Lower Bounds for Linear Locally Decodable Codes and Private Information Retrieval

Oded Goldreich* Howard Karloff† Leonard J. Schulman‡ Luca Trevisan§

November 14, 2001

Abstract

We prove that if a linear error correcting code $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ is such that a bit of the message can be probabilistically reconstructed by looking at two entries of a corrupted codeword, then $m = 2^{\Omega(n)}$. We also present several extensions of this result.

We show a reduction from the complexity of one-round, information-theoretic Private Information Retrieval Systems (with two servers) to Locally Decodable Codes, and conclude that if all the servers' answers are linear combinations of the database content, then $t = \Omega(n/2^a)$, where t is the length of the user's query and a is the length of the servers' answers. Actually, 2^a can be replaced by $O(a^k)$, where k is the number of bit locations in the answer that are actually inspected in the reconstruction.

*Computer Science Department, Weizmann Institute of Science, Rehovot, Israel. Supported by MINERVA Foundation, Germany. E-mail: `oded@wisdom.weizmann.ac.il`

†AT&T Labs–Research, USA. E-mail: `howard@research.att.com`

‡Caltech, USA. E-mail: `schulman@caltech.edu`

§Computer Science Division, University of California, Berkeley. E-mail: `luca@eecs.berkeley.edu`

Contents

1 Introduction **2**
- 1.1 Locally Decodable Codes 2
- 1.2 Private Informational Retrieval 4
- 1.3 Organization 5

2 Preliminaries **6**
- 2.1 Smooth Codes 6
- 2.2 The Recovery Graphs 7

3 Proof of Theorem 1.3 **8**
- 3.1 Getting Rid Of Projected Bits 8
- 3.2 The Combinatorial Lemma 8
- 3.3 A Combinatorial Proof of the Combinatorial Lemma 9
- 3.4 An Information-Theoretic Proof of the Combinatorial Lemma 11

4 Extension To Arbitrary Finite Fields – The Proof of Theorem 1.4 **12**
- 4.1 Getting Rid Of Multiples Of $\vec{e}_i$ 12
- 4.2 Reduction To The Boolean case 13

5 Extension To Binary Linear Block Codes – The Proof of Theorem 1.5 **14**
- 5.1 Reduction to the Boolean case 14
- 5.2 Consequences 15

6 Extension To Binary Linear Block Codes With Block Decoding – The Proof of Theorem 1.6 **16**
- 6.1 Getting Rid of Singular Multiples 17
- 6.2 Reduction To The Boolean case 17

7 Lower Bounds For Private Information Retrieval – Proof of Theorem 1.8 **19**
- 7.1 Constructing Smooth Codes Based on PIR Schemes 19
- 7.2 Consequences 20

Bibliography **21**

1 Introduction

This paper is concerned with two related notions. The first notion is that of locally decodable codes (LDC), which are error-correcting codes that allow recovery of individual information bits based on a few (randomly selected) codeword bits. The second notion is that of private information retrieval (PIR) schemes, which are protocols allowing users to retrieve desired data items from several (non-colluding) servers without yielding any information to any individual server. The relation between these notions has been observed by some researchers before, and is further established in this paper.

The study of LDCs was initiated by Katz and Trevisan [5], who established super-linear (but at most quadratic) lower bounds on the length of codes that allow recovery based on a constant number of bits. In contrast, the best known constructions of LDCs (supporting such efficient recovery) have sub-exponential length. This leaves a huge gap between the known lower and upper bounds, and an important research goal is to try to close this gap. We take a first step in this direction by closing the gap (via improved lower bounds) for the special case of *linear* LDCs in which recovery is based on *two* bits.

The study of PIR schemes was initiated by Chor, Goldreich, Kushilevitz and Sudan [4], who presented (among other schemes) a one-round, 2-server PIR scheme of communication complexity $O(n^{1/3})$. The question of whether their (2-server) PIR scheme has the lowest communication complexity possible has been open since. We present several results that are related to this question, where all our results relate to the special case of *one-round*, 2-server PIR schemes in which the servers' answers are always *linear* combinations of the data bits.

1.1 Locally Decodable Codes

In this paper we consider error-correcting codes with the following *local decodability* property: given a corrupted codeword it is possible to recover each bit of the original message by applying a probabilistic procedure that looks at only *two* entries of the corrupted codeword. The procedure should predict each bit with a constant advantage even when there is a constant fraction of errors in corrupted codeword. The Hadamard code satisfies this requirement, but unfortunately its codewords are exponentially longer than the message they encode. In this paper, we prove that this is essentially the best possible with respect to linear codes.

Let us first define formally the notion of a locally decodable code. For $x \in \Sigma^m$, we let x_i be the ith element of x, that is, $x = x_1 \cdots x_m$. For $y, z \in \Sigma^m$, we denote by $d(y, z)$ the number of locations on which y and z differ, that is, $d(y, z) = |\{i : y_i \neq z_i\}|$.

Definition 1.1 *For reals δ, ϵ and an integer q, we say that $\mathbf{C} : \Sigma^n \to \Gamma^m$ is a (q, δ, ϵ)-*locally decodable code *if there exists a probabilistic oracle machine A such that:*

- *In every invocation, A makes at most q queries (possibly adaptively).*
- *For every $x \in \{0,1\}^n$, for every $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$, and for every $i \in [n]$, we have*

$$\mathbf{Pr}[A^y(i) = x_i] \geq \frac{1}{2} + \epsilon,$$

where the probability is taken over the internal coin tosses of A.

An algorithm A satisfying the above requirements is called an (adaptive) (q, δ, ϵ)-local decoding algorithm *for* $\mathbf{C}$.

While it appears natural to allow adaptive reconstruction algorithms in our definition, we only know how to directly prove lower bounds in the non-adaptive case. Lower bounds for the non-adaptive case can be generalized to the adaptive case by using the following reduction.

Lemma 1.2 ([5]) *Let* $\mathbf{C} : \Sigma^n \to \Gamma^m$ *be an error-correcting code that has an adaptive* $(2, \delta, \epsilon)$*-local decoding algorithm. Then* $\mathbf{C}$ *also has a non-adaptive* $(2, \delta, \epsilon/|\Gamma|)$*-local decoding algorithm.*

All the results that we will state (from now on) refer to non-adaptive reconstruction procedures, and "local decoding algorithm" and "locally decodable code" will always refer to the non-adaptive case. We omit the statement of the results for the adaptive case (which can be obtained by the application of the above lemma).

As stated above, our work focuses on *linear* codes. In particular, we will consider the following settings:

- $\Sigma = \Gamma = F$ is a finite field, and the function $\mathbf{C} : F^n \to F^m$ is a linear mapping between the vector spaces F^n and F^m. In Section 3 we deal with the special case $\Sigma = \Gamma = GF(2)$, while in Section 4 we deal with general fields.
- $\Sigma = \{0,1\}$, $\Gamma = \{0,1\}^l$, and $\mathbf{C} : \{0,1\}^n \to \{0,1\}^{lm}$ is linear. We deal with this case in Section 5.
- $\Sigma = \Gamma = \{0,1\}^l$, and $\mathbf{C} : \{0,1\}^{ln} \to \{0,1\}^{lm}$ is linear. We deal with this case in Section 6.

Our main result is

Theorem 1.3 *Let* $\Sigma = \Gamma = \{0,1\}$*, and let* $\mathbf{C} : \Sigma^n \to \Gamma^m$ *be a* $(2, \delta, \epsilon)$*-locally decodable linear code. Then* $m \geq 2^{\epsilon\delta n/8}$.

The result has the following extensions to larger alphabets.

Theorem 1.4 *Let* $\mathbf{C} : F^n \to F^m$ *be a* $(2, \delta, \epsilon)$*-locally decodable linear code. Then* $m \geq 2^{\frac{\epsilon\delta}{16} \cdot n - 1 - \log_2 |F|}$.

Theorem 1.5 *Let* $\mathbf{C} : \{0,1\}^n \to (\{0,1\}^l)^m$ *be a* $(2, \delta, \epsilon)$*-locally decodable linear code, and suppose that the decoder uses only k predetermined bits out of the l bits that it receives as answer to each query. Then* $m \geq (1/f(k,l)) \cdot 2^{\epsilon\delta n/8f(k,l)}$, *where* $f(k,l) = \sum_{i=0}^{k} \binom{l}{i} \leq \min\{2^l, 2l^k\}$.

