Privacy in Index Coding: k-Limited-Access Schemes

Mohammed Karmoose, Linqi Song, *Member, IEEE*, Martina Cardone, *Member, IEEE*, and Christina Fragouli, *Fellow, IEEE*

***Abstract*— In the traditional index coding problem, a server employs coding to send messages to a set of clients within the same broadcast domain. Each client already has some messages as side information and requests a particular unknown message from the server. All clients learn the coding matrix so that they can decode and retrieve their requested data. Our starting observation comes from the work by Karmoose *et al.*, which shows that learning the coding matrix can pose privacy concerns: it may enable a client to infer information about the requests and side information of other clients. In this paper, we mitigate this privacy concern by allowing each client to have limited access to the coding matrix. In particular, we design coding matrices so that each client needs only to learn some of (and not all) the rows to decode her requested message. We start by showing that this approach can indeed help mitigate that privacy concern. We do so by considering two different privacy metrics. The first one shows the attained privacy benefits based on a geometric interpretation of the problem. Differently, the second metric, referred to as maximal information leakage, provides upper bounds on: (i) the guessing power of the adversaries (i.e., curious clients) when our proposed approach is employed, and (ii) the effect of decreasing the number of accessible rows on the attained privacy. Then, we propose the use of k-limited-access schemes: given an index coding scheme that employs T transmissions, we create a k-limited-access scheme with $T_k \geq T$ transmissions, and with the property that each client needs at most k transmissions to decode her message. We derive upper and lower bounds on T_k for all values of k, and develop deterministic designs for these schemes, which are universal, i.e., independent of the coding matrix. We show that our schemes are order-optimal for some parameter regimes, and we propose heuristics that complement the universal schemes for the remaining regimes.**

***Index Terms*— Index coding, privacy, broadcasting, k-limited-access scheme, maximal information leakage.**

Manuscript received September 19, 2018; revised August 31, 2019; accepted November 16, 2019. Date of publication December 4, 2019; date of current version April 21, 2020. This work was supported in part by NSF under Award 1527550, Award 1514531, Award 1423271, Award 1314937, and Award 1740047; and in part by the Research Grants Council (RGC) of Hong Kong under Grant ECS 9048149. This work was presented in part at the 2017 International Symposium on Information Theory, in part at the 2017 Information Theory Workshop, and in part at the 2018 International Symposium on Information Theory.

M. Karmoose is with Samsung Semiconductors, Inc., San Diego, CA 92121 USA (e-mail: mkarmoose@ucla.edu).

L. Song is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: linqi.song@cityu.edu.hk).

M. Cardone is with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN 55404 USA (e-mail: cardo089@umn.edu).

C. Fragouli are with the Electrical and Computer Engineering Department, University of California at Los Angeles (UCLA), Los Angeles, CA 90095 USA (e-mail: christina.fragouli@ucla.edu).

Communicated by P. Sadeghi, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2019.2957577

I. Introduction

IT IS well recognized that broadcasting can offer significant bandwidth savings compared to point-to-point communication [4], [5], and could be leveraged in several wireless network applications. Use cases include Wi-Fi (cellular) networks where an access point (a base station) is connected to a set of Wi-Fi (cellular) devices through a wireless broadcast channel, and where devices request messages, such as YouTube videos. Another use case has recently emerged in the context of distributed computing [6], [7], where worker nodes exchange data among themselves to complete computational tasks.

A canonical setup which captures the essence of broadcast channels is the index coding framework [8]. In an index coding instance, a server is connected to a set of clients through a noiseless broadcast channel. The server has a database that contains a set of messages. Each client: 1) possesses a subset of the messages that she already knows, which is referred to as the *side information set*, and 2) requests a message from the database which is not in her side information set. The server has full knowledge of the requests and side information sets of all clients. A *linear index code* (or *index code* in short)[1] is a linear coding scheme that comprises a set of coded broadcast transmissions which allow each client to decode her requested message using her side information set. The goal is to find an index code which uses the smallest possible number of broadcast transmissions. The key ingredient in designing efficient (i.e., with a small number of transmissions) index codes is the use of coding across messages.

The starting observation of this work is that, using coding over broadcast channels can cause privacy risks. In particular, a curious client may infer information about the requests and side information sets of other clients, which can be deemed sensitive by their owners. For example, consider a set of clients that use a server to download YouTube videos. Although YouTube videos are publicly available, a client requesting a video about a medical condition may not wish for others to learn her request, or learn what are other videos that she has already downloaded.

To illustrate why coding can create privacy leakage, consider the index coding instance shown in Figure 1. A server possesses a set of 5 messages, which we refer to as $\mathbf{b}_1$ to $\mathbf{b}_5$. The server is connected to a set of 4 clients: client 1 wants message $\mathbf{b}_1$ and has as side information message $\mathbf{b}_2$; client 2

[1] In this work, we solely focus on linear index codes.

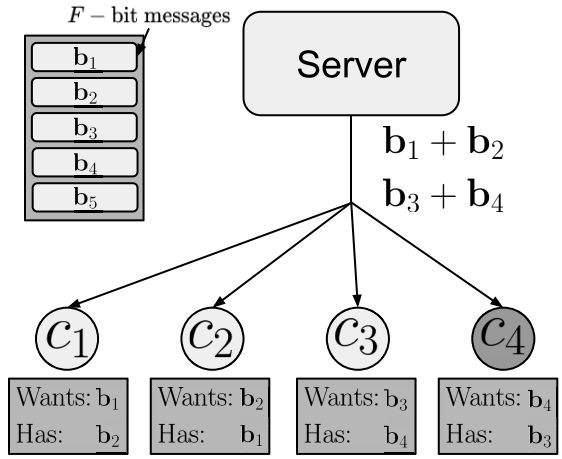


Fig. 1. An index coding example with 5 messages and 4 clients. Each client wants one message and has another as shown above. The optimal index code consists of sending the two transmissions $\mathbf{b}_1+\mathbf{b}_2$ and $\mathbf{b}_3+\mathbf{b}_4$.

wants $\mathbf{b}_2$ and has $\mathbf{b}_1$; client 3 wants $\mathbf{b}_3$ and has $\mathbf{b}_4$; and client 4 wants $\mathbf{b}_4$ and has $\mathbf{b}_3$. In this case, an optimal (i.e., with the minimum number of transmissions) index code consists of sending 2 transmissions, namely $\mathbf{b}_1+\mathbf{b}_2$ and $\mathbf{b}_3+\mathbf{b}_4$: it is easy to see that each client can decode the requested message from one of these transmissions using the side information. However, this index code can allow curious clients to violate the privacy of other clients who share the broadcast channel, by learning information that pertains to their requests and/or side information sets. For example, assume that client 4 is curious. Upon learning the two transmissions, client 4 knows that nobody is requesting message $\mathbf{b}_5$. Moreover, she knows that if a client is requesting $\mathbf{b}_1$ or $\mathbf{b}_2$ (similarly, $\mathbf{b}_3$ or $\mathbf{b}_4$), then this client should have the other message as side information in order to decode the requested message.

The solution that we propose to limit this privacy leakage stems from the following observation: it may not be necessary to provide clients with the entire set of broadcast transmissions. Instead, each client can be given access, and learn the coding operations, for only a subset of the transmissions, i.e., the subset that would allow her to decode the message that she requested. Consider again the example in Figure 1. The optimal index code consists of two transmissions. However, each client is able to decode her request using exactly one of the two transmissions. Therefore, if each client only learns the coding coefficients for the transmission that she needs, then she will have no knowledge of the content of the other transmission, and thus would have less information about the requests of the other clients. Limiting the access of each client to just one out of the two transmissions was possible for this particular example; however, it is not the case that every index code has this property.

Our approach in this paper builds on the idea described above. We primarily turn our attention to a specific technical challenge, namely how to design index codes where each client is limited in her access to the transmitted messages. Our method stems from our recent observation in [1], where we showed that the knowledge of the coding matrix may enable a client to infer information about the requests and side information of other clients. Thus, limiting the access of clients to the coding matrix promises privacy benefits. In particular, given an index coding instance that uses T transmissions, we ask: Can we limit the access of each client to at most $k \leq T$ transmissions, while still allowing each client to decode her requested message? In other words, for a given index coding instance, what is the best (in terms of number of transmissions) index code that we can design such that each client is able to decode her request using at most k out of these transmissions? Towards this end, we propose the use of *k-limited-access schemes*, that transform the coding matrix so as to restrict each client to access at most k rows of the transformed matrix, as opposed to the whole of it. Our contributions can be summarized in two major directions: i) we show formally and operationally how limiting the access of clients can increase the privacy guarantees, and ii) we provide deterministic constructions of our proposed k-limited-access schemes. In more details, our contributions include:

- We formalize the intuition that using k-limited-access-schemes can indeed increase the attained level of privacy against curious clients. We demonstrate this by using two privacy metrics. Our first proposed metric is an *entropy-based* metric, which is inspired by a geometric interpretation of the considered problem. The second proposed metric is known as the *Maximal Information Leakage* (MIL). Using MIL, we show how our scheme affects the guessing power of the adversaries (i.e., curious clients). We also show how, for a particular regime, the guessing power of the adversaries vanishes asymptotically. Finally, for both metrics, we show that the attained level of privacy is linearly dependent on the value of k, i.e., privacy increases linearly with the number of rows of the coding matrix that we hide.
- We design polynomial time (in the number of clients) universal k-limited-access schemes (i.e., that do not depend on the structure of the coding matrix) that require a simple matrix multiplication. We prove that these schemes are order-optimal in some regimes, for instance when $k \geq \lceil T/2 \rceil$. Interestingly, when k is larger than a threshold, these schemes enable to restrict the amount of access to half of the coding matrix with an overhead of exactly one additional transmission. This result indicates that some privacy-bandwidth trade-off points can be achieved with minimal overhead.
- We propose algorithms that depend on the structure of the coding matrix and show that, for some parameter regimes, they provide improved performance with respect to the universal schemes mentioned above. These schemes use a graph-theory representation of the problem, and are optimal for some special instances.
- We provide analytical and numerical performance evaluations of our schemes. We show how our proposed k-limited-access schemes provide a bandwidth-privacy trade-off, namely how much bandwidth usage (i.e., number of transmissions) is needed to achieve a certain level of privacy (captured by the value of k). We identify the parameter regimes where our proposed schemes

provide a trade-off curve that is close to the lower bound. For the remaining regimes, we show through numerical evaluations that our proposed algorithms give an average performance that is close to the lower bound.

The paper is organized as follows. Section II introduces our notation, formulates the problem, and gives a geometric interpretation. Section III discusses how k-limited-access schemes limit the privacy leakage. Section IV shows the construction of k-limited-access schemes and proves their order-optimality in some parameter regimes. Section V designs algorithms which are better-suited for the remaining regimes. Section VI discusses related work and Section VII concludes the paper. Some of the proofs are delegated to the appendices. The results in this paper are presented in part in [1]–[3].

II. Notation, Problem Formulation and Geometric Interpretation

Notation. Calligraphic letters indicate sets; $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$; $[n]$ is the set of integers $\{1, \cdots, n\}$; boldface lower case letters denote vectors and boldface upper case letters indicate matrices; given a vector $\mathbf{b}$, b_i indicates the i-th element of $\mathbf{b}$; given matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{B} \subset_k \mathbf{A}$ indicates that $\mathbf{B}$ is formed by a set of k rows of $\mathbf{A}$; $\mathbf{0}_j$ is the all-zero row vector of dimension j; $\mathbf{1}_j$ denotes a row vector of dimension j of all ones and $\mathbf{I}_j$ is the identity matrix of dimension j; $\mathbf{e}_i^j$ is the all-zero row vector of length j with a 1 in position i; for all $x \in \mathbb{R}$, the floor and ceiling functions are denoted with $\lfloor x \rfloor$ and $\lceil x \rceil$, respectively; logarithms are in base 2; $\mathbf{Pr}(X)$ refers to the probability of event X.

Index Coding. We consider an index coding instance, where a server has a database $\mathcal{B}$ of m messages $\mathcal{B} = \{\mathbf{b}_{\mathcal{M}}\}$, where $\mathcal{M} = [m]$ is the set of message indices, and $\mathbf{b}_j \in \mathbb{F}_2^F, j \in \mathcal{M}$, with F being the message size, and where operations are done over the binary field. The server is connected through a broadcast channel to a set of clients $\mathcal{C} = \{c_{\mathcal{N}}\}$, where $\mathcal{N} = [n']$ is the set of client indices. We assume that $m \geq n'$. Each client $c_i, i \in \mathcal{N}$, has a subset of the messages $\{\mathbf{b}_{\mathcal{S}_i}\}$, with $\mathcal{S}_i \subset \mathcal{M}$, as side information and requests a new message $\mathbf{b}_{q_i}$ with $q_i \in \mathcal{M} \setminus \mathcal{S}_i$ that she does not have. We assume that the server employs a *linear code*, i.e., it designs a set of broadcast transmissions that are linear combinations of the messages in $\mathcal{B}$. The index coding algorithm used to design the linear code is only known by the server and not by the clients. The linear index code can be represented as $\mathbf{AB} = \mathbf{Y}$, where $\mathbf{A} \in \mathbb{F}_2^{T \times m}$ is the coding matrix, $\mathbf{B} \in \mathbb{F}_2^{m \times F}$ is the matrix of all the messages and $\mathbf{Y} \in \mathbb{F}_2^{T \times F}$ is the resulting matrix of linear combinations. Upon receiving $\mathbf{Y}$, client $c_i, i \in \mathcal{N}$, employs linear decoding to decode the requested message $\mathbf{b}_{q_i}$.

