma_2}+c_{1+\gamma_3}$	$x_{2,1,2}=a_{2+\gamma_1}+b_{1+\gamma_2}+c_{2+\gamma_3}$
$x_{2,1,0}=a_{2+\gamma_1}+b_{1+\gamma_2}+c_{\gamma_3}$	$x_{2,1,1}=a_{2+\gamma_1}+b_{1+\gamma_2}+c_{1+\gamma_3}$	$x_{1,2,2}=a_{1+\gamma_1}+b_{2+\gamma_2}+c_{2+\gamma_3}$

(83)

$$\mathcal{S}_k=\bigcup_{\forall p_i,i\neq k}\{X_{p_1,\cdots,p_{k-1},p_k^*,p_{k+1},\cdots,p_K},\ X_{p_1,\cdots,p_{k-1},p_k^*+1,p_{k+1},\cdots,p_K},\cdots\ X_{p_1,\cdots,p_{k-1},p_k^*+N-1,p_{k+1},\cdots,p_K}\} \quad (94)$$

where each decoding set is comprised of one and only one coded symbol from each group.

We verify that the code is correct, perfectly smooth and capacity achieving.

First, to show that the code is correct, we verify that from any coding set in $\mathcal{S}_k$, we can decode $W_k, \forall k \in [1:K]$. Consider any realization of $p_1,\cdots,p_{k-1},p_{k+1},\cdots,p_K$. From (92), we consider the N coded symbols and obtain that $\forall\vec{\gamma}$,

$$X^{\vec{\gamma}}_{p_1,\cdots,p_{k-1},p_k^*,p_{k+1},\cdots,p_K}=\sum_{j=1,j\neq k}^{K}W^{\vec{\gamma}}_{j,p_j+\gamma_j}+W^{\vec{\gamma}}_{k,p_k^*+\gamma_k} \quad (95)$$

$$X^{\vec{\gamma}}_{p_1,\cdots,p_{k-1},p_k^*+1,p_{k+1},\cdots,p_K}=\sum_{j=1,j\neq k}^{K}W^{\vec{\gamma}}_{j,p_j+\gamma_j}+W^{\vec{\gamma}}_{k,p_k^*+1+\gamma_k} \quad (96)$$

$$\cdots \quad (97)$$

$$X^{\vec{\gamma}}_{p_1,\cdots,p_{k-1},p_k^*+N-1,p_{k+1},\cdots,p_K}=\sum_{j=1,j\neq k}^{K}W^{\vec{\gamma}}_{j,p_j+\gamma_j}+W^{\vec{\gamma}}_{k,p_k^*+N-1+\gamma_k} \quad (98)$$

Note that the interference about source symbols $W_{\overline{k}}$ is the same in the above N equations and the desired sub-source-symbol has $N-1$ bits. So we can decode all $N-1$ desired bits, $W^{\vec{\gamma}}_{k,1},W^{\vec{\gamma}}_{k,2},\cdots,W^{\vec{\gamma}}_{k,N-1}$. Repeating the same decoding procedure for all $\vec{\gamma}$, we decode all $L_w=N^K(N-1)$ bits in W_k. Therefore the LDC is correct.

Second, the code is perfectly smooth because from (94), we note that for any source symbol W_k and for any Group $n\in[0:N-1]$, any coded symbol $X_{\vec{p}}$ (from Group n) appears once and only once. Therefore, the definition of perfect smoothness (refer to Definition 4) is satisfied.

Finally, we prove that the code achieves the capacity. To this end, we verify that $H(X_{\vec{p}})=L_x=N^K-1,\forall\vec{p}$. Note that each coded symbol contains N^K sub-coded-symbols, and there exists one and only one sub-coded-symbol that is constantly zero. That is, for any given $\vec{p}$, when

$$\gamma_k=-p_k \text{ (modulo } N),\forall k\in[1:K], \quad (99)$$

we have $X^{\vec{\gamma}}_{\vec{p}}=\sum_{k=1}^{K}W^{\vec{\gamma}}_{k,0}=0$ (refer to (92), (89)). The proof is thus complete. ■

Remark: One might wonder if our SLDC (and the corresponding upload optimal PIR$_{\max}$ scheme) can be constructed from the upload optimal PIR$_{ave}$ scheme in [14] by symmetrization (e.g., as described in Section 5 of [14]), as one sub-code in our scheme is similar to the PIR$_{ave}$ scheme in [14]. This does not work because general symmetrization techniques will increase the upload proportional to the number of concatenations of sub-codes, while in our PIR$_{\max}$ scheme, the upload cost of the concatenated code remains the same as that of one sub-code (i.e., $(K-1)\log(N)$ per database). Therefore, our code is is not constructed by generic symmetrizations. Instead, the specific sub-code has a permutation-invariant property that allows us to shift the symbol indices while retaining the same decoding structure (refer to (92)).

VII. Proof of Corollaries 2.2 and 2.3

For the converse, it suffices to provide the proof for RIR$_{\max}$, which automatically implies the converse for PIR$_{\max}$. The converse proof for RIR$_{\max}$ is as follows.