Theorem 1.6 *Let* $\mathbf{C} : (\{0,1\}^\ell)^n \to (\{0,1\}^\ell)^m$ *be a* $(2, \delta, \epsilon)$*-locally decodable code that is linear block-block code. Then* $m \geq 2^{\frac{\epsilon\delta}{16} \cdot n - (\ell+1)^2}$.

Theorem 1.4 is proved in Section 4, by an extension of the argument used in the proof of Theorem 1.3. Theorem 1.5 is proved in Section 5 by means of a reduction to the case $l = k = 1$ and an application of Theorem 1.3. Theorem 1.6 is proved in Section 6 by an extension of the argument used in the proof of Theorem 1.3.

1.2 Private Informational Retrieval

Loosely speaking, a Private Information Retrieval (PIR) scheme for k servers is a protocol by which a user can obtain the value of a desired bit out of n bits held by the servers without yielding the identity of this bit to any individual server (assuming that the servers do not cooperate in order to learn the identity of the desired bit). The aim is to obtain PIR schemes of low communication complexity (i.e., substantially lower than the obvious solution of having a server sends all n bits to the user). We focus on one-round PIR schemes that are protocols in which the user sends a single message to each server that responds also with a single message. In the definition below, Q represents the algorithm employed by the user to generate its queries, S_j represents the algorithm employed by the jth server, and R represents the recovery algorithm used by the user (once its gets the servers' answers).

Definition 1.7 *A* one-round, $(1-\delta)$-secure, 2-server PIR scheme for database size n, with recovery probability **p**, query size **t** and answer size **a** *is a quadruple of deterministic algorithms* $\overline{A} = (Q, S_1, S_2, R)$ *with the following properties.*

Algorithmic operation: *On input $i \in [n]$ and* (random-tape) *$r \in \{0,1\}^L$, algorithm Q outputs a pair of t-bit long* queries*; that is, $(q_1, q_2) \stackrel{\text{def}}{=} Q(i, r)$.*

On input a database *$x \in \{0,1\}^n$ and a* query *$q \in \{0,1\}^t$, algorithm S_1 (resp., S_2) returns an* answer *$S_1(x, q) \in \{0,1\}^a$ (resp., $S_2(x, q) \in \{0,1\}^a$).*

On input $i \in [n]$, $r \in \{0,1\}^L$, and answers $\alpha_1, \alpha_2 \in \{0,1\}^a$, algorithm R outputs a bit $R(i, r, \alpha_1, \alpha_2)$, which is supposed to be a guess of the entry x_i.

The recovery condition: *We denote by $\overline{A}(i, x)$ the random variable that represents the output of $R(i, r, S_1(x, q_1), S_2(x, q_2))$, where $(q_1, q_2) = Q(i, r)$ and the probability space is induced by the uniform distribution of $r \in \{0,1\}^L$. Then, for every $i \in [n]$ and $x \in \{0,1\}^n$, it must hold that $\mathbf{Pr}[\overline{A}(i, x) = x_i] \geq p$.*

The secrecy condition: *For $i \in [n]$, denote by $Q_1(i)$ (resp., $Q_2(i)$) the distribution induced on the first (resp., second) element of $Q(i, r)$ when r is uniformly distributed in $\{0,1\}^L$. Then, for every $i, j \in [n]$, the distributions $Q_1(i)$ and $Q_1(j)$ (resp., $Q_2(i)$ and $Q_2(j)$) are δ-close.*

Notice that we relax (and quantify) the security and recovery requirements; the traditional perfect requirements are obtained by setting $\delta = 0$ and $p = 1$. On the other hand, in the following, we restrict our attention to PIR schemes in which, for every fixed $q \in \{0,1\}^l$, $S_1(x, q)$ and $S_2(x, q)$ are linear functions of x (i.e., each bit of $S_1(x, q)$ and each bit of $S_2(x, q)$ is a linear combination of the bits of x). (The abovementioned PIR scheme of Chor et. al. [4] satisfies this requirement.)

Theorem 1.8 *Suppose there is a one-round, $(1-\delta)$-secure PIR scheme with 2 servers, linear answers, database size n, query size t, answer size a, and recovery probability $1/2 + \epsilon$. Suppose also that the user uses only k predetermined bit out of the a bits it receives as answer to each query. Then*

$$t > \frac{(\epsilon - \delta) \cdot n}{12 \cdot f(k, a)} - \log_2 f(k, a) - 3$$

where $f(k, a) = \sum_{i=0}^{k} \binom{a}{i} \leq \min\{2^a, 2a^k\}$.

As immediate corollaries we conclude that

- Any (secure, one-round) 2-server PIR scheme with linear answers of constant length must have queries of linear (i.e., $\Omega(n)$) length. (This extends a simple lower bound (of $n-1$ bits) on the length of queries in a 2-server PIR scheme with single-bit linear answers [4, Sec. 5.2].)
- Any (secure, one-round) 2-server PIR scheme with linear answers in which the user uses only one bit from each answer must have communication complexity $\Omega(\sqrt{n})$.
- Any (secure, one-round) 2-server PIR scheme with linear answers in which the user uses only k bits from each answer, must have communication complexity $\Omega(n^{1/(k+1)})$.

In the abovementioned PIR scheme of Chor et. al. [4], both a and t are $O(n^{1/3})$, and $k = 4$. By a minor modification to that scheme, we can reduce k to 3. Thus the third lower bound asserts that for this case (i.e., $k = 3$), communication complexity of $\Omega(n^{1/4})$ is essential. We comment that the first two lower bounds are tight:

- There exists a (perfectly secure, one-round) 2-server PIR scheme that uses n-bit queries and linear answers that are single bits (cf., [4, Sec. 3.1]).
- There exists a (perfectly secure, one-round) 2-server PIR scheme in which the user uses only one bit from each $\sqrt{n}$-bit long linear answer, and the queries are also $\sqrt{n}$-bit long strings (e.g., by a minor modification of the scheme in [4, Sec. 3.2–3.3] as applied to $d = 2$).

Perspective: Computational security. We stress that the above results (as well as Section 7) refer to an information-theoretic notion of security. A relaxed notion of security, requiring only security with respect to polynomial-time servers, was put forward and first investigated by Chor and Gilboa [3]. Assuming the existence of one-way functions, for any $\epsilon > 0$, they presented 2-server computational-secure PIR schemes of communication complexity $O(n^\epsilon)$. Furthermore, their PIR schemes are one-round and use linear 1-bit answers. Combined with our results (or actually even with [4, Sec. 5.2]), this provides another PIR setting in which the relaxed notion of computational security offers an advantage over information-theoretic security. (The other PIR setting we refer to is the single-server setting in which n bits is a lower bound in the case of information-theoretic security [4, Sec. 5.1], whereas communication complexity of $O(n^\epsilon)$ can be achieved for computationally-secure PIR's [6], assuming the intractability of the quadratic residuousity problem.)

1.3 Organization

Most of the paper is devoted to analysis of several types of locally decodable codes, and the application to private information retrieval is postponed to the last section (Section 7).

We start the analysis of locally decodable codes by using a known reduction (due to Katz and Trevisan [5]) to a combinatorial problem. In case of linear codes the reduction yields a special case for which we obtain (in Section 3) stronger bounds than the ones obtained in [5]. Indeed, this improvement (applicable for the case of linear codes) is the source of all our lower bounds. We extend our analysis in three directions:

1. In Section 4, we consider linear codes over arbitrary fields (rather than over the field $GF(2)$).

   Our lower bound in this case is exponential in n, but inversely proportional to the size of the field.
2. In Section 5, we consider linear codes in which the decoder may read two l-bit long blocks in order to recover one input bit.

   Our lower bound in this case is exponential in $n/2^l$, with an improvement to $n/\min\{2^l, l^k\}$ in case the decoder only uses k out of the l bits in each retrieved block.
3. In Section 6, we consider linear codes in which the decoder may read two l-bit long blocks in order to recover one l-bit long input block.

   Our lower bound in this case is exponential in $n - l^2$.

2 Preliminaries

The notions and results in this section are mostly due to Katz and Trevisan [5]. In particular, their notion of smooth codes and its relation to locally decodable codes are central to our analysis. Here we generalize their definition to the case in which the message is over a non-Boolean alphabet.

2.1 Smooth Codes

Informally, a code is smooth if a corresponding local decoding algorithm "spreads its queries almost uniformly" (or, actually, does not query any code location too frequently).