Problem Formulation. In [8], it was shown that the index coding problem is equivalent to the rank minimization of an $n' \times m$ matrix $\mathbf{G} \in \mathbb{F}_2^{n' \times m}$, whose i-th row $\mathbf{g}_i$, $i \in [n']$, has the following properties: (i) has a 1 in the position q_i (i.e., the index of the message requested by client c_i), (ii) has a 0 in the j-th position for all $j \in \mathcal{M} \setminus \mathcal{S}_i$, (iii) can have either 0 or 1 in all the remaining positions. For instance, with reference to the example in Figure 1, we would have

$$\mathbf{G} = \begin{bmatrix} 1 & \star & 0 & 0 & 0 \\ \star & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \star & 0 \\ 0 & 0 & \star & 1 & 0 \end{bmatrix},$$

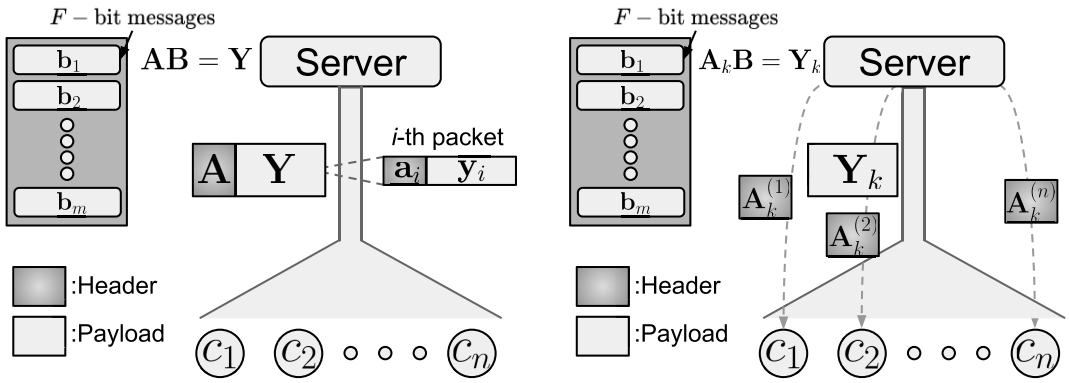
where $\star$ can be either 0 or 1. It was shown in [8] that finding an optimal linear coding scheme (i.e., with minimum number of transmissions) is equivalent to completing $\mathbf{G}$ (i.e., assigning values to the $\star$ components of $\mathbf{G}$) so that it has the minimum possible rank. Once we have completed $\mathbf{G}$, we can use a basis of the row space of $\mathbf{G}$ (of size $T = \text{rank}(\mathbf{G})$) as a coding matrix $\mathbf{A}$. In this case, client c_i can construct $\mathbf{g}_i$ as a linear combination of the rows of $\mathbf{A}$, i.e., c_i performs the decoding operation $\mathbf{d}_i\mathbf{AB} = \mathbf{d}_i\mathbf{Y}$, where $\mathbf{d}_i \in \mathbb{F}_2^T$ is the decoding row vector of c_i chosen such that $\mathbf{d}_i\mathbf{A} = \mathbf{g}_i$. Finally, client c_i can successfully decode $\mathbf{b}_{q_i}$ by subtracting from $\mathbf{d}_i\mathbf{Y}$ the messages corresponding to the non-zero entries of $\mathbf{g}_i$ (other than the requested message). We remark that any linear index code that satisfies all clients with T transmissions (where T is not necessarily optimal) – and can be obtained by any index code design algorithm [9]–[11] – corresponds to a completion of $\mathbf{G}$ (i.e., given $\mathbf{A} \in \mathbb{F}_2^{T \times m}$, we can create a corresponding $\mathbf{G}$ in polynomial time). Such a matrix $\mathbf{G}$ is not necessarily unique. However, it is of rank at most T and follows the structure described above. The following observation on the matrix $\mathbf{G}$ is important for next discussions: although the n' clients may not be identical (i.e., they do not share the same requests and side information sets), the matrix $\mathbf{G}$ does not necessarily have n' distinct vectors. Consider for example the situation where $n' = 3$, $q_1 = 1$ and $\mathcal{S}_1 = \{2\}$, $q_2 = 2$ and $\mathcal{S}_2 = \{1\}$ and $q_3 = 3$ and $\mathcal{S}_3 = \emptyset$. Let $\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. Then a corresponding matrix $\mathbf{G}$ is

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In what follows, we assume that a matrix $\mathbf{G}$ of n' rows has $n \leq n'$ *distinct* rows; without loss of generality, we assume that these distinct vectors are the first n vectors of $\mathbf{G}$, i.e., $\mathbf{g}_i, i \in [n]$. Since identical vectors are reconstructed identically using the matrix $\mathbf{A}$, it therefore suffices to focus on reconstructing the n distinct vectors $\mathbf{g}_i, i \in [n]$. To further simplify the problem, we henceforth assume the existence of only n clients $c_{[n]}$ with corresponding n distinct rows $\mathbf{g}_{[n]}$; more clients with corresponding rows in $\mathbf{G}$ that are identical to $\mathbf{g}_{[n]}$ would apply the same decoding operations as the clients $c_{[n]}$.

In our problem formulation, we assume that we start with a linear index code $\mathbf{A} \in \mathbb{F}_2^{T \times m}$, and a particular realization of a corresponding matrix $\mathbf{G}$ of rank T with n distinct rows.[2] Then, we ask: *Given n distinct vectors $\mathbf{g}_i$, $i \in [n]$, in a T-dimensional space, can we find a minimum-size set $\mathcal{A}_k$ with $T_k \geq T$ vectors, such that each $\mathbf{g}_i$ can be expressed as a linear combination of at most k vectors in $\mathcal{A}_k$ (with*

[2]We remark that our scheme starts by assuming a particular realization of the matrix $\mathbf{G}$. Optimizing over the choice of the matrix $\mathbf{G}$ is out of scope of this work.



(a) Conventional Transmission Protocol.

(b) Proposed Transmission Protocol.

Fig. 2. A comparison between the conventional and the proposed transmission protocols. The proposed transmission protocol incurs a negligible increase in the transmission overhead when both n and m are $o(F)$.

$1 \leq k \leq T$)? The vectors in $\mathcal{A}_k$ form the rows of the coding matrix $\mathbf{A}_k$ that we will employ. Then by definition, client c_i will be able to reconstruct $\mathbf{g}_i$ using the matrix $\mathbf{A}_k^{(i)} \subset_k \mathbf{A}_k$.We can equivalently restate the question as follows: *Given a coding matrix* $\mathbf{A}$*, can we find* $\mathbf{P} \in \mathbb{F}_2^{T_k \times T}$*, with* T_k *as small as possible, such that* $\mathbf{A}_k = \mathbf{PA}$ *and each row of* $\mathbf{G}$ *can be reconstructed by combining at most* k *rows of* $\mathbf{A}_k$*?* Note that $k = T$ corresponds to the conventional transmission scheme of an index coding problem for which $\mathbf{P} = \mathbf{I}_T$. In the remainder of the paper we will refer to a scheme that chooses $\mathbf{A}_k$ to be the coding matrix as k-limited-access scheme.

Transmission Protocol. In order to realize the privacy benefits of using k-limited-access schemes – which we will thoroughly illustrate in Section III – we propose a different transmission protocol for the index coding setup. Figure 2 shows both the conventional and the proposed transmission protocols. In the conventional protocol, the server designs a set of T packets, each corresponding to an equation from the set of equations $\mathbf{AB} = \mathbf{Y}$. As shown in Figure 2(a), packet $i \in [T]$ consists of (i) a payload which contains the linear combination $\mathbf{y}_i$ and (ii) a header which contains the coefficients $\mathbf{a}_i$ used to create the equation. In the conventional protocol, the server sends these packets (both headers and payloads) on the broadcast channel to all clients. Our proposed protocol, however, operates differently. Specifically, the server generates packets which correspond to the set of equations $\mathbf{A}_k\mathbf{B} = \mathbf{Y}_k$ in a way that is similar to the conventional protocol. The server then sends *only* the payloads of these packets on the broadcast channel. Differently, the server sends the coefficients corresponding to *only* $\mathbf{A}_k^{(i)} \subset_k \mathbf{A}_k$ to client c_i using a private key or a dedicated private channel (e.g., the same channel used by c_i to convey her request to the server). Thus, using a k-limited-access scheme incurs an extra transmission overhead to privately convey the coding vectors. In particular, the total number of transmitted bits C_k can be upper bounded as $\mathrm{C}_k \leq nkm + T_kF$, while the total number of transmitted bits C using a conventional scheme is $\mathrm{C} = T(F + m)$. The extra overhead incurred is negligible in comparison to the broadcast transmissions that convey the encoded messages when n and m are both $o(F)$, which is a reasonable assumption for large file sizes (for instance, when sharing YouTube videos).

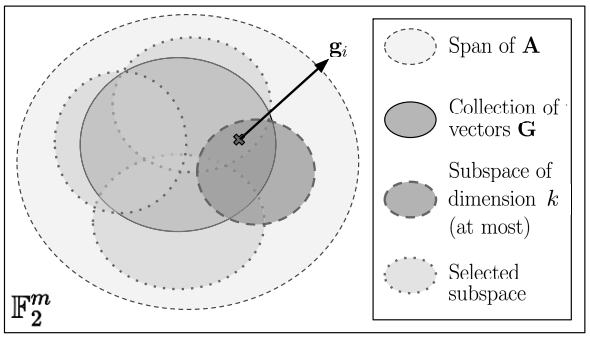


Fig. 3. A geometric interpretation of k-limited-access schemes. An index code $\mathbf{A}$ is obtained from a particular filling of the matrix $\mathbf{G}$. Therefore, the collection of row vectors of $\mathbf{G}$ lies in the span of $\mathbf{A}$. Finding $\mathbf{A}_k$ is equivalent to finding a collection of subspaces, each of dimension at most k, to cover $\mathbf{G}$. Client c_i is sent a collection of (at most) k rows of $\mathbf{A}_k$; these correspond to one subspace which covers $\mathbf{g}_i$.

Geometric Interpretation. The geometric interpretation of our problem is depicted in Figure 3. An index code $\mathbf{A}$ corresponds to a particular completion of the matrix $\mathbf{G}$. Therefore, the set of row vectors in $\mathbf{G}$ lies in the row span of $\mathbf{A}$ (which is of dimension T). We denote this subspace of dimension T by L. The problem of finding a matrix $\mathbf{A}_k$ can be interpreted as finding a set of subspaces, each of dimension at most k, such that each row vector $\mathbf{g}_i$, $i \in [n]$, is covered by at least one of these smaller subspaces. Once these subspaces are selected, then the rows of $\mathbf{A}_k$ are taken as the union of the basis vectors of all these subspaces. Client c_i is then given the basis vectors of subspace L_i, i.e., the one which covers $\mathbf{g}_i$, instead of the whole matrix $\mathbf{A}_k$. Therefore c_i would have perfect knowledge of L_i instead of L. Having less information about L naturally translates to less information about the requests of other clients, as we more formally discuss in the next section.

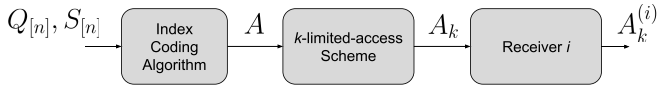


Fig. 4. The procedure of designing an index code and applying k-limited-access schemes.

III. Achieved Privacy Levels

In this section, we investigate and quantify the level of privacy that k-limited-access schemes can achieve compared to a conventional index coding scheme (i.e., when each client has access to the entire coding matrix). In particular, we are interested in understanding how a curious client can obtain information about the identity of the request of other clients. Towards this end, we quantify the amount of information that a curious client can obtain about the coding matrix $\mathbf{A}$ when a k-limited-access scheme is used. This approach stems from our recent observation in [1], where we showed that the knowledge of the coding matrix may enable a curious client to infer information about the requests and side information of other clients. In other words, quantifying the privacy leakage in the coding matrix offers a proxy to understanding the privacy leakage in the request and side information set. In what follows, we consider the setup described in the previous section and suppose that the last client, client c_n, is curious.

We assume that the index coding instance is random, i.e., we consider the requests and side information sets of clients as random variables and denote them as $Q_{[n]}$ and $S_{[n]}$, respectively. The operation of the server is shown in Figure 4 and is described as follows:

Step-1: The server obtains the information about the requests $Q_{[n]}$ and side information sets $S_{[n]}$ of all clients $c_{[n]}$.

Step-2: Based on this information, the server designs an index code $\mathbf{A}$ by means of some index coding algorithm [9]–[11]. In particular, the index coding algorithm used to design this index code is only known by the server and not by the clients.

Step-3: The server then applies the k-limited-access scheme to obtain $\mathbf{A}_k = \mathbf{P}\mathbf{A}$, where $\mathbf{P}$ is a deterministic mapping from $\mathbf{A}$ to $\mathbf{A}_k$ (see Section IV for the construction of $\mathbf{P}$). This implies that T_k is a deterministic function of T and k (i.e., the parameter of the scheme).

Step-4: The server sends $\mathbf{A}_k^{(i)}$ to client c_i. If multiple $\mathbf{A}_k^{(i)}$ can be selected, then the server picks and transmits one such matrix uniformly at random, independently of the underlying $\mathbf{A}$ which might have generated this $\mathbf{A}_k$.

We are now interested in quantifying the level of privacy (measured in terms of the amount of information about $\mathbf{A}$ learnt by the curious client c_n leveraging the knowledge of $\mathbf{A}_k^{(n)}$) that is achieved by the protocol described above. Towards this end, we use two privacy metrics, namely an *entropy-based* metric and the *Maximal Information Leakage* (MIL).

A. Entropy-Based Privacy Metric

The entropy-based privacy metric is inspired by the geometric interpretation of our problem in Figure 3. We let L (respectively, L_n) be the random variable associated with the subspace spanned by the T rows of the coding matrix $\mathbf{A}$ (respectively, spanned by the k row vectors of $\mathbf{A}_k^{(n)}$). Client c_n receives the matrix $\mathbf{Y}_k$ and as such she knows T_k. Given this, we now define the entropy-based privacy metric and evaluate it for the proposed protocol.

Definition III.1. The entropy-based privacy metric is defined as

$$P_k^{(\text{Ent})} = H\left(L|L_n, T_k\right),$$

and quantifies the amount of uncertainty that c_n has about the subspace spanned by the T rows of the index coding matrix $\mathbf{A}$.

Before characterizing $P_k^{(\text{Ent})}$, we state the following lemma, which is a special case of [12, Theorem 2]. We provide the proof for this special case in Appendix A for completeness.