To set up the proof by contradiction, suppose on the contrary that we have a capacity achieving RIR$_{\max}$ scheme such that the upload cost from some database is strictly less than $(K-1)\log(N)$, i.e., there exists a set of answers $\mathcal{X}^{[n]}$ from one database such that $|\mathcal{X}^{[n]}|<N^{K-1}$. Then by Observation 1, we have a capacity achieving ULDC such that there exists at least one group of strictly fewer than N^{K-1} coded symbols (this group corresponds to the set of answers $\mathcal{X}^{[n]}$ from the database in PIR) such that any decoding set must contain one coded symbol from this group (as any decoding set in PIR must contain one answer from each database, including the one with answer set $\mathcal{X}^{[n]}$). Note that for any full N-ary tree (refer to Figure 4), the N^K leaf nodes form N^{K-1} decoding sets. As any one of these N^{K-1} decoding sets must contain one coded symbol from $\mathcal{X}^{[n]}$ (where $|\mathcal{X}^{[n]}|<N^{K-1}$), the leaf nodes must have at least two identical coded symbols. Then from the converse proof of Theorem 2, it follows that the ULDC can not achieve capacity and we arrive at the contradiction.

For the achievability, it suffices to provide the proof for PIR$_{\max}$, which automatically implies the achievability for RIR$_{\max}$. The achievable scheme for PIR$_{\max}$ is based on the SLDC from Theorem 2. The SLDC has an N-partite property, that *any decoding set is comprised of one symbol from each*

group. Group $n, n \in [0 : N-1]$ maps to answer set $\mathcal{X}^{[n+1]}$, i.e., the coded symbols from Group $n, n \in [0 : N-1]$ of the SLDC (refer to (90)) form the answers from the $(n+1)^{th}$ database in $\text{PIR}_{\max}$. The decoding supersets $\mathcal{S}_{[1:K]}$ of $\text{PIR}_{\max}$ are chosen to be the same as the decoding supersets $\mathcal{S}_{[1:K]}$ of the SLDC. Now if the user wishes to retrieve W_k, the user simply asks for one of the decoding sets for W_k of the SLDC, uniformly over all N^{K-1} choices of decoding sets (refer to (94)). Thus, the user downloads exactly N answers, one from each database. The correctness and perfect smoothness of LDC translate to the correctness and privacy of $\text{PIR}_{\max}$ directly. ■

VIII. Discussion

We introduce the notion of capacity for LDC, and show that the capacity of ULDCs and SLDCs with K source symbols and locality N is $C^* = N\left(1+1/N+1/N^2+\cdots+1/N^{K-1}\right)^{-1}$. We further show that the minimum length of capacity achieving ULDCs and SLDCs is N^K. The results are translated into the context of $\text{PIR}_{\max}$ and $\text{RIR}_{\max}$, where we show that the capacity of $\text{RIR}_{\max}$ is equal to that of $\text{PIR}_{\max}$, and the minimum upload cost of both $\text{PIR}_{\max}$ and $\text{RIR}_{\max}$ is equal to $(K-1)\log N$.

In this work, we have focused on the capacity achieving regime for LDCs. That is, the number of bits in each coded symbol is equal to $1/C^*$ times the number of bits in each source symbol, $L_x = \frac{L_w}{C^*} = \frac{L_w(1-1/N^K)}{N-1} < \frac{L_w}{N-1}$. In other words, the size of each coded symbol is (sometimes much) smaller than the size of each source symbol, a regime that is rarely studied in classical coding theory or theoretical computer science. Specifically, when the coded symbol has the smallest size (capacity achieving), the code length M must be exponential, i.e., $M \geq N^K$ in order to preserve either universality or perfect smoothness. It is an interesting avenue for future work to study other rate regimes. In particular, the minimum symbol rate for which the code length is polynomial remains an interesting question.

As a final remark, we note that in the $\text{PIR}_{\max}$ problem formulation of this work, we have defined the $\max$ to be over all queries *and* all databases, as this formulation is the one that connects to LDCs and is consistent with most scenarios. Essentially, we restrict the downloads to be symmetric and constant over all databases. An alternative formulation could be defining the $\max$ to be only over all queries, e.g., this formulation was adopted in [15], where the downloads are constant for one database, but could be asymmetric across the databases. These two formulations have the same capacity, but could behave differently in terms of other metrics, such as message size, upload cost etc. It is an interesting question to compare these models and identify their similarities and differences.

References

[1] J. Katz and L. Trevisan, "On the efficiency of local decoding procedures for error-correcting codes," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. ACM, 2000, pp. 80–86.

[2] L. Trevisan, "Some applications of coding theory in computational complexity," in *Quaderni di Matematica*, no. 13, 2004, pp. 347–424.

[3] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[4] R. De Wolf, "Error-correcting data structures," in *26th International Symposium on Theoretical Aspects of Computer Science STACS 2009*. IBFI Schloss Dagstuhl, 2009, pp. 313–324.

[5] A. Gal and A. Mills, "Three-query locally decodable codes with higher correctness require exponential length," *ACM Transactions on Computation Theory (TOCT)*, vol. 3, no. 2, p. 5, 2012.