Definition 2.1 *For reals c, ϵ, and integer q we say that $\mathbf{C} : \Sigma^n \to \Gamma^m$ is a (q, c, ϵ)-*smooth code *if there exists a probabilistic oracle machine A such that:*

- *In every invocation, A makes at most q queries nonadaptively.*
- *For every $x \in \{0,1\}^n$ and for every $i \in [n]$, we have:*

$$\mathbf{Pr}[A^{\mathbf{C}(x)}(i) = x_i] \geq \frac{1}{2} + \epsilon.$$

- *For every $i \in [n]$ and $j \in [m]$, the probability that on input i machine A queries index j is at most c/m.*

*(The probabilities are taken over the internal coin tosses of A.) An algorithm A satisfying the above requirements is called a (q, c, ϵ)-*smooth decoding algorithm *for $\mathbf{C}$.*

We stress that the decoding condition in Definition 2.1 refers only to valid codewords, whereas the corresponding condition in Definition 1.1 refers to all oracles that are sufficiently close to valid codewords. To get a feeling for the smoothness condition note that if the decoding machine spreads its queries uniformly, then we would get $c = q$ (and this is the lowest possible assuming that the machine always makes q queries). It turns out that any locally decodable code is smooth, for suitable parameters and by possible modification of the decoding machine.

Theorem 2.2 (See Theorem 1 in [5]) *Let $\mathbf{C} : \Sigma^n \to \Gamma^m$ be a (q, δ, ϵ)-locally decodable code. Then $\mathbf{C}$ is also a $(q, q/\delta, \epsilon)$-smooth code.*

We note that [5, Thm. 1] is stated for the case $\Sigma = \{0, 1\}$, but the proof (in [5]) applies to the general case as well.

2.2 The Recovery Graphs

Let $\mathbf{C} : \Sigma^n \to \Gamma^m$ be a $(2, c, \epsilon)$-smooth code and let algorithm A be a (non-adaptive) $(2, c, \epsilon)$-smooth decoding algorithm for $\mathbf{C}$. Let (q_1, q_2) be a pair of elements of $[m]$. We say that a given invocation of A reads (q_1, q_2) if the set of indices that A reads in that invocation equals $\{q_1, q_2\}$.[1] We say that (q_1, q_2) is good for i if:

$$\mathbf{Pr}[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } (q_1, q_2)] > 1/2.$$

where the probability is taken over x uniformly chosen from $\{0, 1\}^n$, and over the internal coin tosses of A. For every $i \in [n]$, we consider the graph with edge set consisting of the set of good pairs.

Definition 2.3 *Fixing a code $\mathbf{C} : \{0, 1\}^n \to \Gamma^m$ and a 2-query recovery algorithm A, the* recovery graph for $i \in [n]$, *denoted G_i, consists of the vertex set $[m]$ and the edge set E_i that equals the set of pairs (q_1, q_2) that are good for i.*

We have the following result about such graphs.

Lemma 2.4 ([5]) *Let $\mathbf{C}$ be a $(2, c, \epsilon)$-smooth code and $\{G_i\}_{i=1}^n$ be the associated set of recovery graphs. Then, for every i, the graph $G_i = ([m], E_i)$ has a matching $M_i \subseteq E_i$ of size at least $\epsilon m/c$.*

This is essentially Lemma 4 in [5], but, since we slightly changed the definition of the recovery graph (from [5]), and get slightly better bounds, we present a proof below.
Proof: We may assume without loss of generality that, for every $i \in [n]$ and $j_1, j_2 \in [m]$,

$$\mathbf{Pr}[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } (j_1, j_2)] \geq \frac{1}{2} \tag{1}$$

where the probability is taken uniformly over $x \in \{0, 1\}^n$ and A's internal coin tosses. (For example, we can modify A so that it outputs a random bit whenever $i \in [n]$ and $j_1, j_2 \in [m]$ do not satisfy Eq. (1).) Using a Markov argument, it follows that with probability at least 2ϵ, on input $i \in [n]$, algorithm A generates a pair that is good for i. In other words, with probability at least 2ϵ, the pair generated by $A(i)$ is an edge in G_i. Thus, if $C \subseteq [m]$ is a *vertex cover* of G_i, then the probability that $A(i)$ queries at least one element of C is at least 2ϵ. On the other hand, no element of $[m]$ is queried by A with probability greater than c/m, and so it follows that $|C| \geq (2\epsilon)/(c/m) = 2\epsilon m/c$. Since, the size of the maximum matching in a graph is at least half the size of the minimum vertex cover, we conclude that G_i has a matching of size at least $\epsilon m/c$. ■

[1] We assume, without loss of generality, $q_1 \neq q_2$.

3 Proof of Theorem 1.3

3.1 Getting Rid Of Projected Bits

To simplify the rest of our analysis, we would like to get rid of bits in the range of the code that are identical to some input (data) bit. That is, we wish the code to be such that no single bit of the output is equal to a particular bit of the input. We can accommodate this condition by fixing bits of the input that are identical to too many bits in the output. This gives the following lemma.

Lemma 3.1 *For $n > 4c/\epsilon$, let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ be a (q,c,ϵ)-smooth code. Then there is another code $\mathbf{C}' : \{0,1\}^{n'} \to \{0,1\}^{m'}$ that has a $(q,c,\epsilon/2)$-smooth reconstruction procedure A', such that $n' \geq n/2$, $m' \leq m$, and for every i and j there exists an $x \in \{0,1\}^{n'}$ such that the jth bit of $\mathbf{C}'(x)$ is different from x_i. Furthermore, if $\mathbf{C}$ is a linear code, then so is $\mathbf{C}'$.*

Thus lower bounds on the length of smooth codes satisfying the conclusion of the lemma yield lower bounds on general smooth codes.

Proof: Consider the set I of bits in the input that occur in more than a fraction $2/n$ of the bits of the output. Clearly, $|I| \leq n/2$. For each $i \in [n] \setminus I$, consider the behavior of the smooth reconstruction procedure $A^{\mathbf{C}(x)}(i)$ for some x. Since $i \notin I$, at most a fraction $2/n$ of the bits of $\mathbf{C}(x)$ contain copies of x_i. By the smoothness condition, such code bits are examined with probability at most $2c/n$, which is less than $\epsilon/2$ (provided that $n > 4c/\epsilon$). Thus, if we modify A such that it does not read such bits, we may decrease the probability that it recovers x_i by at most $\epsilon/2$, so the recovery condition is met.

We construct the code $\mathbf{C}'$ from $\mathbf{C}$ by omitting the output bits that are copies of any input bit $i \in [n]$, fixing arbitrary[2] values for the bits in I, "hardwiring" these values into $\mathbf{C}'$, and modifying A so that it queries only bits in $\mathbf{C}'$ (rather than bits in $\mathbf{C}$). Note that the fact that the length of $\mathbf{C}'$ may be shorter than the length of $\mathbf{C}$ only makes the smoothness condition easier to meet. ■

3.2 The Combinatorial Lemma

We will deal with the linear error correcting code $\mathbf{C}'$ of Lemma 3.1. In the following we will use e_i to denote a vector in $\{0,1\}^n$ that has 1 in the i-th coordinate and 0 elsewhere. We can identify our error-correcting code $\mathbf{C}'$ with a sequence of m' vectors $a_1, \ldots, a_{m'} \in \{0,1\}^{n'}$, such that the jth bit of $\mathbf{C}(x)$ is $a_j \cdot x$. Recall that, by Lemma 3.1, none of these a_j's equals any unit vector e_i. Let $\{G_i\}_{i=1}^{n'}$ be the sequence of recovery graphs associated to $\mathbf{C}'$ as in Lemma 2.4.

Lemma 3.2 *For every i, and for every $(q_1, q_2) \in E_i$, e_i is in the span of $\{a_{q_1}, a_{q_2}\}$.*

Proof: Suppose e_i is linearly independent of a_{q_1} and a_{q_2}. Then, for a random x, the value $x \cdot e_i$ is independent (in the statistical sense) of the values $x \cdot a_{q_1}$ and $x \cdot a_{q_2}$, and so it

[2] Actually, in order to preserve linearity, these bits should all be set to zero. However, in fact, all our results apply also to affine codes.

is not possible to gain any advantage in predicting x_i by looking at the q_1th and the q_2-th bit of the encoding of x. ■

Since we are dealing with the field $\{0,1\}$, when e_i is in the span of $\{a_{q_1}, a_{q_2}\}$ there are only three possibilities: either a_{q_1} or a_{q_2} equals e_i itself, or $e_i = a_{q_1} \oplus a_{q_2}$. But for $\mathbf{C}'$ (as in Lemma 3.1) the only possible case is that $e_i = a_{q_1} \oplus a_{q_2}$. Thus proving Theorem 1.3 reduces to proving the following result.