Lemma III.1. *Given a subspace $L_n \subseteq \mathbb{F}_2^m$ of dimension k, let $\mathcal{L}(T, L_n)$ be the set of subspaces $L \subseteq \mathbb{F}_2^m$ of dimension $T \geq k$ where $L_n \subseteq L$. Then $|\mathcal{L}(T, L_n)|$ is equal to*

$$|\mathcal{L}(T, L_n)| = \prod_{\ell=0}^{T-k-1} \frac{2^m - 2^{k+\ell}}{2^T - 2^{k+\ell}}.$$

Assume an index coding setting with c_n observing a particular subspace $L_n = \ell_n$ and a number of transmissions $T_k = t_k$ for the k-limited access scheme. Moreover, we consider a stronger adversary (i.e., curious client) and assume that she also knows the specific realization of $T = t$. Given this, we can compute

$$\begin{aligned} P_k^{(\text{Ent})} &= H\left(L|L_n = \ell_n, T_k = t_k, T = t\right) \\ &\overset{(\text{a})}{=} H\left(L|L_n = \ell_n, T = t\right) \overset{(\text{b})}{=} \log\left(|\mathcal{L}(t, \ell_n)|\right) \\ &\overset{(\text{c})}{=} \log\left(\prod_{\ell=0}^{t-k-1} \frac{2^m - 2^{k+\ell}}{2^t - 2^{k+\ell}}\right) \overset{m \gg t}{\approx} m(t-k), \end{aligned} \quad (1)$$

where: (i) the equality in (a) follows because T_k is a deterministic function of T and k, which is the parameter of the scheme (see Step-3); (ii) the equality in (b) follows by assuming that the underlying system maintains a uniform distribution across all feasible t-dimensional subspaces of $\mathbb{F}_2^m$; (iii) the equality in (c) follows by virtue of Lemma III.1. We note that when $m \gg t$, then the quantity in (1) decreases linearly with k, i.e., as intuitively expected, the less rows of the coding matrix c_n learns, the less she can infer about the subspace spanned by the T rows of the coding matrix $\mathbf{A}$. This suggests that, by increasing k, c_n has less uncertainty about q_i. Note also that $P_k^{(\text{Ent})}$ is zero when $k = t$; this is because, under this condition, c_n receives the entire index coding matrix, i.e., $L_n = L$, and hence she is able to perfectly reconstruct the subspace spanned by its rows. However, although $P_k^{(\text{Ent})} = 0$ when $k = t$, c_n might still have uncertainty about q_i [1]. Quantifying this uncertainty is an interesting open problem; this uncertainty, in fact, depends on the underlying system, e.g., on the index code used by the server and on the distribution with which the index code matrix is selected.

B. Maximal Information Leakage (MIL)

The second metric that we consider as our privacy metric is the MIL [13], [14]. Given two discrete random variables X and Y with alphabets $\mathcal{X}$ and $\mathcal{Y}$, the MIL from X to Y is denoted by $\mathcal{L}(X \to Y)$ and defined as

$$\mathcal{L}(X \to Y) = \sup_{U-X-Y-\hat{U}} \log \frac{\Pr\left[U = \hat{U}\right]}{\max_{u\in\mathcal{U}} p_U(u)} \tag{2}$$

$$= \log \sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}: p_X(x)>0} p_{Y|X}(y|x), \tag{3}$$

where equality in (3) is shown in [13, Theorem 1]. The MIL metric captures the amount of information leaked about X through Y to an adversary, who is interested in estimating a (possibly probabilistic) function U of X. The adversary's estimate is $\hat{U}(Y)$ which is a function of Y. This is captured by the fact that $U - X - Y - \hat{U}$ forms a Markov chain as shown in the expression in (2). The MIL computes the multiplicative gain in the guessing power of the adversary, $\Pr(\hat{U} = U)$, in comparison to the best uninformed guess, $\max_{u\in\mathcal{U}} p_U(u)$. Moreover, the metric considers a worst-case such adversary, that is, an adversary who is interested in computing an estimate $\hat{U}(Y)$ of a function $U(X)$ for which the maximum information can be leaked out of Y; thus the initial supremum step. This definition admits an operational interpretation to privacy [14]: if $\mathcal{L}(X \to Y) = \ell$ bits, then the guessing power of the adversary for any function U is upper-bounded by $\Pr(\hat{U} = U) \leq 2^{\ell} \max_{u\in\mathcal{U}} p_U(u)$. We kindly refer the Reader to [13] for more details about this metric.

The result in [13] shows that this quantity depends only on the joint distribution of X and Y. The following properties of the MIL are useful [13, Corollary 2]:

- (Property 1): If $X - Y - Z$, then $\mathcal{L}(X \to Z) \leq \min\{\mathcal{L}(X \to Y), \mathcal{L}(Y \to Z)\}$,
- (Property 2): $\mathcal{L}(X \to Y) \leq \min\{\log|\mathcal{X}|, \log|\mathcal{Y}|\}$,
- (Property 3): $\mathcal{L}(X \to X) = \log|\{x : p_X(x) > 0\}|$.

To describe how we use the MIL as a privacy metric in our setup, we first need to define what are the corresponding random variables X and Y, and then argue that the estimation of client c_n of the requests of other clients forms a Markov chain as required by the MIL definition. To do so, we first define the following sets:

1) Given $\mathbf{g}_i$, $\mathbf{A}_k$ and an integer r, let $\mathcal{P}(\mathbf{g}_i, \mathbf{A}_k, r)$ be the set of all possible sub-matrices $\mathbf{A}_k^{(i)}$ of $\mathbf{A}_k$ with *exactly* r rows, that client c_i can use to reconstruct the vector $\mathbf{g}_i$:

$$\mathcal{P}(\mathbf{g}_i, \mathbf{A}_k, r) = \{\mathbf{Z} \subset_r \mathbf{A}_k \mid \exists \mathbf{d} \in \mathbb{F}_2^r \text{ s.t. } \mathbf{g}_i = \mathbf{dZ}\},$$

2) Given q_i, $\mathcal{S}_i$ and $\mathbf{A}_k$, let $\mathcal{T}(q_i, \mathcal{S}_i, \mathbf{A}_k)$ be the set of all possible sub-matrices $\mathbf{A}_k^{(i)}$ of $\mathbf{A}_k$ with the minimum possible number of rows, such that client c_i with side information $\mathcal{S}_i$ can decode q_i:

$$\mathcal{T}(q_i, \mathcal{S}_i, \mathbf{A}_k) = \bigcup_{\mathbf{g}_i\in\mathcal{G}(q_i,\mathcal{S}_i)} \mathcal{P}(\mathbf{g}_i, \mathbf{A}_k, r_{\min}),$$

where

$$\mathcal{G}(q_i, \mathcal{S}_i) = \left\{\mathbf{g} \in \mathbb{F}_2^m \mid g_{q_i} = 1, g_{[m]\setminus\{q_i\cup\mathcal{S}_i\}} = 0\right\},$$

and $r_{\min} = \min\mathcal{R}$, $\mathcal{R} = \{r \in \mathbb{N}^+ : \exists \mathbf{g}_i \in \mathcal{G}(q_i, \mathcal{S}_i) \text{ such that } \mathcal{P}(\mathbf{g}_i, \mathbf{A}_k, r) \neq \emptyset\}$.

Since the requests and the side information sets are considered as random variables, then all subsequently generated codes, namely $\mathbf{A}$, $\mathbf{A}_k$ and $\mathbf{A}_k^{(i)}$ can be treated as random variables as well. We denote the corresponding random variables of these quantities as A, A_k and $A_k^{(i)}$ respectively. In other words, for a given realization of $Q_{[n]} = q_{[n]}$ and $S_{[n]} = \mathcal{S}_{[n]}$, the corresponding realizations of the aforementioned codes used by the server are $A = \mathbf{A}$, $A_k = \mathbf{A}_k$ and $A_k^{(i)} = \mathbf{A}_k^{(i)}$.

When using conventional index codes (i.e., without k-limited-access schemes), client c_n (i.e., the curious client and hence the adversary) would try to infer information about $Q_{[n-1]}$ from observing A and given her information of Q_n, S_n. Therefore, one can think of client c_n's estimate of $Q_{[n-1]}$ as being a particular estimation function, the input of which is A. Differently, after using k-limited-access schemes, client c_n would only have observed $A_k^{(n)}$ instead of A. Therefore, in the context of MIL, one choice of the variables X and Y is A and $A_k^{(n)}$, respectively. The function U would therefore be client c_n's estimate of $Q_{[n-1]}$ out of A. The following proposition shows that this choice of variables X, Y and U allows us to use the MIL as a metric.

Proposition III.2. *The following Markov chain holds*

$$Q_{[n-1]} - A - A_k - A_k^{(n)}, \tag{4}$$

conditioned on the knowledge of Q_n, S_n in every stage of the chain.

Proof: We have the following:

- $Q_{[n-1]} - A - A_k$ holds since A_k is a deterministic function of A (see also Step-3 of the proposed protocol);
- $A - A_k - A_k^{(n)}$ holds since $p(A_k^{(n)}|A_k, Q_n, S_n) = 1/|\mathcal{T}(Q_n, S_n, A_k)|$, independent of A, as described in Step-4 of the proposed protocol.

■

We define $P_k^{(\text{MIL})} = \mathcal{L}\left(A \to A_k^{(n)} | Q_n = q_n, S_n = \mathcal{S}_n\right)$ as our MIL privacy metric.[3] The quantity $P_k^{(\text{MIL})}$ gives the maximum amount of information that c_n can extract about $Q_{[n-1]}$ given the knowledge of Q_n and S_n. The following theorem – proved in Appendix B – provides a guarantee on $P_k^{(\text{MIL})}$.

Theorem III.3. *Using the MIL, the attained level of privacy against a curious client when k-limited-access schemes are used is*

$$P_k^{(MIL)} = O(|\mathcal{S}_n| + mk). \tag{5}$$

The quantity in (5) characterizes the maximum amount of information that can be leaked to a curious client when k-limited-access schemes are used. It is clear that decreasing k would decrease this amount of information; this aligns with the intuition that the less rows a server gives to a client, the less information a client would be able to infer about other clients sharing the broadcast domain. In order to shed

[3] We use the notation $\mathcal{L}(X \to Y|Z)$ to denote that the variables X and Y are conditioned on Z.

more light on the benefits of using k-limited-access schemes, one could compare the quantity $P_k^{(\text{MIL})}$ with the MIL obtained when k-limited-access schemes are not used, i.e., when a client observes the whole matrix A. Let this quantity be denoted as $\bar{P}_k^{(\text{MIL})} = \mathcal{L}(A \to A|Q_n = q_n, S_n = \mathcal{S}_n)$. Our target is to provide a lower bound on $\bar{P}_k^{(\text{MIL})}$. This would indeed offer a best-case estimate on the amount of information leaked when conventional codes are used. We have the following result, which is proved in Appendix C.

Theorem III.4. *Using the MIL, the attained level of privacy against a curious client for a conventional index coding setup is*

$$\Omega\left(mT - T^2\right) \leq \bar{P}_k^{(MIL)} \leq O(|\mathcal{S}_n| + mT). \tag{6}$$

Discussion. The results in Theorem III.3 and Theorem III.4 shed light on the guessing power of the adversary when using k-limited-access schemes. In particular,

- The upper bounds in (5) and (6) can be restated as follows

$$\Pr\left[\hat{Q}_i(A_k^{(n)}) = Q_i | Q_n = q_n, S_n = \mathcal{S}_n\right] \leq 2^{|\mathcal{S}_n|+mk} \cdot \Pr\left[\hat{Q}_i = Q_i | Q_n = q_n, S_n = \mathcal{S}_n\right], \tag{7}$$

$$\Pr\left[\hat{Q}_i(A) = Q_i | Q_n = q_n, S_n = \mathcal{S}_n\right] \leq 2^{|\mathcal{S}_n|+mT} \cdot \Pr\left[\hat{Q}_i = Q_i | Q_n = q_n, S_n = \mathcal{S}_n\right]. \tag{8}$$

  Note that the term $2^{|\mathcal{S}_n|}$ is present in both upper bounds. This term suggests that the guessing power of an adversary (i.e., curious client) increases with the size of the side information set. This can be explained as follows. A client with a small side information set is naturally limited in the choices of an index coding solution (conventional or k-limited-access schemes) that can satisfy her. In contrast, a client with a large side information set can have a large set of possible index coding solutions that can satisfy her. Therefore, when one specific index code (conventional or not) is revealed to her, this can potentially reveal more information about other clients for the later case (i.e., large side information set) than the former one (i.e., small side information set). Therefore, this term corresponds to natural guessing capabilities that the adversary has which depend on its side information set. The second term is affected by the choice of the code: the bound in (8) suggests that the guessing power can be increased depending on the value of T (the number of transmissions), whereas the bound in (7) limits the guessing power growth of the adversary to k, regardless of the number of transmissions.
- The aforementioned arguments compare the growth rates of the *upper bounds* on MIL for both schemes. In addition, we can make the assumption that Q_i and $\hat{Q}_i(A)$ correspond to the pair of functions which satisfy the supremum step in computing the MIL for conventional codes.[4] Under this assumption, we can show that

$$\frac{\Pr\left[\hat{Q}_i(A_n^{(k)}) = Q_i\right]}{\Pr\left[\hat{Q}_i(A) = Q_i\right]} \overset{(a)}{\leq} \frac{2^{P_k^{(\text{MIL})}}}{2^{\bar{P}_k^{(\text{MIL})}}} \overset{(b)}{\leq} 2^{|\mathcal{S}_n|} \cdot \frac{2^{mk+T^2}}{2^{mT}}, \tag{9}$$

  where (a) follows by multiplying the numerator and denominator by $\Pr\left[\hat{Q}_i = Q_i\right]$ and then using the definition of MIL in (2) and the assumption that $\hat{Q}_i(A)$ and Q_i satisfy the supremum step, and (b) follows by using the upper and lower bounds in (5) and (6), respectively. The bound in (9) gives an upper bound on the gain in the guessing power of the adversary when k-limited-access schemes are used with respect to conventional schemes. The term $2^{|\mathcal{S}_n|}$ is independent from the used scheme. However, the extra gain term vanishes as T grows when $k = o(T)$ and $T = o(\sqrt{m})$; in other words, in this regime of parameters, the probability of guessing the right request approaches zero. This result can be interpreted with the help of Figure 5. The k-limited-access schemes always achieve privacy gains as compared to conventional index codes, when the upper bound in (5) and the lower bound in (6) strictly mismatch. A sufficient (but not necessary) condition for this is to select $k = o(T)$ and $T = o(\sqrt{m})$.