[6] A. Romashchenko, "Reliable computations based on locally decodable codes," in *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 2006, pp. 537–548.

[7] Y. Ishai and E. Kushilevitz, "On the hardness of information-theoretic multiparty computation," in *Advances in Cryptology-EUROCRYPT 2004*. Springer, 2004, pp. 439–455.

[8] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 1995, pp. 41–50.

[9] A. Beimel, Y. Ishai, and E. Kushilevitz, "General constructions for information-theoretic private information retrieval," *Journal of Computer and System Sciences*, vol. 71, no. 2, pp. 213–247, 2005.

[10] S. Yekhanin, "Locally Decodable Codes and Private Information Retrieval Schemes," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[11] ——, "Locally decodable codes," *Foundations and Trends in Theoretical Computer Science*, vol. 6, no. 3, pp. 139–255, 2012. [Online]. Available: http://dx.doi.org/10.1561/0400000030

[12] S. Kopparty and S. Saraf, "Local testing and decoding of high-rate error-correcting codes." in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 24, 2017, p. 126.

[13] H. Sun and S. A. Jafar, "The Capacity of Private Information Retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, July 2017.

[14] C. Tian, H. Sun, and J. Chen, "Capacity-achieving Private Information Retrieval Codes with Optimal Message Size and Upload Cost," *IEEE Transactions on Information Theory*, vol. 11, no. 65, pp. 7613–7627, 2019.

[15] H. Sun and S. A. Jafar, "Optimal download cost of private information retrieval for arbitrary message length," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2920–2932, 2017.

[16] ——, "The Capacity of Robust Private Information Retrieval with Colluding Databases," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2361–2370, April 2018.

[17] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. El Rouayheb, "Private information retrieval schemes for codec data with arbitrary collusion patterns," *IEEE International Symposium on Information Theory (ISIT)*, pp. 1908–1912, 2017.

[18] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private Information Retrieval from MDS Coded Data in Distributed Storage Systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[19] K. Banawan and S. Ulukus, "The Capacity of Private Information Retrieval from Coded Databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.

[20] H. Sun and S. A. Jafar, "The Capacity of Symmetric Private Information Retrieval," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 322–329, 2019.

[21] D. Asonov and J.-C. Freytag, "Repudiative information retrieval," *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pp. 32–40, 2002.

Hua Sun (S'12-M'17) received his B.E. in Communications Engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2011, M.S. in Electrical and Computer Engineering from University of California Irvine, USA, in 2013, and Ph.D. in Electrical Engineering from University of California Irvine, USA, in 2017. He is an Assistant Professor in the Department of Electrical Engineering at the University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage.

Dr. Sun received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016, an IEEE GLOBECOM Best Paper Award in 2016, and the University of California Irvine CPCC Fellowship for the year 2011-2012.

Syed Ali Jafar (S'99-M'04-SM'09-F'14) received his B. Tech. from IIT Delhi, India, in 1997, M.S. from Caltech, USA, in 1999, and Ph.D. from Stanford, USA, in 2003, all in Electrical Engineering. His industry experience includes positions at Lucent Bell Labs and Qualcomm. He is a Professor in the Department of Electrical Engineering and Computer Science at the University of California Irvine, Irvine, CA USA. His research interests include multiuser information theory, wireless communications and network coding.

Dr. Jafar is a recipient of the New York Academy of Sciences Blavatnik National Laureate in Physical Sciences and Engineering, the NSF CAREER Award, the ONR Young Investigator Award, the UCI Academic Senate Distinguished Mid-Career Faculty Award for Research, the School of Engineering Mid-Career Excellence in Research Award and the School of Engineering Maseeh Outstanding Research Award. His co-authored papers have received the IEEE Information Theory Society Paper Award, IEEE Communication Society and Information Theory Society Joint Paper Award, IEEE Communications Society Best Tutorial Paper Award, IEEE Communications Society Heinrich Hertz Award, IEEE Signal Processing Society Young Author Best Paper Award, IEEE Information Theory Society Jack Wolf ISIT Best Student Paper Award, and three IEEE GLOBECOM Best Paper Awards. Dr. Jafar received the UC Irvine EECS Professor of the Year award six times, in 2006, 2009, 2011, 2012, 2014 and 2017 from the Engineering Students Council, a School of Engineering Teaching Excellence Award in 2012, and a Senior Career Innovation in Teaching Award in 2018. He was a University of Canterbury Erskine Fellow in 2010, an IEEE Communications Society Distinguished Lecturer for 2013-2014, and an IEEE Information Theory Society Distinguished Lecturer for 2019-2020. Dr. Jafar was recognized as a Thomson Reuters/Clarivate Analytics Highly Cited Researcher and included by Sciencewatch among The World's Most Influential Scientific Minds in 2014, 2015, 2016, 2017 and 2018. He served as Associate Editor for IEEE Transactions on Communications 2004-2009, for IEEE Communications Letters 2008-2009 and for IEEE Transactions on Information Theory 2009-2012.