Lemma 3.3 (Combinatorial Lemma) *Let $\gamma > 0$ be a real and $a_1, \ldots, a_m$ be elements of $\{0,1\}^n$ such that for every $i \in [n]$ there is a set M_i of at least γm disjoint pairs of indices (j_1, j_2) such that $e_i = a_{j_1} \oplus a_{j_2}$. Then $m \geq 2^{\gamma n}$. Furthermore, the conclusion holds even when the hypothesis only states that $\frac{1}{n}\sum_{i=1}^{n}|M_i| \geq \gamma m$.*

Below, we will present two alternative proofs of Lemma 3.3. Combining all the above lemmas, we get:

Corollary 3.4 *Let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ be a $(2, c, \epsilon)$-smooth linear code. Then $m \geq 2^{\epsilon n/(4c)}$.*

Notice that Theorem 1.3 is an immediate consequence of Corollary 3.4 and Theorem 2.2:

Proof: We first apply Lemma 3.1 to obtain a $(2, c, \epsilon')$-smooth linear code $\mathbf{C}' : \{0,1\}^{n'} \to \{0,1\}^{m'}$, for $n' \geq n/2$, $m' \leq m$ and $\epsilon' = \epsilon/2$. Combining Lemmas 2.4 and 3.2, it follows that $\frac{1}{n'}\sum_{i=1}^{n'}|M_i| \geq \epsilon' m'/c$. Finally, applying Lemma 3.3, we get $m' \geq 2^{\epsilon' n'/c} \geq 2^{\epsilon n/4c}$, and using $m \geq m'$ the claim follows. ■

3.3 A Combinatorial Proof of Lemma 3.3

For starters, let us suppose that all the vectors $a_1, \ldots, a_m$ are different. In this special case, Lemma 3.3 is a consequence of the following known combinatorial result.[3]

Lemma 3.5 (See Section 16 in [2]) *For any subset $S \subseteq \{0,1\}^n$ of the hypercube, the number of edges of the hypercube having both endpoints in S is at most $\frac{1}{2}|S|\lceil \log_2 |S| \rceil$.*

Note that our (distinct) vectors $a_1, \ldots, a_m$ are all vertices of a hypercube, and we are assuming that, for every i, there are at least γm edges in the ith "direction" between such vertices. This gives a total of at least γmn edges, but this number has to be no more than $\frac{1}{2}m\lceil \log_2 m \rceil < \frac{1}{2}m \log_2 2m$, and so it follows that $2m \geq 2^{2\gamma n}$.

To complete the proof of Lemma 3.3, we have to consider the case that $a_1, \ldots, a_m$ are not all different. Note that an analogue of Lemma 3.5 does not hold in this case (e.g., if $a_1 = \cdots = a_{m/2} = 0^n$ and $a_{(m/2)+1} = \cdots = a_m = 10^{n-1}$ then we get $(m/2)^2$ edges).[4]

---

[3] The proof of Theorem 2 in [2, Sec. 16] implies that the subset $S \subseteq \{0,1\}^n$ for which the number of internal edges is maximum is the set of the first $m = |S|$ strings in lexicographic order (of $\{0,1\}^n$). Since each such vertex has at most $\lceil \log_2 m \rceil$ internal edges, we get a bound of $\frac{1}{2}|S|\lceil \log_2 |S| \rceil$ on the number of internal edges.

[4] Note that this example does not violate Lemma 3.3: for every sequence of M_i's as in Lemma 3.3, it holds that $\sum_{i=1}^{n}|M_i| \leq 1$ (since $|M_1| \leq 1$ and all the other M_i's must be empty). Thus, the furthermore-hypothesis only holds with $\gamma \leq 1/nm$, implying a lower bound of $m \geq 2^{\gamma n} \geq 2$ (which indeed holds).

For every $a \in \{0,1\}^n$, let us denote by ν_a the number of indices j such that $a_j = a$ (so that $\sum_{a\in\{0,1\}^n} \nu_a = m$). That is, ν_a is the multiplicity of the vector a in the sequence $a_1, \ldots, a_m$. For every k, let us denote by S_k the set of vectors a such that $\nu_a \geq k$, and let $s_k = |S_k|$; observe that

$$\sum_k s_k = m \tag{2}$$

because each vector a that occurs in the sequence $a_1, \ldots, a_m$ is counted exactly ν_a times. Finally, define $\chi(a, j)$ to be 1 if $\nu_a \geq j$ and to be 0 otherwise. With this new piece of notation we can write

$$\sum_{a\in\{0,1\}^n} \sum_{k\geq 1} \chi(a, k) = m, \tag{3}$$

and we also note that for any two vectors $a, b \in \{0,1\}^n$ we have

$$\min\{\nu_a, \nu_b\} = \sum_{k\geq 1} \chi(\nu_a, k)\chi(\nu_b, k). \tag{4}$$

Now we would like to argue that for every i, the following upper bound holds on the size of the matching M_i:

$$|M_i| \leq \sum_{a,b:a\oplus b=e_i} \min\{\nu_a, \nu_b\} \tag{5}$$

Indeed, for starters we have by definition that M_i is the set of all pairs (j_1, j_2) such that $a_{j_1} \oplus a_{j_2} = e_i$, and all such pairs are disjoint. Let us fix two vectors a and b such that $a \oplus b = e_i$, and consider how many possible pairs (j_1, j_2) can belong to M_i subject to $a_{j_1} = a$ and $a_{j_2} = b$; since the pairs have to be disjoint, both ν_a and ν_b are upper bounds on the number of such possible pairs. Summing over all choices of a and b gives the bound of (5).

Combining the lemma's hypothesis with Equations (5) and (4), we get

$$\begin{aligned}
\gamma mn &\leq \sum_{i=1}^{n} |M_i| \\
&\leq \sum_{i=1}^{n} \sum_{a\in\{0,1\}^n} \min\{\nu_a, \nu_{a\oplus e_i}\} \\
&= \sum_{i=1}^{n} \sum_{a\in\{0,1\}^n} \sum_{k\geq 1} \chi(\nu_a, k)\chi(\nu_{a\oplus e_i}, k)
\end{aligned}$$

and so

$$\gamma mn \leq \sum_{k\geq 1} \sum_{i=1}^{n} \sum_{a\in\{0,1\}^n} \chi(\nu_a, k)\chi(\nu_{a\oplus e_i}, k). \tag{6}$$

Note that $\sum_{i=1}^{n} \sum_{a\in\{0,1\}^n} \chi(\nu_a, k)\chi(\nu_{a\oplus e_i}, k)$ counts (twice) the number of hypercube edges with both endpoints in S_k. Thus, by Lemma 3.5 we have, for every k, that

$$\begin{aligned}\sum_{i=1}^{n}\sum_{a\in\{0,1\}^n}\chi(\nu_a,k)\chi(\nu_{a\oplus e_i},k) &\leq 2\cdot\frac{1}{2}|S_k|\,\lceil\log_2|S_k|\rceil \\ &= s_k\lceil\log_2 s_k\rceil \;<\; s_k\cdot(1+\log_2 m).\end{aligned}$$

Combining this inequality with (6), and recalling (2), we have

$$\gamma mn < \sum_k s_k\cdot(1+\log_2 m) = m\cdot\log_2 2m,$$

from which it follows that $m \geq 2^{\gamma n-1}$. This is almost what the lemma claims (i.e., the lemma's claim is $m \geq 2^{\gamma n}$). The alternative proof (presented below) establishes the lemma as stated. The difference is insignificant to our applications (i.e., it amounts to a loss of a factor of 2 in the exponential lower-bounds).

3.4 An Information-Theoretic Proof of Lemma 3.3

The proof in this section is due to Alex Samorodnitsky, and was suggested to us after we found the combinatorial proof presented in the previous subsection.

Let X be a random variable uniformly distributed in the multi-set $\{a_1,\ldots,a_m\}$. We will write $X = X_1X_2\cdots X_n$, where X_i denotes the ith bit of X, and $X_{i,j}$ denotes $X_i\cdots X_j$. We consider the entropy of X, denoted $H(X)$. On one hand, $H(X)\leq \log m$. On the other hand, we will prove that $H(X)\geq \gamma n$, and Lemma 3.3 will follow immediately.