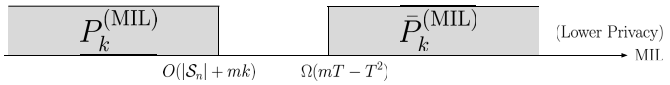


Fig. 5. This figure shows how the MIL privacy metrics compare for the conventional index coding schemes and the k-limited-access schemes. Taking $k = o(T)$ and $T = o(\sqrt{m})$ would guarantee privacy gains when using k-limited-access schemes.

C. Metrics Comparison

The two proposed metrics in this manuscript provide two *different* interpretations of the adversary's capabilities. In particular, while the entropy-based metric provides a *geometric* interpretation, the MIL offers a *computational* interpretation. Although the values of the two metrics measure in bits (assuming that logarithms in both metrics are in base 2), the two metrics are not directly comparable. The main reason is that the two metrics have opposite indications of privacy for k-limited-access schemes: a high value of P_k^{MIL} indicates less privacy, while a high value of P_k^{Ent} indicates more privacy. In an attempt to provide a comparison of the two metrics, note that $P_k^{\text{MIL}} \geq 0$ and that $P_k^{\text{Ent}} \leq \log|\mathcal{L}(t, \ell_n)|$. Therefore, in order to make a consistent comparison, we instead consider the two metrics P_k^{MIL} and $\bar{P}_k^{\text{Ent}} = \log|\mathcal{L}(t, \ell_n)| - P_k^{\text{Ent}}$.

In the remaining part of this section, we prove that neither of the two metrics gives more pessimistic values than the other. Towards this end, we show two index coding instances, where in one we have $P_k^{\text{MIL}} < \bar{P}_k^{\text{Ent}}$, and vice versa in the other case. We show these cases with the help of the next lemma (the proof of which is in Appendix D), where we let the index coding strategy adopted by the server be described by the distribution $p(A|Q_{[n]}, S_{[n]})$: the distribution of picking

[4]This assumption is suggested when studying the MIL in applications when the system designer does not have access to the distribution f_{UX} and $f_{Y\hat{U}}$ [14].

a conventional index code A given the index coding instance $Q_{[n]}$ and $S_{[n]}$.

Lemma III.5. *For a uniform server strategy, i.e., $p(A|Q_{[n]}, S_{[n]})$ is uniform over all values of A which could satisfy the particular index coding instance described by $Q_{[n]}, S_{[n]}$, then we have $P_k^{MIL} = \log \left| A_k^{(n)} | Q_n, S_n \right|$.*

Using Lemma III.5, we can now show the following two cases. For simplicity, we consider the specific case of $m = 4$ and $T = 2$. We assume also that $Q_n = 1$ and $S_n = \emptyset$.

Case 1: $k = T$ and the server uses a uniform strategy. In this case, we have

$$P_k^{\text{MIL}} \overset{(a)}{=} \log \left| A_k^{(n)} | Q_n = 1, S_n = \emptyset \right| \overset{(b)}{=} \log |A| Q_n = 1, S_n = \emptyset| \overset{(c)}{=} \log(14 + 7),$$

where: (i) the equality in (a) follows using Lemma III.5; (ii) the equality in (b) follows by noting that $k = T$ corresponds to the conventional transmission; (iii) the equality in (c) follows by noting that there are 14 valid A matrices constructed as follows: one row vector is of the form $[1,\ 0,\ 0,\ 0]$ and the other row vector is any other non-zero vector, and there are 7 valid A matrices where the sum of the two row vectors add up to the vector $[1,\ 0,\ 0,\ 0]$. On the other hand, we have

$$\begin{aligned} \bar{P}_k^{\text{Ent}} &= \log |\mathcal{L}(t, \ell_n)| - P_k^{\text{Ent}} \\ &= \log |\mathcal{L}(t, \ell_n)| - H(L|L_n = l_n, T = t) \\ &\overset{(a)}{=} \log |\mathcal{L}(t, \ell_n)| - H(L|L = l_n, T = t) \\ &= \log |\mathcal{L}(t, \ell_n)| \overset{(b)}{=} \log \prod_{\ell=0}^{2-2-1} \left(\frac{2^4 - 2^{1+\ell}}{2^2 - 2^{1+\ell}} \right) = \log 1 = 0, \end{aligned}$$

where: (i) the equality in (a) follows by noting that $k = T$ corresponds to the conventional transmission; (ii) the equality in (b) follows from Lemma III.1. Therefore, we have $P_k^{\text{MIL}} > \bar{P}_k^{\text{Ent}}$.

Case 2: $k = 1$ and the server does not necessarily use a uniform strategy. In this case, we have

$$P_k^{\text{MIL}} \overset{(a)}{\leq} \log \left| A_k^{(n)} | Q_n = 1, S_n = \emptyset \right| \overset{(b)}{=} \log 1 = 0,$$

where: (i) the inequality in (a) follows by noting property 2 of the MIL; (ii) the equality in (b) follows by noting that the way client n can reconstruct the required message using $k = 1$ transmission is by receiving it uncoded. On the other hand, we have

$$\begin{aligned} \bar{P}_k^{\text{Ent}} &= \log |\mathcal{L}(t, \ell_n)| - P_k^{\text{Ent}} \\ &= \log |\mathcal{L}(t, \ell_n)| - H(L|L_n = [1,\ 0,\ 0,\ 0], T = t) \overset{(a)}{>} 0, \end{aligned}$$

where the inequality in (a) follows by using a transmission strategy that slightly deviates from the uniform distribution. Therefore, we have $P_k^{\text{MIL}} < \bar{P}_k^{\text{Ent}}$.

IV. Construction of k-Limited-Access Schemes

In this section, we focus on designing k-limited-access schemes and assessing their theoretical performance in terms of number of additional transmissions required with respect to a conventional index coding scheme. Recall that we are given a coding matrix $\mathbf{A}$ that requires T transmissions. Then, we seek to construct a matrix $\mathbf{P} \in \mathbb{F}_2^{T_k \times T}$, so that $\mathbf{A}_k = \mathbf{PA}$, and each client needs to access at most k rows of $\mathbf{A}_k$ to decode her requested message. In particular, we aim at constructing matrices $\mathbf{P}$ with T_k as small as possible. Trivially, $T_k \geq T$. Towards this end, we first derive upper and lower bounds on T_k. Our main result is stated in the theorem below.

Theorem IV.1. *Given an index coding matrix $\mathbf{A} \in \mathbb{F}_2^{T \times m}$ with $T \geq 2$, it is possible to transform it into $\mathbf{A}_k = \mathbf{PA}$ with $\mathbf{P} \in \mathbb{F}_2^{T_k \times T}$, such that each client can decode her requested message by combining at most k rows of $\mathbf{A}_k$, if and only if*

$$T_k \geq \max\{T, T^\star\}, \quad T^\star = \min \left\{ T_k : \sum_{i=1}^{k} \binom{T_k}{i} \geq n \right\}, \tag{10}$$

where n is the number of distinct rows of the matrix $\mathbf{G}$. Moreover, we provide polynomial time (in n) constructions of $\mathbf{P}$ such that:

- *When $\lceil T/2 \rceil \leq k < T$, then*

$$T_k \leq \min\{n, T + 1\}; \tag{11}$$

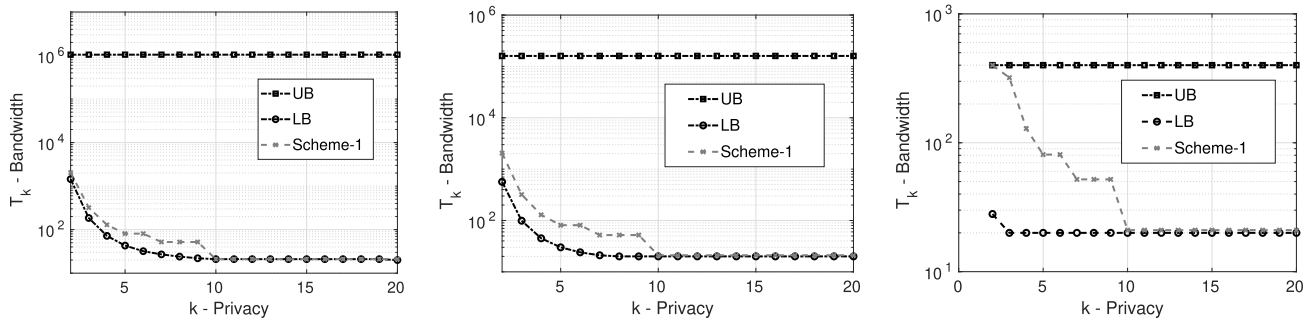
- *When $1 \leq k < \lceil T/2 \rceil$, then*

$$T_k \leq \min \left\{ n, k 2^{\lceil \frac{T}{k} \rceil} \right\}. \tag{12}$$

Proof: The lower bound on T_k in (10) is proved in Appendix E. In particular, the bound in (10) says that, if we are allowed to combine at most k out of the T_k vectors, then we should be able to create a sufficient number of vectors. The two upper bounds on T_k in (11) and (12) are proved in Section IV-A, where we give explicit constructions for $\mathbf{P}$. ■

We note that, as expected, the smaller the value of k that we require, the larger the value of T_k that we need to use. Trivially, for $k = 1$ we would need $T_k = n$, i.e., the server would need to send uncoded transmissions. Thus, there is a trade-off between the bandwidth – measured as the number T_k of broadcast transmissions – and privacy – captured by the value of k that we require. Interestingly, with just one extra transmission, i.e., $T_k = T + 1$, we can restrict the access of each client to at most half of the coding matrix, independently of the coding matrix $\mathbf{A}$ (i.e., $k = \lceil T/2 \rceil$). In other words, for this regime, we can achieve a certain level of privacy with minimal overhead. However, as we further reduce the value of k, the overhead becomes more significant. Moreover, the results in Theorem IV.1 also imply that our constructions are order-optimal in the case of sufficiently large values of n (i.e., when $n = \Theta(2^T)$).[5] In addition, when $\lceil T/2 \rceil \leq k < T$, our scheme is at most one transmission away from the optimal number of transmissions, and this holds for *any* value of n.

[5] Note that n is always $O(2^T)$ (i.e., the number of distinct non-zero vectors $\mathbf{g}_i$ for a given T is at most $2^T - 1$). The case of sufficiently large values of n corresponds to the case where this bound on the number of distinct vectors $\mathbf{g}_i$ is not loose: there is a corresponding lower bound on n, i.e., $n = \Omega(2^T)$. Therefore, the case of sufficiently large values of n corresponds to $n = \Theta(2^T)$.



(a) $n = 2^T - 1$ (b) $n = T^4$ (c) $n = T^2$

Fig. 6. Bandwidth (T_k on the y-axis) versus privacy (k on the x-axis) trade-off when using the k-limited-access schemes in Theorem IV.1 for different values of n. The plots in this figure are for $T = 20$.

This is shown in the following lemma, which is proved in Appendix E.

Lemma IV.2. *Consider an index coding setup. We have*

- *When $n = 2^T - 1$ and $\lceil T/2 \rceil \leq k < T$, the bounds in* (10) *and* (11) *coincide, i.e., the provided construction of $\mathbf{P}$ is optimal;*
- *For any value of $n < 2^T - 1$ and $\lceil T/2 \rceil \leq k < T$, the bound in* (11) *is at most one transmission away from the bound in* (10)*;*
- *When $n = \Theta(2^T)$ and for any value of k, then $T_k = \Theta(k2^{\frac{T}{k}})$, i.e., the provided construction is order-optimal.*

Figure 6 shows the trade-off exhibited by our proposed k-limited-access schemes between bandwidth usage (T_k) and the attained privacy (k) - we use k as a proxy to the amount of attained privacy against a curious client (see Section III). The figure shows the performance of our constructions in Theorem IV.1 (labeled as Scheme-1), as well as the lower bound in (10) (labeled as LB) and an upper bound which corresponds to uncoded transmissions (labeled as UB). Figure 6(a) confirms the order-optimality of our constructions when $n = 2^T - 1$. In addition, our schemes perform similarly well when n is sufficiently large (and not necessarily equal to $2^T - 1$) as shown in Figure 6(b) where $n = T^4$. Finally, Figure 6(c) shows the performance for a small value of n, i.e., $n = T^2$. The figure shows that our proposed constructions do can be improved when n and k are both small, a case which we study in more details in Section V.

We now conclude this section by giving explicit constructions of the $\mathbf{P}$ matrix and prove the two upper bounds on T_k in (11) and (12). Our design of $\mathbf{P}$ allows to reconstruct any of the $2^T - 1$ non-zero vectors of size T. As such our constructions are universal, in the sense that the matrix $\mathbf{P}$ that we construct does not depend on the specific index coding matrix $\mathbf{A}$.

A. Proof of Theorem IV.1, Equations (11) *and* (12)

Recall that $\mathbf{A}$ is full rank and that the i-th row of $\mathbf{G}$ can be expressed as $\mathbf{g}_i = \mathbf{d}_i\mathbf{A}$, where $\mathbf{d}_i \in \mathbb{F}_2^T$ is the coefficients row vector associated with $\mathbf{g}_i$. We next analyze two different cases/regimes, which depend on the value of k.

Case I: $\lceil T/2 \rceil \leq k < T$. When $n \geq T + 1$, let

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_T \\ \mathbf{1}_T \end{bmatrix}, \tag{13}$$

which results in a matrix $\mathbf{A}_k$ with $T_k = T + 1$, matching the bound in (11). We now show that each $\mathbf{g}_i = \mathbf{d}_i\mathbf{A}, i \in [n]$, can be reconstructed by combining up to k vectors of $\mathbf{A}_k$. Let $w(\mathbf{d}_i)$ be the Hamming weight of $\mathbf{d}_i$. If $w(\mathbf{d}_i) \leq \lceil T/2 \rceil$, then we can reconstruct $\mathbf{g}_i$ as $\mathbf{g}_i = [\mathbf{d}_i 0]\mathbf{A}_k$, which involves adding $w(\mathbf{d}_i) \leq \lceil T/2 \rceil \leq k$ rows of $\mathbf{A}_k$. Differently, if $w(\mathbf{d}_i) \geq \lceil T/2 \rceil + 1$, then we can reconstruct $\mathbf{g}_i$ as $\mathbf{g}_i = [\bar{\mathbf{d}}_i\, 1]\mathbf{A}_k$, where $\bar{\mathbf{d}}_i$ is the bitwise complement of $\mathbf{d}_i$. In this case, reconstructing $\mathbf{g}_i$ involves adding $T - w(\mathbf{d}_i) + 1 \leq \lfloor T/2 \rfloor \leq k$ rows of $\mathbf{A}_k$.