We can express the entropy of X as

$$H(X) = H(X_1) + H(X_2|X_1) + \cdots + H(X_n|X_1\cdots X_{n-1}).$$

The value of the ith term, $H(X_i|X_1\cdots X_{i-1}) = H(X_i|X_{1,i-1})$, is given by the following formula:

$$H(X_i|X_{1,i-1}) = \sum_{b\in\{0,1\}^{i-1}} \mathbf{Pr}[X_{1,i-1}=b]\cdot H(X_i|X_{1,i-1}=b).$$

Observe that for any $0/1$ random variable Y, we have $H(Y)\geq \min\{\mathbf{Pr}[Y=0],\mathbf{Pr}[Y=1]\}$, so the above expression is at least

$$\sum_{b\in\{0,1\}^{i-1}} \mathbf{Pr}[X_{1,i-1}=b]\cdot\min\{\mathbf{Pr}[X_i=0|X_{1,i-1}=b],\mathbf{Pr}[X_i=1|X_{1,i-1}=b],\}$$

and this is at least

$$\sum_{b\in\{0,1\}^{i-1}} \mathbf{Pr}[X_{1,i-1}=b]\cdot\frac{1}{2}\cdot\mathbf{Pr}[X \text{ is an endpoint in an edge of } M_i|X_{1,i-1}=b],$$

which is exactly

$$\frac{1}{2}\cdot\mathbf{Pr}[X \text{ is an endpoint in an edge of } M_i] \geq \frac{1}{2}\cdot 2\gamma = \gamma.$$

Then $H(X)\geq\gamma n$ and so $m\geq 2^{\gamma n}$.

4 Extension To Arbitrary Finite Fields – The Proof of Theorem 1.4

We extend Theorem 1.3 to linear codes over any finite field F, where $F = GF(2)$ is a special case treated (slightly better) in Theorem 1.3.

Remember that Theorem 1.4 states that if we let $\mathbf{C} : F^n \to F^m$ be a $(2, \delta, \epsilon)$-locally decodable linear code, then $m \geq 2^{\frac{\epsilon\delta}{16} \cdot n - 1 - \log_2 |F|}$.

This result is proven by an argument analogous to the one in Section 3. Here we deal with vector spaces over an arbitrary finite field. Specifically, we let F denote any such field, and consider n-dimensional vectors over F. In particular, $\vec{e}_i$ denotes the n-dimensional vector that has 1 in coordinate i and zero in all other coordinates. We say that a pair of vectors $(\vec{u}, \vec{v}) \in F^n \times F^n$ spans a third vector $\vec{w} \in F^n$ if there exists $\alpha, \beta \in F$ such that $\vec{w} = \alpha\vec{u} + \beta\vec{v}$. Again, the analysis reduces to providing lower bounds on the cardinality of multi-sets that contain many disjoint pairs that span each $\vec{e}_i$.

Lemma 4.1 *Let F be a finite field, n an integer, and S a multi-set of F^n. For $i = 1, ..., n$, let M_i be a set of disjoint pairs of elements of S that span $\vec{e}_i$. Then*

$$\sum_{i=1}^{n} |M_i| \leq 2|S| + 2 \cdot |S| \cdot \log_2(|S| \cdot |F|).$$

Thus, if $\frac{1}{n} \sum_{i=1}^{n} |M_i| \geq \gamma|S|$, then $|S| \geq 2^{(\gamma n/2) - 1 - \log_2 |F|}$.

4.1 Getting Rid Of Multiples Of $\vec{e}_i$

Motivation: *Our first goal is to get rid of queries that are multiples of some unit vector $\vec{e}_i$. Intuitively, such queries have limited utility, as shown in Claim 4.2. One benefit of getting rid of such queries is that recovery via a remaining pair of queries requires to use both answers, that is, if the query vectors $\vec{u}$ and $\vec{v}$ span $\vec{e}_i$ then it must be the case that $e_i = \alpha\vec{u} + \beta\vec{v}$, for some $\alpha, \beta \in F \setminus \{0\}$.*

Let S be as in the lemma, and E_i denote the set of all pairs in S that span $\vec{e}_i$. (Recall that M_i is a subset of E_i, consisting only of disjoint pairs.) Define

$$S' \stackrel{\text{def}}{=} S \setminus \{\alpha\vec{e}_i : \alpha \in F \,\&\, i = 1, ..., n\} \tag{7}$$

$$E'_i \stackrel{\text{def}}{=} E_i \cap (S' \times S') \tag{8}$$

$$M'_i \stackrel{\text{def}}{=} M_i \cap (S' \times S'). \tag{9}$$

Claim 4.2 $\sum_{i=1}^{n} |M_i| \leq 2|S| + \sum_{i=1}^{n} |M'_i|$.

Proof: We bound from above the number of pairs in $\cup_i M_i$ with an endpoint in $S \setminus S'$. We consider two types of pairs:

1. *A pair $(\vec{u}, \vec{v})$ such that either $\vec{u}$ or $\vec{v}$ is a multiple of some $\vec{e}_i$.* The number of such pairs is bounded from above by $2 \cdot |S \setminus S'|$, because element of the form $\alpha\vec{e}_i$ can "account" for at most one pair.

2. *A pair $(\vec{u},\vec{v})$ such that for some i and $\alpha,\beta \in F\setminus\{0\}$, $\vec{e}_i = \alpha\vec{u}+\beta\vec{v}$.* Suppose, without loss of generality, that $\vec{u} = \gamma\vec{e}_j$ and $\vec{v} = \delta\vec{e}_i + \eta\vec{e}_j$. Then $\vec{v}$ contributes to $M_i \setminus M'_i$, but cannot contribute (under this case) to any $M_k \setminus M'_k$ with $k \notin \{i,j\}$ (because if $(\vec{u}',\vec{v}) \in M_k \setminus M'_i$ for $k \notin \{i,j\}$, then $\vec{u}'$ must be a multiple of $\vec{e}_k$ and it must hold that $\vec{e}_k = \alpha'\vec{u}' + \beta'\vec{v}$ with $\beta' = 0$; so this pair is not counted in the current case). It follows that the number of such pairs is bounded above by $2|S'|$.

Combining the two types, the claim follows. ■

4.2 Reduction To The Boolean case

Motivation: *The first step in the reduction is to convert the system into one in which recovery is via fixed coefficients. Specifically, we shall define a redundant form of S' such that each $\vec{v} \in S'$ will be represented by its $|F|-1$ nonzero multiples. Recovery of the ith bit via queries $\vec{u}$ and $\vec{v}$ with multipliers α and $-\beta$ will be replaced by queries $\alpha\vec{u}$ and $\beta\vec{v}$ and straight addition.*

Let S' be a multi-set as above. Define

$$S'' \stackrel{\rm def}{=} \{\langle\vec{u},\alpha\rangle : \vec{u} \in S' \,\&\, \alpha \in F\setminus\{0\}\}, \tag{10}$$

$$E''_i \stackrel{\rm def}{=} \{(\langle\vec{u},\alpha\rangle,\langle\vec{v},\beta\rangle) \in S''\times S'' : \exists\gamma \in F\setminus\{0\} \text{ s.t. } \alpha\vec{u}-\beta\vec{v} = \gamma\vec{e}_i\}. \tag{11}$$

That is, if $\vec{u}$ occurs with multiplicity m in S', then (for every $\alpha \in F\setminus\{0\}$) $\langle\vec{u},\alpha\rangle$ occurs with multiplicity m in S''. Clearly, if $(\langle\vec{u},\alpha\rangle,\langle\vec{v},\beta\rangle) \in E''_i$, then $(\vec{u},\vec{v}) \in E'_i$. On the other hand, if $(\vec{u},\vec{v}) \in E'_i$, then there exists $\alpha,\beta,\gamma \in F\setminus\{0\}$ such that $\alpha\vec{u}-\beta\vec{v} = \gamma\vec{e}_i$, and thus there exists $\delta \in F\setminus\{0\}$ (i.e., $\delta = \beta/\alpha$) such that $(\langle\vec{u},\eta\rangle,\langle\vec{v},\delta\eta\rangle) \in E''_i$ for every $\eta \in F\setminus\{0\}$.

Let M''_i be defined as follows. For every $(\vec{u},\vec{v}) \in M'_i$ such that $\alpha\vec{u}-\beta\vec{v}$ is a multiple of $\vec{e}_i$, and for every $\delta \in F\setminus\{0\}$, add $(\langle\vec{u},\delta\alpha\rangle,\langle\vec{v},\delta\beta\rangle)$ to M''_i. Note that since $\vec{u},\vec{v} \in S'$ are not multiples of $\vec{e}_i$, it must be the case that $\alpha,\beta \neq 0$, and thus indeed $M''_i \subseteq E''_i$.

Claim 4.3 *1. $|S''| = (|F|-1)\cdot|S'|$.*

2. $\sum_{i=1}^n |M''_i| = (|F|-1)\cdot\sum_{i=1}^n |M'_i|$.

3. M''_i is a set of disjoint pairs in E''_i.

Proof: All items are obvious by the definition. In particular, by the above discussion, $M''_i \subseteq E''_i$, and the disjointness of pairs introduced for each single $(\vec{u},\vec{v}) \in M'_i$ follows similarly. Specifically, for every $(\vec{u},\vec{v}) \in M'_i$, there exist $\alpha,\beta \in F\setminus\{0\}$ such that $\alpha\vec{u}-\beta\vec{v}$ is a multiple of $\vec{e}_i$. Thus the pairs $(\langle\vec{u},\delta\alpha\rangle,\langle\vec{v},\delta\beta\rangle)$ added to M''_i, for every $\delta \in F\setminus\{0\}$, are disjoint (because $\delta\alpha = \delta'\alpha$ implies $\delta = \delta'$, and similarly for $\delta\beta = \delta'\beta$). ■

Motivation: *The main reduction step in the reduction is carried out in the following proof. It relies on the fact that if $\vec{u}'-\vec{v}' = \gamma\vec{e}_i$, with $\gamma \in F\setminus\{0\}$, then $\vec{u}'$ and $\vec{v}'$ agree on all but their ith coordinate (and they differ on their ith coordinate).*

Claim 4.4 *Let S''' be an arbitrary subset of $F^n \times F$ and M'''_i be an arbitrary set of disjoint pairs such that $(\langle\vec{u},\alpha\rangle,\langle\vec{v},\beta\rangle) \in M'''_i$ implies $\alpha\vec{u}-\beta\vec{v} = \gamma\vec{e}_i$ for some $\gamma \in F\setminus\{0\}$. Then $|S'''|\log_2|S'''| \geq \frac{1}{2}\cdot\sum_{i=1}^n |M'''_i|$.*

Proof: We consider a randomized mapping of $F^n \times F$ to $\{0,1\}^n$. The mapping is based on a uniformly chosen 2-coloring of F, denoted χ, and $\langle \vec{u}, \alpha \rangle \in F^n \times F$ is mapped to $\chi(v_1)\cdots\chi(v_n)$, where $(v_1,...,v_n) = \alpha\vec{u}$. Let us denote by $\mu_\chi : F^n \times F \to \{0,1\}^n$ the mapping induced by the 2-coloring $\chi : F \to \{0,1\}$, that is, $\mu_\chi(\vec{u},\alpha) = \chi(v_1)\cdot\chi(v_n)$, where $(v_1,...,v_n) = \alpha\vec{u}$. Thus the multi-set S''' is randomly mapped (by μ_χ) to a multi-set B_χ of $\{0,1\}^n$ such that $|B_\chi| = |S'''|$.

The key observation is that for every $(\langle \vec{u},\alpha\rangle, \langle \vec{v},\beta\rangle) \in M_i'''$, with probability $\frac{1}{2}$, it holds that $\mu_\chi(\vec{u},\alpha) \oplus \mu_\chi(\vec{v},\beta) = e_i$ (and otherwise $\mu_\chi(\vec{u},\alpha) = \mu_\chi(\vec{v},\beta)$). The observation follows by combining the fact that $\alpha\vec{u} = \beta\vec{v} + \gamma\vec{e}_i$, with $\gamma \in F \setminus \{0\}$, and the fact that $\mathbf{Pr}[\chi(e) = \chi(e+\gamma)] = \frac{1}{2}$ for every $e \in F$ (and $\gamma \in F \setminus \{0\}$). Letting $M_{i,\chi}$ denote the pairs in M_i''' that are mapped (by μ_χ) to pairs (u,v) such that $u \oplus v = e_i$, we conclude that the expected size of $M_{i,\chi}$ equals $\frac{1}{2} \cdot |M_i'''|$, where the expectation is taken uniformly over all possible χ's.

It follows that there exists a 2-coloring χ such that $\sum_{i=1}^n |M_{i,\chi}| \geq \frac{1}{2} \cdot \sum_{i=1}^n |M_i'''|$. Fixing this χ, we apply (the "furthermore" part of) Lemma 3.3 to B_χ and the $M_{i,\chi}$'s, and conclude that $|B_\chi| \log_2 |B_\chi| \geq \sum_{i=1}^n |M_{i,\chi}|$. Thus

$$|S'''| \log_2 |S'''| \;=\; |B_\chi| \log_2 |B_\chi| \;\geq\; \sum_{i=1}^n |M_{i,\chi}| \;\geq\; \frac{1}{2} \cdot \sum_{i=1}^n |M_i'''|.$$

■

Finishing the proof of Lemma 4.1: Using Item 3 of Claim 4.3, we may apply Claim 4.4 to S'' and the M_i'''s, and get $|S''| \log_2 |S''| \geq \frac{1}{2} \cdot \sum_{i=1}^n |M_i''|$. Applying the other items of Claim 4.3, we get

$$(|F|-1) \cdot |S'| \log_2(|F| \cdot |S'|) \geq (|F|-1) \cdot |S'| \log_2((|F|-1)|S'|) \geq \frac{1}{2} \cdot (|F|-1) \cdot \sum_{i=1}^n |M_i'|.$$

Thus $\sum_{i=1}^n |M_i'| \leq 2|S'| \log_2(|F| \cdot |S'|)$. Combining this with Claim 4.2, we get $\sum_{i=1}^n |M_i| \leq 2|S| + 2|S| \log_2(|F| \cdot |S|)$.

5 Extension To Binary Linear Block Codes – The Proof of Theorem 1.5

In this section we deal with linear codes mapping $\{0,1\}^n$ to $(\{0,1\}^\ell)^m$, where the case $\ell = 1$ corresponds to the main result (presented in Section 3). Thus each output symbol is an ℓ-bit long string, where each of these bits is a linear combination of the n input bits. We show that providing lower bounds for the general case reduces to providing lower bounds for the special case of $\ell = 1$.

5.1 Reduction to the Boolean case

Lemma 5.1 *Let* $\mathbf{C} : \{0,1\}^n \to (\{0,1\}^\ell)^m$ *be a* (q,c,ϵ)*-smooth linear error-correcting code. Then there is a code* $\mathbf{C}' : \{0,1\}^n \to \{0,1\}^{2^\ell \cdot m}$ *that is* $(q, c \cdot 2^\ell, \epsilon)$*-smooth. Furthermore, suppose that* $\mathbf{C}$ *has a decoding algorithm that uses only* k *predetermined bits out of the* ℓ

bits that it receives as answer to each query. Then there is a code $\mathbf{C}'' : \{0,1\}^n \to \{0,1\}^{t \cdot m}$ *that is* $(q, c \cdot t, \epsilon)$*-smooth, where* $t = \sum_{i=0}^{k} \binom{\ell}{i}$.

Proof: Let $x \in \{0,1\}^n$. We define $\mathbf{C}'(x)$ as follows: for every $j \in [m]$ and for every $a \in \{0,1\}^\ell$, the entry of $\mathbf{C}'(x)$ indexed by (j, a) contains the inner product between the jth (ℓ-bit long) block of $\mathbf{C}(x)$ and the (ℓ-bit long) string a. This encoding has length $m' \stackrel{\text{def}}{=} 2^\ell m$. We now describe a smooth decoding procedure for $\mathbf{C}'$.