When $n < T + 1$, then it is sufficient to send n uncoded transmissions, where the i-th transmission satisfies $c_i, i \in [n]$. In this case c_i has access only to the i-th transmission, i.e., $k = 1$. This completes the proof of the upper bound in (11).

Example: We show how the scheme works via a small example, where $T = 4$ and $k = 2$. In this case, we have

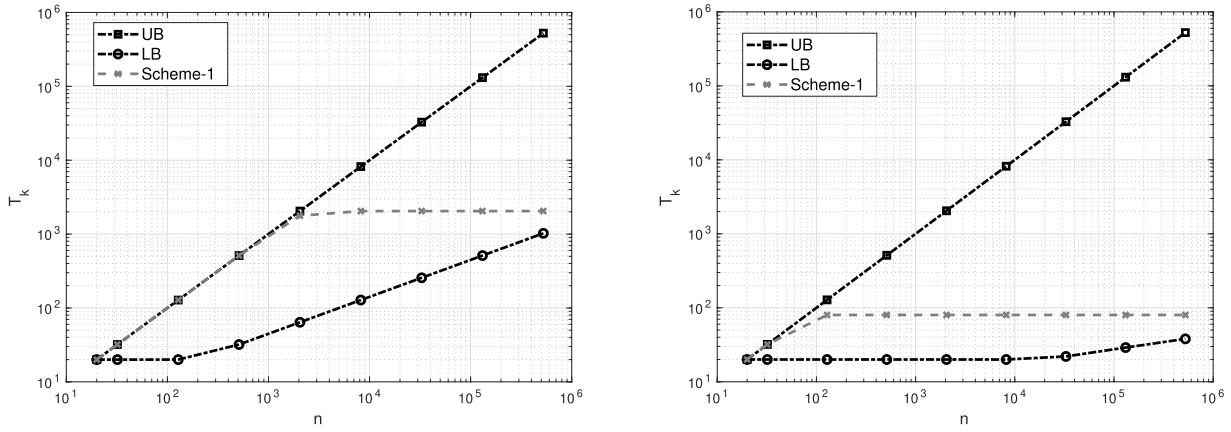
$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

If $\mathbf{g}_i = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}\mathbf{A}$, then it can be reconstructed as $\mathbf{g}_i = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \end{bmatrix}\mathbf{PA}$ with 2 rows of $\mathbf{PA}$ used in the reconstruction. Differently, if $\mathbf{g}_i = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}\mathbf{A}$, then it can be reconstructed as $\mathbf{g}_i = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \end{bmatrix}\mathbf{PA}$ with again 2 rows of $\mathbf{PA}$ used in the reconstruction.

Case II: $1 \leq k < \lceil T/2 \rceil$. Let $Q = \left\lfloor T/\left\lceil \frac{T}{k} \right\rceil \right\rfloor$ and $T_{\text{rem}} = T - Q\left\lceil \frac{T}{k} \right\rceil$. If k divides T, then $Q = k$, $T_{\text{rem}} = 0$, otherwise $Q \leq k - 1$ and $T_{\text{rem}} \leq \left\lceil \frac{T}{k} \right\rceil$. Then, we can write

$$\mathbf{P} = \begin{bmatrix} \bar{\mathbf{Z}}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{Z}}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{Z}}_Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \bar{\mathbf{Z}}_{Q+1} \end{bmatrix},$$

In other words, the matrix $\mathbf{P}$ is constructed as a block-diagonal matrix, with the diagonal elements being $\bar{\mathbf{Z}}_i$ for all $i \in [Q + 1]$, where $\bar{\mathbf{Z}}_i$, of dimension $\lambda_i \times \left\lceil \frac{T}{k} \right\rceil$, has as rows all



(a) $k = 2$ (b) $k = 5$

Fig. 7. Performance of the scheme in Theorem IV.1 (referred to as scheme-1) for different values of n, compared against the lower bound LB in equation (10) and the upper bound UB of sending uncoded transmissions - $T = 20$.

non-zero vectors of length $\lceil \frac{T}{k} \rceil$. Therefore, $\lambda_i = 2^{\lceil T/k \rceil} - 1$. Similarly, the matrix $\bar{\mathbf{Z}}_{Q+1}$, of dimension $\lambda_{Q+1} \times T_{\text{rem}}$, has as rows all non-zero vectors of length T_{rem}. Therefore, $\lambda_{Q+1} = 2^{T_{\text{rem}}} - 1$.

Therefore, equation (12) holds by computing

$$T_k = \sum_{i=1}^{Q+1} \lambda_i = Q\left(2^{\lceil \frac{T}{k} \rceil} - 1\right) + 2^{T_{\text{rem}}} - 1 \leq k 2^{\lceil \frac{T}{k} \rceil},$$

where we used the facts that $Q \leq k - 1$ and $T_{\text{rem}} \leq \lceil \frac{T}{k} \rceil$.

What remains is to show that any vector $\mathbf{g}_i, i \in [n]$, can be reconstructed by adding at most k vectors of $\mathbf{P}$. To show this, we prove that any vector $\mathbf{v} \in \mathbb{F}_2^T$ can indeed be constructed with the proposed design of $\mathbf{P}$. We note that we can express the vector $\mathbf{v}$ as $\mathbf{v} = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_{Q+1}]$, where $\mathbf{v}_i$'s, $i \in [Q]$ are parts of the vector $\mathbf{v}$ each of length $\lceil \frac{T}{k} \rceil$, while $\mathbf{v}_{Q+1}$ is the last part of $\mathbf{v}$ of length T_{rem}. Then, we can write $\mathbf{v} = \sum_{i \in \mathcal{K}(\mathbf{v})} \bar{\mathbf{v}}_i$, where $\bar{\mathbf{v}}_i = \begin{bmatrix} \mathbf{0}_{(i-1)\lceil \frac{T}{k} \rceil} & \mathbf{v}_i & \mathbf{0}_{(Q-i)\lceil \frac{T}{k} \rceil} & \mathbf{0}_{T_{\text{rem}}} \end{bmatrix}$ for $i \in [Q]$, $\bar{\mathbf{v}}_{Q+1} = \begin{bmatrix} \mathbf{0}_{Q\lceil \frac{T}{k} \rceil} & \mathbf{v}_{Q+1} \end{bmatrix}$ and $\mathcal{K}(\mathbf{v}) \subseteq [Q+1]$ is the set of indices for which $\mathbf{v}_i$ is not all-zero. According to the construction of $\mathbf{P}$, for all $i \in \mathcal{K}(\mathbf{v})$, the corresponding vector $\mathbf{v}_i$ is one of the rows in $\mathbf{Z}_i$. The proof concludes by noting that $|\mathcal{K}(\mathbf{v})| \leq k$. This is true because, if k does not divide T, then $Q \leq k - 1$; otherwise, $Q = k$ but $T_{\text{rem}} = 0$ (i.e., $\mathbf{v}_{Q+1}$ does not exist), therefore $\mathcal{K}(\mathbf{v}) \subseteq [k]$. This completes the proof of the upper bound in (12).

Example: We show how the scheme works via a small example, where $T = 8$ and $k = 3$. For this particular example, we have $Q = \lfloor T / \lceil \frac{T}{k} \rceil \rfloor = 2$ and $T_{\text{rem}} = T - Q\lceil \frac{T}{k} \rceil = 2$. Thus, the idea is that, to reconstruct a vector $\mathbf{v} \in \mathbb{F}_2^8$, we treat $\mathbf{v}$ as $Q+1 = 3$ disjoint parts; the first 2 are of length $\lceil \frac{T}{k} \rceil = 3$ and the remaining part is of length $T_{\text{rem}} = 2$. We then construct $\mathbf{P}$ as $Q+1 = 3$ disjoint sections, where each section allows us to reconstruct one part of the vector. Specifically, we construct

$$\mathbf{P} = \begin{bmatrix} \bar{\mathbf{Z}}_1 & \mathbf{0}_{7\times 3} & \mathbf{0}_{7\times 2} \\ \mathbf{0}_{7\times 3} & \bar{\mathbf{Z}}_2 & \mathbf{0}_{7\times 2} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \bar{\mathbf{Z}}_3 \end{bmatrix}$$

where

$$\bar{\mathbf{Z}}_1 = \bar{\mathbf{Z}}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \bar{\mathbf{Z}}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Any vector $\mathbf{v}$ can be reconstructed by picking at most k vectors out of $\mathbf{P}$, one from each section. For example, let $\mathbf{v} = [0\,1\,0\,0\,1\,1\,1\,0]$. This vector can be reconstructed by adding vectors number 2, 10 and 16 from $\mathbf{P}$.

V. Constructions for Small Values of n and k

In Section IV, we have proved that, independently of the value of n, if $k \geq \lceil T/2 \rceil$, then it is sufficient to add one additional transmission to the T transmissions of the conventional index coding scheme. Moreover, the analysis provided in Lemma IV.2 showed the order-optimality of our universal scheme in Theorem IV.1 (referred to as Scheme-1) for values of $k < \lceil T/2 \rceil$ when n is sufficiently large (i.e., exponential in T). Figure 7 shows the performance of Scheme-1 in Theorem IV.1 as a function of the values of n for $T = 20$, with $k = 2$ in Figure 7(a) and $k = 5$ in Figure 7(b). The performance of Scheme-1 was obtained by averaging over 1000 random index coding instances. In each instance, a code is constructed using the scheme described in Section IV-A, and only the rows actually used by the clients $c_{[n]}$ are retained. The performance of the scheme is finally computed by the average number of rows retained in those 1000 iterations. Figure 7 shows that our proposed scheme performs well not only for the case of sufficiently large n (i.e., $n = \Theta(2^T)$) but also for lower values of n. However, Figure 7 also suggests that for small values of both n and k (note the left-half of the plot in Figure 7(a)), we need to devise schemes that better adapt to the specific values of the index coding matrix $\mathbf{A}$ and vectors $\mathbf{g}_i, i \in [n]$ (recall that Scheme-1 is universal, and

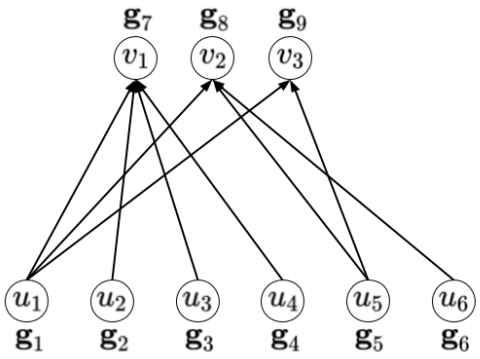


Fig. 8. Bipartite graph representation.

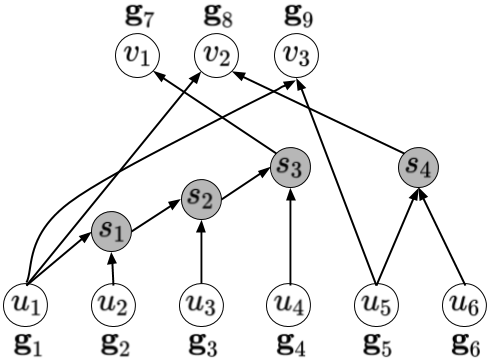


Fig. 9. Optimal representation when $k = 2$.

hence independent of $\mathbf{A}$). We next propose and analyze the performance of such algorithms.

A. Special Instances

We first represent the problem through a bipartite graph as follows. We assume that the rank of the matrix $\mathbf{G}$ is T. Then, there exists a set of T linearly independent vectors in $\mathbf{G}$; without loss of generality, we denote them as $\mathbf{g}_1$ to $\mathbf{g}_T$. Therefore, each vector $\mathbf{g}_{i+T}, i \in [n-T]$, can be expressed as a linear combination of some/all vectors from $\mathbf{g}_{[T]}$; we denote these vectors as the component vectors of $\mathbf{g}_{i+T}$. We can then represent the problem as a bipartite graph $(\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ with $|\mathcal{U}| = T$ and $|\mathcal{V}| = n - T$, where $u_i \in \mathcal{U}$ represents the vector $\mathbf{g}_i$ for $i \in [T]$, $v_j \in \mathcal{V}$ represents the vector $\mathbf{g}_{j+T}$ for $j \in [n-T]$, and an edge exists from node u_i to node v_j if $\mathbf{g}_i$ is one of the component vectors of $\mathbf{g}_{j+T}$. Figure 8 shows an example of such graph, where $n = 9$ and $T = 6$. For instance, v_1 (i.e., $\mathbf{g}_7$) can be reconstructed by adding $u_i, i \in [4]$ (i.e., $\mathbf{g}_i, i \in [4]$). Given a node s in the graph, we refer to the sets $\mathcal{O}_s$ and $\mathcal{I}_s$ as the *outbound* and *inbound* sets of s, respectively: the inbound set contains the nodes which have edges outgoing to node s, and the outbound set contains the nodes to which node s has outgoing edges (i.e., the nodes each of which has an incoming edge from s). Nodes on either sides of the bipartite graph have either inbound or outbound sets.

For instance, with reference to Figure 8, $\mathcal{O}_{u_1} = \{v_1, v_2, v_3\}$ and $\mathcal{I}_{v_1} = \{u_1, u_2, u_3, u_4\}$. For this particular example, there exists a scheme with $T_2 = 6$ which can reconstruct any vector with at most $k = 2$ additions. The matrix $\mathbf{A}_2$ which corresponds to this solution has the following vectors as rows: $\mathbf{g}_1$, $\mathbf{g}_1 + \mathbf{g}_2$, $\mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3$, $\mathbf{g}_1 + \mathbf{g}_2 + \mathbf{g}_3 + \mathbf{g}_4$, $\mathbf{g}_5$ and $\mathbf{g}_5 + \mathbf{g}_6$. It is not hard to see that each vector in $\mathbf{G}$ can be reconstructed by adding at most 2 vectors in $\mathbf{A}_2$. The row vectors in $\mathbf{A}_2$ that are not in $\mathbf{G}$ can be aptly represented as intermediate nodes on the previously described bipartite graph. These intermediate nodes are shown in Figure 9 as highlighted nodes. Each added node represents a new vector, which is the sum of the vectors associated to the nodes in its inbound set. We refer to the process of adding these intermediate nodes as creating a *branch*, which is defined next.