Let A be the $(2, c, \epsilon)$-smooth decoding procedure for $\mathbf{C}$. The smooth decoding procedure A' for $\mathbf{C}'$ will first simulate A, and get two queries (j_1, j_2). If x_i is in the span of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$, then A' will reconstruct x_i as a linear combination of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$, a computation that can be done by looking at two entries of $\mathbf{C}'(x)$ (i.e., specifically the entries (j_1, a_1) and (j_2, a_2), where $x_i = \langle a_1, \mathbf{C}(x)_{j_1} \rangle + \langle a_2, \mathbf{C}(x)_{j_2} \rangle$). If x_i is not in the span of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$, then A' will output a random guess. As argued in the proof of Lemma 2.4, with probability at least 2ϵ, algorithm A (on input i) samples a pair (j_1, j_2) that is good for i (i.e., allows reconstruction with average success probability above 1/2, when averaging over all possible x's). However, whenever (j_1, j_2) is good for i, it must be the case that x_i is in the span of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$, and A' correctly reconstructs x_i. Combining these two observations, we lower bound the reconstruction probability of A' by $2\epsilon \cdot 1 + (1 - 2\epsilon) \cdot (1/2) = 1/2 + \epsilon$ (as required). Turning to the smoothness condition, observe that each entry in $\mathbf{C}'(x)$ is queried with probability at most c/m, which equals $(2^\ell \cdot c)/m'$ as required.

In order to prove the furthermore part, we do a similar construction, except that the entries of $\mathbf{C}''(x)$ correspond to pairs (j, a) where $j \in [m]$ and $a \in \{0,1\}^n$ is a vector of weight at most k. When introducing the decoding procedure A'' (for $\mathbf{C}''$), we refer not only to the queries made by A but also the the predetermined bit locations in the answer that are inspected by A. Specifically, A'' first simulate A, and get two queries (j_1, j_2) as well as two corresponding sets of bit locations $S_1, S_2 \subseteq [\ell]$. If x_i is in the span of the bit positions S_1 in $\mathbf{C}(x)_{j_1}$ and the bit positions S_2 in $\mathbf{C}(x)_{j_2}$, then A'' will reconstruct x_i as a linear combination of these bit positions, a computation that can be done by looking at two entries of $\mathbf{C}''(x)$, since $|S_1|, |S_2| \leq k$. In the analysis we note that whenever a pair of queries (made by A) is good for i, it must be the case that x_i is in the span of the bits of $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$ that are inspected by A, and A'' correctly reconstructs x_i. ■

5.2 Consequences

Combining Lemma 5.1 and Corollary 3.4, we obtain the following result.

Corollary 5.2 *Let* $\mathbf{C} : \{0,1\}^n \to (\{0,1\}^\ell)^m$ *be a* (q, c, ϵ)*-smooth linear error-correcting code. Then* $m \geq (1/2^l) \cdot 2^{\epsilon n / 4 \cdot 2^l \cdot c}$*. Furthermore, if* $\mathbf{C}$ *has a decoding algorithm that uses only* k *of the* ℓ *bits that it receives as answer to each query, then* $m \geq (1/t) \cdot 2^{\epsilon n / 4 \cdot t \cdot c}$*, where* $t = \sum_{i=0}^{k} \binom{\ell}{i}$.

Theorem 1.5 follows by combining Corollary 5.2 and Theorem 2.2.

6 Extension To Binary Linear Block Codes With Block Decoding – The Proof of Theorem 1.6

Here we deal with codes mapping $(\{0,1\}^\ell)^n$ to $(\{0,1\}^\ell)^m$, that is, mapping a sequence of n blocks, each being a string of length ℓ, to a sequence of m such blocks. We consider algorithms that recover a desired (entire) block by making two block-queries.

We focus on such codes in which the bits of each output block are a linear combination of the ℓn input bits (so indeed the $\ell = 1$ case corresponds to the main result presented in Section 3). We stress that the ℓ linear combinations corresponding to one output block are not necessarily consistent with one linear combination of the input blocks. (In case they were, this could be handled as a special case of the results presented in Section 4.)[5] We call such codes linear block-block codes.

We seek stronger bounds than the ones presented in Section 5, and we obtain them by extending Theorem 1.3. This extension is analogous to but different from the one presented in Section 5.

Recall that Theorem 1.6 states that if we let $\mathbf{C} : (\{0,1\}^\ell)^n \to (\{0,1\}^\ell)^m$ be a $(2, \delta, \epsilon)$-locally decodable code that is linear block-block, then $m \geq 2^{\frac{\epsilon\delta}{16}\cdot n-(\ell+1)^2}$.

This result is proven by an argument analogous to the one in Section 3. Here we deal with ℓn-bit long vectors, and consider queries consisting of ℓ (ℓn-dimensional) vectors over $\{0,1\}$. For every $i = 1,..,n$, we focus on pairs of queries that allow one to recover the entire ith block. Thus the 2ℓ vectors corresponding to this pair of queries must span the vectors $\vec{e}_{(i-1)\ell+j}$ for $j = 1,...,\ell$, where a sequence of vectors $\vec{v}_1,...,\vec{v}_t \in \{0,1\}^{\ell n}$ spans the vector $\vec{w} \in \{0,1\}^{\ell n}$ if for some $I \subseteq [t]$, it holds that $\oplus_{i\in I}\vec{v}_i = \vec{w}$. We say that a pair of queries spans the ith block if the 2ℓ vectors corresponding to this pair of queries span the vectors $\vec{e}_{(i-1)\ell+j}$ for $j = 1,...,\ell$. Again, the analysis reduces to providing lower bounds on the cardinality of multi-sets that contain many disjoint pairs that span each block.

Lemma 6.1 *Let $\ell \geq 2$ and n be integers, and S a multi-set of $Q \stackrel{\text{def}}{=} (\{0,1\}^{\ell n})^\ell$. For $i = 1,...,n$, let M_i be a set of disjoint pairs of elements of S that span the ith block. Then*

$$\sum_{i=1}^{n} |M_i| \leq (\ell + 2\ell^2)\cdot|S| + 2\cdot|S|\log_2|S|$$

Thus, if $\frac{1}{n}\sum_{i=1}^{n}|M_i| \geq \gamma|S|$, then $|S| \geq 2^{(\gamma n/2)-\ell^2-\ell}$.

Notations: It will be more convenient to view queries as $\ell \times \ell n$ Boolean matrices (i.e., $Q \equiv \{0,1\}^{\ell\times\ell n}$), rather than as ℓ-sequences of ℓn-dimensional vectors. Correspondingly, it is more convenient to view the recovery (or spanning) condition in matrix form: Two

[5] A sequence of ℓ vectors, $v^{(1)},...,v^{(\ell)}$, of $\{0,1\}^{\ell n}$ (i.e., ℓ linear combinations of the ℓn input bits) is consistent with one n-dimensional vector $(b_1,...,b_n) \in \{0,1\}^n$ (i.e., a linear combination of the n input blocks) if, for every $j = 1,...,\ell$, the $v^{(j)} = (b_1^{(j)},....,b_{\ell n}^{(j)})$ such that $b_k^{(j)} = b_{\lceil k/\ell\rceil}$ if $k \equiv j \pmod{\ell}$, and $b_k^{(j)} = 0$ otherwise. To see that this case is a special case of Section 4, consider the blocks as elements of the field $GF(2^\ell)$, and observe that the output symbols (i.e., the input blocks viewed as elements of $GF(2^\ell)$) are merely linear combinations (over $GF(2^\ell)$) of the input symbols (and that, furthermore, these linear combinations over the extension field $GF(2^\ell)$ are restricted to having entries in the base field $GF(2) = \{0,1\}$.

queries U and V span the ith block if there exist two $\ell \times \ell$ matrices A and B such that $AU + BV = I_i$, where I_i is the $\ell \times \ell n$ matrix that consists of $\ell \times \ell$ sub-matrices such that all but the ith sub-matrix are identically zero and the ith is the identity matrix.

6.1 Getting Rid of Singular Multiples

Motivation: *Unlike the analogous part of the proof of Lemma 4.1, here we do not modify S but rather only modify the M_i''s. Again, we wish to maintain only query pairs that allow recovery (spanning) via full-rank matrices. It can be shown that such query-pairs are few in number.*

Let S and the M_i's be as in the lemma, and E_i be the set of all pairs in S that span the ith block. Recall that $(U,V) \in E_i$ implies that there exist two $\ell \times \ell$ matrices A and B such that $AU + BV = I_i$. Let F denote the set of full rank $\ell \times \ell$ matrices. Define

$$E_i' \stackrel{\text{def}}{=} \{(U,V) \in S \times S : \exists A, B \in F \text{ s.t. } AU + BV = I_i\}, \tag{12}$$

$$M_i' \stackrel{\text{def}}{=} M_i \cap E_i'. \tag{13}$$

Claim 6.2 $\sum_{i=1}^{n} |M_i \setminus M_i'| \leq \ell |S|$.