Definition V.1. Given an ordered set $\mathcal{S} = \{s_1, \cdots, s_S\}$ of nodes, where s_i precedes s_{i+1} for $i \in [S-1]$, a *branch on* $\mathcal{S}$ is a set $\mathcal{S}' = \{s'_1, \cdots, s'_{S-1}\}$ of $S-1$ intermediate nodes added to the graph with the following connections: node s'_1 has two incoming edges from s_1 and s_2, and for $i \in [S-1] \setminus \{1\}$, s'_i has two incoming edges from nodes s'_{i-1} and s_{i+1}.

For the example in Figure 9, we created branches on two ordered sets, $\mathcal{S}_1 = \{u_1, u_2, u_3, u_4\}$ and $\mathcal{S}_2 = \{u_5, u_6\}$. Once the branch is added, we can change the connections of the nodes in $\mathcal{V}$ in accordance to the added vectors. For the example in Figure 9, we can replace $u_{[4]}$ in $\mathcal{I}_{v_1}$ with only s_3. Using this representation, we have the following lemma.

Lemma V.1. *If $\mathcal{O}_{u_{i_T}} \subseteq \mathcal{O}_{u_{i_{T-1}}} \subseteq \cdots \subseteq \mathcal{O}_{u_{i_1}}$ for some permutation $i_1, \cdots, i_T$ of $[T]$, then this instance can be solved by exactly T transmissions for any $k \geq 2$.*

Proof: One solution of such instance would involve creating a branch on the set $\mathcal{S} = \{u_{i_1}, u_{i_2}, \cdots, u_{i_T}\}$. The scheme used would have the matrix $\mathbf{A}_2$ with its t-th row $\mathbf{a}_t = \sum_{\ell=1}^{t} \mathbf{g}_{i_\ell}$ for $t \in [T]$. Note that $\mathbf{g}_{i_1} = \mathbf{a}_1$ and $\mathbf{a}_t + \mathbf{a}_{t-1} = \mathbf{g}_{i_t}$ for all $t \in [T] \setminus \{1\}$. Moreover, for $j \in [n] \setminus [T]$, if $v_{j-T} \in \mathcal{O}_{u_{i_t}}$ for some i_t, then $v_{j-T} \in \mathcal{O}_{u_{i_\ell}}$ for all $\ell \leq t$. If we let t be the maximum index for which $v_{j-T} \in \mathcal{O}_{u_{i_t}}$, then we have $\mathcal{I}_{v_{j-T}} = \{u_{i_1}, \cdots, u_{i_t}\}$, and so we get $\mathbf{g}_j = \sum_{\ell=1}^{t} \mathbf{g}_{i_\ell} = \mathbf{a}_t$. This completes the proof. ■

Corollary V.2. *For $\mathbf{G} \in \mathbb{F}_2^{n \times T}$ of rank T, if $n = T + 1$, then this instance can be solved in T transmissions for any $k \geq 2$.*

Proof: Without loss of generality, let $\mathbf{g}_{[T]}$ be a set of linearly independent vectors of $\mathbf{G}$. Then, we have $\mathcal{O}_{u_i} = \{v_1\}$ for $i \in \mathcal{I}_{v_1}$ and $\mathcal{O}_{u_j} = \emptyset$ for $j \in [T] \setminus \mathcal{I}_{v_1}$. Thus, from Lemma V.1, this instance can be solved in T transmissions. This completes the proof. ■

B. Algorithms for General Instances

We here propose two different algorithms, namely Successive Circuit Removing (SCR) and Branch-Search, and analyze their performance.

Algorithm 1: Successive Circuit Removing (SCR). Our first proposed algorithm is based on Corollary V.2, which can be interpreted as follows: any matrix $\mathbf{G}$ of $r+1$ row vectors and rank r can be reconstructed by a corresponding $\mathbf{A}_2$ matrix with r rows. If there does not exist any subset of rows of $\mathbf{G}$ with

rank less than r, we call $\mathbf{G}$ *a circuit*.[6] Our algorithm works for the case $k = 2^q$, for some integer q. We first describe SCR for the case where $q = 1$, and then extend it to general values of q. The algorithm works as follows:

1) *Circuit Finding:* find a set of vectors of $\mathbf{G}$ that form a circuit of small size. Denote the size of this circuit as $r + 1$.
2) *Matrix Update:* apply Corollary V.2 to find a set of r vectors that can optimally reconstruct the circuit by adding at most $k = 2$ of them, and add this set to $\mathbf{A}_2$.
3) *Circuit Removing:* update $\mathbf{G}$ by removing the circuit. Repeat the first two steps until the matrix $\mathbf{G}$ is of size $T' \times T$ and of rank T', where $T' \leq T$. Then, add these vectors to $\mathbf{A}_2$.

Once SCR is executed, the output is a matrix $\mathbf{A}_2$ such that any vector in $\mathbf{G}$ can be reconstructed by adding at most $k = 2$ vectors of $\mathbf{A}_2$. Consider now the case where $q = 2$ (i.e., $k = 4$) for example. In this case, a second application of SCR on the matrix $\mathbf{A}_2$ would yield another matrix, denoted as $\mathbf{A}_4$, such that any row in $\mathbf{A}_2$ can be reconstructed by adding at most 2 vectors of $\mathbf{A}_4$. Therefore, any vector in $\mathbf{G}$ can now be reconstructed by adding at most 4 vectors of $\mathbf{A}_4$. We can therefore extrapolate this idea for a general q by successively applying SCR q times on $\mathbf{G}$ to obtain $\mathbf{A}_k$, with $k = 2^q$.

The following theorem gives a closed form characterization of the best and worst case performance of SCR.

Theorem V.3. *Let T_q^{SCR} be the number of vectors in $\mathbf{A}_k$ obtained via SCR. Then, for $k = 2^q$ and integer q, we have*

$$\underbrace{f^{Best}(f^{Best}(\cdots f^{Best}(n)))}_{q \text{ times}} \leq T_q^{SCR} \leq \underbrace{f^{Worst}(f^{Worst}(\cdots f^{Worst}(n)))}_{q \text{ times}}, \tag{14}$$

where $f^{Best}(n) = 2\left\lfloor \frac{n}{3} \right\rfloor$ and $f^{Worst}(n) = T\left(\left\lfloor \frac{n}{T+1} \right\rfloor + 1\right)$.

Proof: First we focus on the case $q = 1$. The lower bound in (14) corresponds to the best case when the matrix $\mathbf{G}$ can be partitioned into disjoint circuits of size 3. In this case, if SCR finds one such circuit in each iteration, then each circuit is replaced with 2 vectors in $\mathbf{A}_2$ according to Corollary V.2. To obtain the upper bound, note that any collection of $T+1$ has at most T independent vectors, and therefore contains a circuit of at most size $T+1$. Therefore, the upper bound corresponds to the case where the matrix $\mathbf{G}$ can be partitioned into circuits of size $T + 1$ and an extra T linearly independent vectors. In that case, the algorithm can go through each of these circuits, adding T vectors to $\mathbf{A}_2$ for each of these circuits, and then add the last T vectors in the last step of the algorithm. Finally, the bounds in (14) for a general q can be proven by a successive repetition of the above arguments. ■

Algorithm 2: Branch-Search. A naive approach to determining the optimal matrix $\mathbf{A}_k$ is to consider the whole space $\mathbb{F}_2^T$, loop over all possible subsets of vectors of $\mathbb{F}_2^T$ and, for every subset, check if it can be used as a matrix $\mathbf{A}_k$. The minimum-size subset which can be used as $\mathbf{A}_k$ is indeed the optimal matrix. However, such algorithm requires in the worst case $O\left(2^{2^T}\right)$ number of operations, which makes it prohibitively slow even for very small values of T. Instead, the heuristic that we here propose finds a matrix $\mathbf{A}_k$ more efficiently than the naive search scheme. The main idea behind the heuristic is based on providing a subset $\mathcal{R} \subset \mathbb{F}_2^T$ which is much smaller than 2^T and is guaranteed to have at least one solution. The heuristic then searches for a matrix $\mathbf{A}_k$ by looping over all possible subsets of $\mathcal{R}$. Our heuristic therefore consists of two sub-algorithms, namely Branch and Search. Branch takes as input $\mathbf{G}$, and produces as output a set of vectors $\mathcal{R}$ which contains at least one solution $\mathbf{A}_k$. The algorithm works as follows:

1) Find a set of T vectors of $\mathbf{G}$ that are linearly independent. Denote this set as $\mathcal{B}$.
2) Create a bipartite graph representation of $\mathbf{G}$ as discussed in Section V-A, using $\mathcal{B}$ as the independent vectors for $\mathcal{U}$.
3) Pick the dependent node v_i with the highest degree, and split ties arbitrarily. Denote by $\deg(v_i)$ the degree of node v_i.
4) Consider the inbound set $\mathcal{I}_{v_i}$, and sort its elements in a descending order according to their degrees. Without loss of generality, assume that this set of ordered independent nodes is $\mathcal{I}_{v_i} = \{u_1, u_2, \cdots, u_{\deg(v_i)}\}$.
5) Create a branch on $\mathcal{I}_{v_i}$. Denote the new branch nodes as $\{u_1^\star, u_2^\star, \cdots, u_{\deg(v_i)}^\star\}$.
6) Update the connections of all dependent nodes in accordance with the constructed branch. This is done as follows: for each node $v_j \in \mathcal{V}$ with $\deg(v_j) \geq k$, if $\mathcal{I}_{v_j} \cap \mathcal{I}_{v_i}$ is of the form $\{u_1, u_2, \cdots, u_\ell\}$ for some $\ell \leq \deg(v_i)$, then replace $\{u_1, u_2, \cdots, u_\ell\}$ in $\mathcal{I}_{v_j}$ with the single node $u_\ell^\star$. Do such replacement for the maximum possible value of ℓ.
7) Repeat 3) to 6) until all nodes in $\mathcal{V}$ have degree at most k.

The output $\mathcal{R}$ is the set of vectors corresponding to all nodes in the graph. The next theorem shows that $\mathcal{R}$ in fact contains one possible $\mathbf{A}_k$, and characterizes the performance of Branch.

Theorem V.4. *For a matrix $\mathbf{G}$ of dimension $n \times T$, (a) Branch produces a set $\mathcal{R}$ which contains at least one possible $\mathbf{A}_k$, (b) the worst-case time complexity t_{Branch} of Branch is $O(n^2)$, and (c) $|\mathcal{R}| \leq (n - T)T$.*

Proof: To see (a), note that the algorithm terminates when all dependent nodes have a degree of k or less. In every iteration of the algorithm, the degrees of all dependent nodes either remain the same or are reduced. In addition, at least one dependent node is updated and its degree is reduced to 1. Therefore the algorithm is guaranteed to terminate. Since all dependent nodes have degrees k or less, their corresponding vectors can be reconstructed by at most k vectors in $\mathcal{R}$. Therefore, $\mathcal{R}$ contains at least one solution $\mathbf{A}_k$.

To prove (b), the worst-case runtime of Branch corresponds to going over all nodes in $\mathcal{V}$ and creating a branch for each one. For the i-th node considered by Branch, the algorithm would update the dependencies of all dependent nodes with degrees greater than k, which are at most $n - i$ nodes. Therefore $t_{\text{Branch}} = \sum_{i=0}^{n-1} (n - i) = n(n-1) = O(n^2)$.

To prove (c), note that $|\mathcal{R}|$ is equal to the total number of nodes in all branches created by the algorithm. Therefore we can write $|\mathcal{R}| \leq \sum_{v_i \in \mathcal{V}} \deg(v_i) \leq (n - T)T = O(nT)$. ■

[6] This is in accordance to the definition of a circuit for a matroid [15].

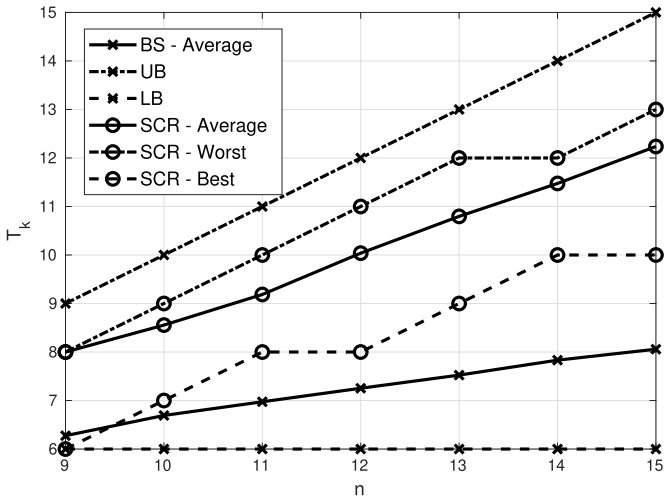


Fig. 10. Performance comparison for different schemes - $T = 6$, $k = 2$.

Let t_{Search} be the worst-time complexity of the Search step in Branch-Search. Then the worst-case time complexity of Branch-Search is equal to $t_{\text{BS}} = t_{\text{Branch}} + t_{\text{Search}} \leq O(n^2) + 2^{|\mathcal{R}|} = O(n^2) + O(2^{nT}) = O(2^{nT})$, which is exponentially better than the complexity of the naive search. Although our heuristic is still of exponential runtime complexity, we observe from numerical simulations that $|\mathcal{R}|$ is usually much less than $(n - T)T$. Finding more efficient ways of searching through the set $\mathcal{R}$ to find a solution $\mathbf{A}_k$ is an open question.

C. Numerical Evaluation

We here explore the performance of our proposed schemes through numerical evaluations. Specifically, we assess the performance in terms of T_k of SCR and Branch-Search (labeled as BS). We compare their performance against the lower bound in equation (10) (labeled as LB), and the upper bound of sending uncoded transmissions (labeled as UB). In particular, we are interested in regimes for which $k < \lceil T/2 \rceil$, because otherwise we know from Theorem IV.1 that $T_k = T + 1$. Moreover, we consider values of $n < 2^T - 1$, because if $n = 2^T - 1$ we know from Lemma IV.2 that Scheme-1 is order optimal. For SCR, we evaluate its average performance (averaged over 1000 iterations) as well as its upper and lower bounds performance established in Theorem V.3. For Branch-Search, we evaluate its average performance (averaged over 1000 iterations). Figure 10 shows the performance of all the aforementioned schemes for $T = 6$ and $k = 2$. As can be seen from Figure 10, SCR consistently performs better than uncoded transmissions. In addition, although the current implementation of SCR greedily searches for a small circuit to remove, more sophisticated algorithms for small circuit finding could potentially improve its performance. However, the bounds in (14) suggest that the performance of SCR is asymptotically $O(n)$. Branch-Search appears to perform better than other schemes in the average sense. Understanding its asymptotic behavior in the worst-case is an interesting open problem.

VI. Related Work

Index coding was introduced in [8], where the problem was proven to be NP-hard. Given this, several works have aimed at providing approximate algorithms for the index coding problem [9], [11], [16]. In our work, we were interested in studying the index coding problem from the perspective of private information delivery.

The problem of protecting privacy was initially proposed to enable the disclosure of databases for public access, while maintaining the anonymity of the clients [17]. Similar concerns have been raised in the context of *Private Information Retrieval* (PIR), which was introduced in [18] and has received a fair amount of attention [19]–[23]. In particular, in PIR the goal is to ensure that no information about the identity of clients' requests is revealed to a set of malicious databases when clients are trying to retrieve information from them. Similarly, the problem of *Oblivious Transfer* was studied [24], [25] to establish, by means of cryptographic techniques, two-way private connections between the clients and the server. We note that it is not clear how the use of cryptographic approaches would help in our setup. A curious client, in fact, obtains information about other clients once she learns the transmitted combinations of the messages, i.e., the coding operations. In other words, given that a curious client has also requested data, she needs to learn how the transmitted messages are coded, in order to be able to decode her own requested message.

We were here interested in addressing privacy concerns in broadcast domains. In particular, we analyzed this problem within the index coding framework, as we recently proposed in [1]. This problem differs from secure index coding [26], [27], where the goal is to guarantee that an external eavesdropper (with her own side information set) in [26], and each client in [27], does not learn any information about the *content* of the messages other than her requested message. Differently, our goal was to limit the information that a client can learn about the *identities* of the requests of other clients (however, the two approaches could be combined). Note that the techniques developed here can fundamentally differ from those designed for secure index coding. As an extreme example, in fact, the server in our setup can trivially send all the messages that it possesses in an uncoded manner on the broadcast channel. In this case, a curious client will be able to decode all messages, but would still not be able to infer which messages were requested/possessed by other clients, and would learn nothing about their side information. This property is what fundamentally contrasts the problem under consideration from the works in [26], [27]. Moreover, our approach here has a significant difference with respect to [1]. In fact, while in [1] our goal was to design the coding matrix to guarantee a high-level of privacy, here we assumed that an index coding matrix (that satisfies all clients) was given to us and we developed methods to increase its achieved level of privacy.

The use of k-limited-access schemes allows the server to transform an existing index code into a *locally decodable index code* [28], [29]. Locally decodable index codes allow each client to decode her request using at most k symbols

out of the codeword, where k is referred to as the locality of the code. In [28], the authors showed that the optimal scalar linear locally decodable index codes with locality 1 are the ones obtained from the coloring of the information graph of the index coding problem. In addition, they provided probabilistic results on the existence (and the impossibility of existence) of locally decodable codes with particular lengths and localities for index coding problems on random graphs. In [29], the authors extended one result in [28] where they showed that the optimal *vector* linear locally decodable index codes with locality 1 are obtained from the fractional coloring of the information graph. In addition, they provided a scheme which allows the construction of locally decodable codes for a particular set of index coding instances with special properties, i.e., when certain covering properties are maintained on the side information graph of the index coding problem. Differently from these works, one of the main results of this paper consisted of providing deterministic constructions/schemes which transform any existing index code into an equivalent code with locality k. In addition, our schemes are universal, i.e., they do not depend on the underlying index coding instance.

The solution that we here proposed to limit the privacy leakage is based on finding overcomplete bases. This approach is closely related to compressed sensing and dictionary learning [30], where the goal is to learn a dictionary of signals such that other signals can be *sparsely* and *accurately* represented using atoms from this dictionary. These problems seek lossy solutions, i.e., signal reconstruction is not necessarily perfect. This allows a convex optimization formulation of the problem, which can be solved efficiently [31]. In contrast, our problem was concerned with lossless reconstructions, in which case the optimization problem is no longer convex.

VII. Conclusion

In this paper, we studied privacy risks in index coding. This problem is motivated by the observation that, since the coding matrix needs to be available to all clients, then some clients may be able to infer the identity of the request and side information of other clients. We proposed the use of k-limited-access schemes: these schemes transform the coding matrix so that we can restrict each client to access at most k rows of the transformed matrix as opposed to the whole of it. We explored two privacy metrics, one based on entropy arguments, and the other on the maximal information leakage. Both metrics indicate that the amount of privacy increases with the number of rows that we hide. We then designed polynomial time universal k-limited-access schemes, that do not depend on the structure of the index coding matrix $\mathbf{A}$ and proved that they are order-optimal for some parameter regimes. For the remaining regimes, we proposed algorithms that depend on the structure of the index coding matrix $\mathbf{A}$ and provide improved performance. We overall found that there exists an inherent trade-off between privacy and bandwidth (number of broadcast transmissions), and that in some cases we can achieve significant privacy with minimal overhead.

Acknowledgment

Most of the work in this paper was done when M. Karmoose, L. Song, and M. Cardone were at UCLA.

Appendix A
Proof of Lemma III.1

The proof is based on simple counting arguments. A subspace L contains all vectors in L_n, the number of which is 2^k. A subspace L therefore consists of a set of $T-k$ linearly independent vectors $\{v_1, \cdots v_{T-k}\}$ that are in $\mathbb{F}_2^m \backslash L_n$, and all linear combinations of $\{v_{[T-k]}\}$ and vectors in L_n. We now enumerate the number of ways such a subspace L, with $L_n \subseteq L$, can be constructed. We first pick a vector $v_1 \in \mathbb{F}_2^m \setminus L_n$. The total number of possible choices for v_1 is equal to $2^m - 2^k$. Once v_1 is selected to be in L, then all vectors in $v_1 + L_n$ are added to L, where $v_1 + L_n$ is the set of vectors obtained by adding v_1 to all possible vectors in L_n. Therefore, by picking v_1, the total number of vectors of $\mathbb{F}_2^m$ that do not belong to L is now equal to $2^m - 2^{k+1}$, out of which we pick v_2. The above process is repeated until all vectors $\{v_{[T-k]}\}$ are selected. Therefore, the total number of such choices becomes $\prod_{\ell=0}^{T-k-1} \left(2^m - 2^{k+\ell}\right)$. In order to compute the total number of subspaces, we need to divide this number by the total number of basis vectors (i.e., linearly independent vectors) used to represent the vectors in $L \backslash L_n$; we denote them by $\{b_1, \cdots, b_{T-k}\}$. The number of vectors in such a basis is $T-k$. Given a subspace L, we pick b_1 from the set of vectors in $L \setminus L_n$, the number of which is $2^T - 2^k$. Then we pick b_2 from the set of vectors $L \setminus (L_n + b_1)$, the number of which is $2^T - 2^{k+1}$. We repeat the previous argument for all $T-k$ vectors. The total number of such basis vectors is therefore equal to $\prod_{\ell=0}^{T-k-1} \left(2^T - 2^{k+\ell}\right)$. Dividing the two quantities therefore proves Lemma III.1.

Appendix B
Proof of Theorem III.3

To prove Theorem III.3, we first recall the definition of $\mathcal{G}(q_i, \mathcal{S}_i)$. Given q_i and $\mathcal{S}_i$, $\mathcal{G}(q_i, \mathcal{S}_i)$ is the set which contains all possible i-th vectors $\mathbf{g}_i$ of the realization G of the matrix $\mathbf{G}$, namely

$$\mathcal{G}(q_i, \mathcal{S}_i) = \left\{ \mathbf{g} \in \mathbb{F}_2^m \mid g_{q_i} = 1, g_{[m]\setminus\{q_i \cup \mathcal{S}_i\}} = 0 \right\}.$$

In addition, we define the following set. Given $\mathbf{g}_i$ and an integer r, we let $\mathcal{D}(\mathbf{g}_i, r)$ be the set of all possible matrices $\mathbf{A}_k^{(i)}$ of r rows from which $\mathbf{g}_i$ can be reconstructed, namely

$$\mathcal{D}(\mathbf{g}_i, r) = \left\{ \mathbf{Z} \in \mathbb{F}_2^{r \times m} \mid \exists \mathbf{d} \in \mathbb{F}_2^r \text{ s.t. } \mathbf{g}_i = \mathbf{d}\mathbf{Z} \right\}.$$

Note that the definition of $\mathcal{D}(\mathbf{g}_i, r)$ is different than that of $\mathcal{P}(\mathbf{g}_i, \mathbf{A}_k, r)$ in that it is not dependent on a specific matrix $\mathbf{A}_k$. Then, we can write

$$\begin{aligned} P_k^{(\text{MIL})} &= \mathcal{L}(A \to A_k^{(n)} | Q_n = q_n, S_n = \mathcal{S}_n) \\ &\overset{(a)}{\leq} \log \left| A_k^{(n)} | Q_n = q_n, S_n = \mathcal{S}_n \right| \\ &\overset{(b)}{=} \log \left| \bigcup_{r=1}^{k} \bigcup_{\mathbf{g}_n \in \mathcal{G}(q_n, \mathcal{S}_n)} \mathcal{D}(\mathbf{g}_n, r) \right| \end{aligned}$$

$$\leq \log\left(\sum_{r=1}^{k}\sum_{\mathbf{g}_n\in\mathcal{G}(q_n,\mathcal{S}_n)}|\mathcal{D}(\mathbf{g}_n,r)|\right)$$
$$\overset{(c)}{=} \log\left(2^{|\mathcal{S}_n|}\sum_{r=1}^{k}|\mathcal{D}(\mathbf{g}'_n,r)|\right)$$
$$\overset{(d)}{\leq} \log\left(2^{|\mathcal{S}_n|}\sum_{r=1}^{k}\prod_{j=0}^{r-2}(2^m-2^{j+1})\right)$$
$$\leq \log\left(2^{|\mathcal{S}_n|}k(2^m-2)^{k-1}\right)$$
$$= O(|\mathcal{S}_n|+mk),$$

where: (i) the equality in (a) follows from Property 2 of the MIL; (ii) the equality in (b) follows by noting that, given Q_n and S_n, a possible $A_k^{(n)}$ would belong to $\mathcal{D}(\mathbf{g}_n,r)$ for some $r\in[k]$ and some $\mathbf{g}_n\in\mathcal{G}(Q_n,S_n)$; (iii) the equality in (c) follows by noting that, by symmetry, the number of matrices with r rows from which the vector $\mathbf{g}_i$ can be reconstructed is the same for every possible vector $\mathbf{g}_i\in\mathcal{G}(q_i,\mathcal{S}_i)$. Therefore, the sum over $\mathbf{g}_n$ can be replaced by $\mathcal{D}(\mathbf{g}'_n,r)\times|\mathcal{G}(q_n,\mathcal{S}_n)|$ where $\mathbf{g}'_n$ is any arbitrary vector in $\mathcal{G}(q_n,\mathcal{S}_n)$. Based on the structure of the vectors $\mathbf{g}_n\in\mathcal{G}(q_n,\mathcal{S}_n)$, i.e., one in position q_n and zeros in the positions $[m]\setminus\{q_n\cup\mathcal{S}_n\}$, it follows that $|\mathcal{G}(q_n,\mathcal{S}_n)|=2^{|\mathcal{S}_n|}$; (iv) the inequality in (d) is obtained by counting arguments similar to those in the proof of Lemma III.1. In particular, we enumerate the number of ways we can construct a matrix $\mathbf{A}_k^{(n)}$ with r linearly independent rows, which when linearly combined gives $\mathbf{g}'_n$. We first pick a row vector $v_1\in\mathbb{F}_2^m\setminus\text{Span}(\mathbf{g}'_n)$, where $\text{Span}(\mathcal{X})$ of a set of row vectors $\mathcal{X}$ is the row span of these vectors; the number of possible vectors v_1 is 2^m-2. Then, we pick a second row vector $v_2\in\mathbb{F}_2^m\setminus\text{Span}(\{\mathbf{g}'_n,v_1\})$; the number of possible vectors v_2 is 2^m-2^2. We repeat this argument for $r-1$ vectors; the r-th vector is then selected so that a linear combination of all r vectors is equal to $\mathbf{g}'_n$.

Appendix C
Proof of Theorem III.4

The upper bound on $\bar{P}_k^{(\text{MIL})}$ follows by using similar steps as in the derivation of the upper bound on $P_k^{(\text{MIL})}$ in Theorem III.3. Namely, we have

$$\bar{P}_k^{(\text{MIL})} = \mathcal{L}(A\to A|Q_n=q_n,S_n=\mathcal{S}_n)$$
$$\leq \log|A|Q_n=q_n,S_n=\mathcal{S}_n|$$
$$= \log\left|\bigcup_{r=1}^{T}\bigcup_{\mathbf{g}_n\in\mathcal{G}(q_n,\mathcal{S}_n)}\mathcal{D}(\mathbf{g}_n,r)\right|$$
$$\leq \log\left(\sum_{r=1}^{T}\sum_{\mathbf{g}_n\in\mathcal{G}(q_n,\mathcal{S}_n)}|\mathcal{D}(\mathbf{g}_n,r)|\right)$$
$$= \log\left(2^{|\mathcal{S}_n|}\sum_{r=1}^{T}|\mathcal{D}(\mathbf{g}'_n,r)|\right)$$
$$\leq \log\left(2^{|\mathcal{S}_n|}\sum_{r=1}^{T}\prod_{j=0}^{r-2}(2^m-2^{j+1})\right)$$
$$\leq \log\left(2^{|\mathcal{S}_n|}T(2^m-2)^{T-1}\right)$$
$$= O(|\mathcal{S}_n|+mT).$$

For the lower bound, we have

$$\bar{P}_k^{(\text{MIL})} = \mathcal{L}(A\to A|Q_n=q_n,S_n=\mathcal{S}_n)$$
$$\overset{(a)}{=} \log|\{A:p(A|Q_n=q_n,S_n=\mathcal{S}_n)>0\}|$$
$$\overset{(b)}{=} \log\left|\bigcup_{\mathbf{g}\in\mathcal{G}(q_n,\mathcal{S}_n)}\{A:\exists\mathbf{d}\in\mathbb{F}_2^T,\mathbf{g}=\mathbf{d}A\}\right|$$
$$\geq \log\left|\{A:\exists\mathbf{d}\in\mathbb{F}_2^T,\mathbf{g}'=\mathbf{d}A\}\right|$$
$$\overset{(c)}{\geq} \log|\{L\subseteq\mathbb{F}_2^m:\dim(L)=T,\mathbf{g}'\in L\}|$$
$$\overset{(d)}{=} \log\prod_{j=1}^{T-1}\left(\frac{2^m-2^j}{2^T-2^j}\right)$$
$$\overset{(e)}{\geq} \log\left(\frac{2^m-2}{2^T-2}\right)^{T-1} = \Omega\left(mT-T^2\right),$$

where: (i) the equality in (a) follows from Property 3 of the MIL; (ii) the equality in (b) follows by noting that the clients do not know the index coding algorithm used by the server; (iii) the inequality in (c) follows by letting $L\subseteq\mathbb{F}_2^m$ be a subspace of dimension $\dim(L)$; (iv) the equality in (d) follows by using Lemma III.1 with $k=1$ (since $\mathbf{g}'$ has only one row) and $t=T$; (iv) the inequality in (e) follows by noting that $\left(\frac{2^m-2^j}{2^T-2^j}\right)\geq\left(\frac{2^m-2}{2^T-2}\right)$ for $j\in[T-1]$.

Appendix D
Proof of Lemma III.5

Recall that the definition of the MIL is

$$P_k^{\text{MIL}} = \log\sum_{A_k^{(n)}\in\mathcal{A}_k^{(n)}}p(A_k^{(n)}|A^\star,Q_n=q_n,S_n=\mathcal{S}_n),$$

where

$$A^\star = \arg\max_{A:p(A|Q_n=q_n,S_n=\mathcal{S}_n)>0}p(A_k^{(n)}|A,Q_n=q_n,S_n=\mathcal{S}_n),$$

$\mathcal{A}_k^{(n)}$ is the set of all possible matrices $A_k^{(n)}$, and we denote by $q_n,\mathcal{S}_n$ the particular realizations of Q_n,S_n. Note also that,

$$p(A|Q_n=q_n,S_n=\mathcal{S}_n) = \sum_{Q_{[n-1]},S_{[n-1]}}p(Q_{[n-1]},S_{[n-1]})\times$$
$$p(A|Q_{[n-1]},S_{[n-1]},Q_n=q_n,S_n=\mathcal{S}_n).$$

Therefore, for $p(A|Q_n=q_n,S_n=\mathcal{S}_n)$ to be non-zero for a particular realization of A, it suffices that $p(A|Q_{[n]},S_{[n]})$ be non-zero for this A for some realization of $Q_{[n]},S_{[n]}$ inside the summation (since $p(Q_{[n]},S_{[n]})$ is a uniform distribution and therefore is positive for all values of $Q_{[n]},S_{[n]}$).

Then, consider the matrix

$$A^\star = \begin{bmatrix}A_k^{(n)}\\ B\end{bmatrix},$$

where B is a $T-k\times m$ matrix consisting of a collection of $T-k$ rows, each of weight 1 (i.e., each row vector of B consists of one 1 and $m-1$ zeros). Moreover, none of the

row vectors of B contains a 1 in the location corresponding to q_n. Denote by $\mathcal{P} \subseteq [m]$ the set of locations where there are 1 values in the rows of the matrix B. Clearly, $q_n \notin \mathcal{P}$.

Note that there are some realizations of the variables $Q_{[n]}, S_{[n]}$ that can be satisfied by the matrix $A^\star$ – specifically, consider the case where $Q_n = q_n, S_n = \mathcal{S}_n$ and $Q_i \in \mathcal{P}$ for all $i \in [n-1]$. For this configuration, client n can reconstruct its request using $A_k^{(n)}$, and clients $i \in [n-1]$ can reconstruct their requests using the matrix B. Therefore, we have $p(A^\star | Q_n = q_n, S_n = \mathcal{S}_n) > 0$. However, note that the only way for client n to be able to reconstruct its requested message is by using $A_k^{(n)}$ (because the matrix B does not include message q_n in any of the rows). Therefore, $p(A_k^{(n)} | A^\star, Q_n = q_n, S_n = \mathcal{S}_n) = 1$. The aforementioned argument is true for any matrix $A_k^{(n)}$. This concludes the proof of Lemma III.5.

Appendix E
Proof of Theorem IV.1 - Equation (10) and Lemma IV.2

Theorem IV.1 - Equation (10)**.** Given an index coding matrix $\mathbf{A}$, we denote by $V_{\mathbf{A}} \subseteq \mathbb{F}_2^m$ the subspace formed by the span of the rows of $\mathbf{A}$. It is clear that the dimension of $V_{\mathbf{A}}$ is at most T (exactly T if $\mathbf{A}$ is full rank) and that the n distinct rows of $\mathbf{G}$ lie in $V_{\mathbf{A}}$. Let $\mathbf{a}_i \in \mathbb{F}_2^m, i \in [T_k]$, be the i-th row of $\mathbf{A}_k$. Then, the problem of finding a lower bound on the value of T_k can be formulated as follows: *what is a minimum-size set of vectors $\mathcal{A}_k = \{\mathbf{a}_{[T_k]}\}$ such that any row vector of $\mathbf{G}$ can be represented by a linear combination of at most k vectors of $\mathcal{A}_k$?*

A lower bound on T_k can be obtained as follows. Given $\mathcal{A}_k$, there must exist a linear combination of at most k vectors of $\mathcal{A}_k$ that is equal to each of the n distinct row vectors of $\mathbf{G}$. The number of *distinct* non-zero linear combinations of up to k vectors is at most equal to $\sum_{j=1}^{k} \binom{T_k}{j}$. Thus, we have

$$\sum_{i=1}^{k} \binom{T_k}{i} \geq n. \tag{15}$$

Combining this with the fact that $T_k \geq T$ gives precisely the bound in (10).

Lemma IV.2. We now derive the lower bound in Lemma IV.2. We first consider the case where $n = 2^T - 1$. From (15), we obtain

$$\sum_{i=1}^{k} \binom{T_k}{i} \geq 2^T - 1. \tag{16}$$

Since in general $T_k \geq T$, to prove that $T_k \geq T+1$ for $k < T$, it is sufficient to show that we have a contradiction for $T_k = T$. Indeed, by setting $T_k = T$, the bound in (16) becomes

$$\sum_{i=1}^{k} \binom{T}{i} \geq 2^T - 1 = \sum_{i=1}^{T} \binom{T}{i},$$

which clearly is not possible since $k < T$. Hence, $T_k \geq T+1$ for all $k < T$.

For a general n and $1 \leq k < \lceil T/2 \rceil$, we have

$$k\left(\frac{T_k e}{k}\right)^k \geq k\binom{T_k}{k} \geq \sum_{i=1}^{k} \binom{T_k}{i} \geq n$$
$$\implies T_k \geq \frac{k^{\frac{k-1}{k}}}{e} n^{1/k} = \Omega(k n^{\frac{1}{k}}).$$

Therefore, $T_k = \Omega(k 2^{\frac{T}{k}})$ when $n = \Theta(2^T)$. This lower bound, along with the upper bound in equation (12) concludes the proof of Lemma IV.2.

References

[1] M. Karmoose, L. Song, M. Cardone, and C. Fragouli, "Private broadcasting: An index coding approach," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2543–2547.

[2] M. Karmoose, L. Song, M. Cardone, and C. Fragouli, "Preserving privacy while broadcasting: K-limited-access schemes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017, pp. 514–518.

[3] M. Karmoose, L. Song, M. Cardone, and C. Fragouli, "Privacy in index coding: Improved bounds and coding schemes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 831–835.

[4] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[5] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, 2006.

[6] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[7] Y. H. Ezzeldin, M. Karmoose, and C. Fragouli, "Communication vs distributed computation: An alternative trade-off curve," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2017, pp. 279–283.

[8] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Feb. 2011.

[9] H. Esfahanizadeh, F. Lahouti, and B. Hassibi, "A matrix completion approach to linear index coding problem," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2014, pp. 531–535.

[10] X. Huang and S. El Rouayheb, "Index coding and network coding via rank minimization," in *Proc. IEEE Inf. Theory Workshop-Fall (ITW)*, Oct. 2015, pp. 14–18.

[11] M. A. R. Chaudhry and A. Sprintson, "Efficient algorithms for index coding," in *Proc. IEEE INFOCOM Workshops*, Apr. 2008, pp. 1–4.

[12] M. Sved, "Gaussians and binomials," *Ars Combinatoria*, vol. 17, no. 1, pp. 325–351, 1984.

[13] I. Issa, S. Kamath, and A. B. Wagner, "An operational measure of information leakage," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2016, pp. 234–239.

[14] I. Issa, A. B. Wagner, and S. Kamath, "An operational approach to information leakage," 2018, *arXiv:1807.07878*. [Online]. Available: https://arxiv.org/abs/1807.07878

[15] J. G. Oxley, *Matroid Theory*, vol. 3. New York, NY, USA: Oxford Univ. Press, 2006.

[16] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Index coding via linear programming," 2010, *arXiv:1004.1379*. [Online]. Available: https://arxiv.org/abs/1004.1379

[17] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-Preserving Data Mining*. Boston, MA, USA: Springer, 2008, pp. 11–52.

[18] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, Nov. 1998.

[19] R. Freij-Hollanti, O. Gnilke, C. Hollanti, and D. Karpuk, "Private information retrieval from coded databases with colluding servers," Nov. 2016, *arXiv:1611.02062*. [Online]. Available: https://arxiv.org/abs/1611.02062

[20] Z. Chen, Z. Wang, and S. Jafar, "The capacity of T-private information retrieval with private side information," 2017, *arXiv:1709.03022*. [Online]. Available: https://arxiv.org/abs/1709.03022

[21] H. Sun and S. A. Jafar, "Private information retrieval from MDS coded data with colluding servers: Settling a conjecture by Freij-Hollanti et al." *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1000–1022, Feb. 2018.

[22] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.
[23] K. Banawan and S. Ulukus, "The capacity of private information retrieval from Byzantine and colluding databases," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 1206–1219, Feb. 2018.
[24] G. Brassard, C. Crepeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proc. Conf. Theory Appl. Cryptograph. Techn.* New York, NY, USA: Springer-Verlag, 1987, pp. 234–238.
[25] M. Mishra, B. K. Dey, V. M. Prabhakaran, and S. Diggavi, "The oblivious transfer capacity of the wiretapped binary erasure channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 1539–1543.
[26] S. H. Dau, V. Skachek, and Y. M. Chee, "On the security of index coding with side information," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3975–3988, Jun. 2012.
[27] V. Narayanan, V. M. Prabhakaran, J. Ravi, V. K. Mishra, B. K. Dey, and N. Karamchandani, "Private index coding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 596–600.
[28] I. Haviv and M. Langberg, "On linear index coding for random graphs," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2231–2235.
[29] L. Natarajan, P. Krishnan, and V. Lalitha, "On locally decodable index codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 446–450.
[30] G. Chen and D. Needell, "Compressed sensing and dictionary learning," in *Finite Frame Theory: A Complete Introduction to Overcompleteness*, vol. 73. New York, NY, USA: American Mathematical Society, Jul. 2016, p. 201.
[31] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

Mohammed Karmoose received the B.S. and M.S. degrees in electrical engineering from the Faculty of Engineering, Alexandria University, Egypt, in 2009 and 2013, respectively, and the Ph.D. degree from the Electrical and Computer Engineering Department, University of California at Los Angeles (UCLA). He was a Graduate Research Assistant with the CRN Research Group, E-JUST, Egypt, from 2011 to 2014. He is currently an Engineer with Samsung Semiconductors, Inc. His research interests are distributed detection, cooperative caching, and wireless communications. He received the Annual Tribute Ceremony Award for Top-Ranked Students from Alexandria University from 2005 to 2009 and the Electrical Engineering Department Fellowship from UCLA for his first year of Ph.D. study in 2014/2015.

Linqi Song (M'17) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, China, and the Ph.D. degree in electrical engineering from the University of California at Los Angeles (UCLA). He was a Post-Doctoral Scholar with the Electrical and Computer Engineering Department, UCLA. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests are in coding techniques, algorithms, big data, and machine learning. He received the UCLA Fellowship for his graduate studies.

Martina Cardone received the Ph.D. degree in electronics and communications from Télécom ParisTech, Paris, France (with work done at Eurecom, Sophia Antipolis, France), in 2015. From November 2017 to January 2018, she was a Post-Doctoral Associate with the Electrical and Computer Engineering Department, University of Minnesota (UMN). From July 2015 to August 2017, she was a Post-Doctoral Research Fellow with the Electrical and Computer Engineering Department, UCLA Henri Samueli School. She is currently an Assistant Professor with the Electrical and Computer Engineering Department, UMN. Her main research interests are in network information theory, network coding, and wireless networks with special focus on their capacity, security, and privacy aspects. She was a recipient of the NSF CRII Award in 2019, the Second Prize in the Outstanding Ph.D. Award from Télécom ParisTech, and the Qualcomm Innovation Fellowship in 2014.

Christina Fragouli (F'16) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles (UCLA), Los Angeles. She has worked at the Information Sciences Center, AT&T Labs, Florham Park New Jersey, and the National University of Athens. She also visited Bell Laboratories, Murray Hill, NJ, USA, and DIMACS, Rutgers University. From 2006 to 2015, she was an Assistant and Associate Professor with the School of Computer and Communication Sciences, EPFL, Switzerland. She is currently a Professor with the Electrical and Computer Engineering Department, UCLA. Her research interests are in network coding, wireless communications, network security, and privacy. She has served as an Information Theory Society Distinguished Lecturer and an Associate Editor for the IEEE COMMUNICATIONS LETTERS, the *Journal on Computer Communication* (Elsevier), the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON INFORMATION THEORY, and the IEEE TRANSACTIONS ON MOBILE COMPUTING.