Proof: Suppose that $(U,V) \in M_i \setminus M_i'$, and let $AU + BV = I_i$. Then either A or B is not full rank. Suppose, without loss of generality, that A is not full rank and let w be a nonzero vector such that $wA = 0$. Then $w(AU + BV) = wBV$ is a vector that is spanned by I_i, and so V spans a vector in the set $\{e_{(i-1)\ell+j} : j = 1, ..., \ell\}$. It follows that V can appear in at most ℓ pairs in $\cup_{i=1}^{n}(M_i \setminus M_i')$ (because if V appears in a pair $(U,V) \in M_i \setminus M_i'$ such that $AU + BV = I_i$ for a singular matrix A, then V spans a vector in the set $\{e_{(i-1)\ell+j} : j = 1, ..., \ell\}$). Thus each $V \in S$ contributes at most ℓ pairs to $\cup_{i=1}^{n}(M_i \setminus M_i')$, and the claim follows. ∎

6.2 Reduction To The Boolean case

Let S be a multi-set as above and define

$$S'' \stackrel{\text{def}}{=} \{\langle U, A\rangle : U \in S \,\&\, A \in F\}, \tag{14}$$

$$E_i'' \stackrel{\text{def}}{=} \{(\langle U, A\rangle, \langle V, B\rangle) \in S'' \times S'' : A, B \in F \,\&\, \exists C \in F \; AU + BV = CI_i\}. \tag{15}$$

That is, if U occurs with multiplicity m in S, then (for every A) $\langle U, A\rangle$ occurs with multiplicity m in S''. Clearly, if $(\langle U, A\rangle, \langle V, B\rangle) \in E_i''$, then $(U,V) \in E_i'$. On the other hand, if $(U,V) \in E_i'$, then there exist $A, B \in F$ such that $AU + BV = I_i$, and so for every $C \in F$ we have $(\langle U, CA\rangle, \langle V, CB\rangle) \in E_i''$. In other words, there exists $D \in F$ (i.e., $D = A^{-1}B$) such that for every $A' \in F$ we have $(\langle U, A'\rangle, \langle V, A'D\rangle) \in E_i''$.

Let M_i'' be defined as follows. For every $(U,V) \in M_i'$ and $A, B \in F$ such that $AU+BV = I_i$, and for every $C \in F$, add $(\langle U, CA\rangle, \langle V, CB\rangle)$ to M_i''.

Claim 6.3 *1. $|S''| = |F| \cdot |S|$ and $|M_i''| = |F| \cdot |M_i'|$.*

2. M_i'' is a set of disjoint pairs in E_i''.

Proof: The claim follows immediately by the definition of S'' and the M_i'''s. Specifically, for every $(U,V) \in M_i'$, there exist $A, B \in F$ such that $AU - BV = I_i$. Thus the pairs $(\langle U, CA\rangle, \langle V, CB\rangle)$ added to M_i'', for every $C \in F$, are disjoint (because $CA = C'A$ implies $C = C'$, and similarly for $CB = C'B$). ■

Claim 6.4 *Let S''' be an arbitrary subset of $Q \times F$ and M_i''' be an arbitrary set of disjoint pairs such that $(\langle U, A\rangle, \langle V, B\rangle) \in M_i'''$ implies $AU - BV = CI_i$ for some $C \in F$. Then $|S'''| \log_2 |S'''| \geq \frac{1}{2} \cdot \sum_{i=1}^{n} |M_i'''|$.*

Recall that $Q = (\{0,1\}^{\ell n})^{\ell}$, but it will be more convenient to view Q as a set of n sequences of $\ell \times \ell$ matrices, that is, $Q = (\{0,1\}^{\ell \times \ell})^n$.

Proof: The proof mimics the proof of Claim 4.4. This time, we consider a randomized mapping of $Q \times F$ to $\{0,1\}^n$. The mapping is based on a uniformly chosen 2-coloring of $M \stackrel{\text{def}}{=} \{0,1\}^{\ell \times \ell}$, denoted χ, and $\langle U, A\rangle \in Q \times F$ is mapped to $\chi(m_1) \cdot \chi(m_n)$, where $(U_1, ..., U_n) = AU$. Let us denote by $\mu_\chi : M^n \times F \to \{0,1\}^n$ the mapping induced by the 2-coloring χ, that is, $\mu_\chi(U, A) = \chi(U_1) \cdot \chi(U_n)$, where $(U_1, ..., U_n) = AU$. Thus the multi-set S''' is randomly mapped (by μ_χ) to a multi-set B_χ of $\{0,1\}^n$ such that $|B_\chi| = |S'''|$. Again, the key observation is that for every $(\langle U, A\rangle, \langle V, B\rangle) \in M_i'''$, with probability $\frac{1}{2}$, it holds that $\mu_\chi(U, A) \oplus \mu_\chi(V, B) = e_i$, and otherwise $\mu_\chi(U, A) = \mu_\chi(V, B)$. Letting $M_{i,\chi}$ denote the pairs in M_i''' that are mapped (by μ_χ) to pairs (u, v) such that $u \oplus v = e_i$, we conclude that the expected size of $M_{i,\chi}$ equals $\frac{1}{2} \cdot |M_i'''|$, where the expectation is taken uniformly over all possible χ's.

It follows that there exists a 2-coloring χ such that $\sum_{i=1}^{n} |M_{i,\chi}| \geq \frac{1}{2} \cdot \sum_{i=1}^{n} |M_i'''|$. Fixing this χ, we apply (the "furthermore" part of) Lemma 3.3 to B_χ and the $M_{i,\chi}$'s, and conclude that $|B_\chi| \log_2 |B_\chi| \geq \sum_{i=1}^{n} |M_{i,\chi}|$. Thus

$$|S'''| \log_2 |S'''| \;=\; |B_\chi| \log_2 |B_\chi| \;\geq\; \sum_{i=1}^{n} |M_{i,\chi}| \;\geq\; \frac{1}{2} \cdot \sum_{i=1}^{n} |M_i'''|.$$

■

Finishing the proof of Lemma 6.1: Using Item 2 of Claim 6.3, we may apply Claim 6.4 to S'' and the M_i'''s, and get $|S''| \log_2 |S''| \geq \frac{1}{2} \cdot \sum_{i=1}^{n} |M_i''|$. Applying the other item of Claim 6.3, we get

$$|F| \cdot |S| \log_2(|F| \cdot |S|) \geq \frac{1}{2} \cdot |F| \cdot \sum_{i=1}^{n} |M_i'|.$$

Thus $\sum_{i=1}^{n} |M_i'| \leq 2|S| \log_2(|F| \cdot |S|)$. Combining this with Claim 6.2 (and using $|F| \leq 2^{\ell^2}$), we get

$$\begin{aligned} \sum_{i=1}^{n} |M_i| &\leq \ell \cdot |S| + \sum_{i=1}^{n} |M_i'| \\ &\leq \ell \cdot |S| + 2 \cdot |S| \log_2(2^{\ell^2} \cdot |S|) \\ &\leq (\ell + 2\ell^2) \cdot |S| + 2 \cdot |S| \log_2 |S|. \end{aligned}$$

7 Lower Bounds For Private Information Retrieval – Proof of Theorem 1.8

The main result of this section is a reduction showing that a one-round PIR system can be converted to a smooth error-correcting code. This transformation preserves linearity, and hence, combined with the lower bound for smooth linear codes, yields a lower bound for linear one-round PIR systems.

7.1 Constructing Smooth Codes Based on PIR Schemes

Actually, we consider a relaxed notion of a PIR. First, recovery is not required to always be correct but rather only to be correct with probability at least $1/2 + \epsilon$, where the probability is taken over the PIR's randomization for any fixed input (i.e., a database and a desired bit). Second, we do not require perfect secrecy (i.e., $\delta = 0$), but rather that the distributions of each query for each desired bit are at pairwise statistical distance of at most δ.

Lemma 7.1 *Suppose there is a one-round, $(1 - \delta)$-secure PIR scheme with two servers, database size n, query size t, answer size a, and recovery probability at least $1/2 + \epsilon$. Then there is a $(2, 3, \epsilon - \delta)$-smooth error-correcting code $\mathbf{C} : \{0,1\}^n \rightarrow (\{0,1\}^a)^m$, where $m \leq 6 \cdot 2^t$. Furthermore:*

1. *If in the PIR scheme the answer bits are a linear combination of the data, then $\mathbf{C}$ is linear.*
2. *If, in the PIR scheme, the user only uses k predetermined bits out of the a bits it receives as an answer to each question, then the same property is true for the decoding algorithm of $\mathbf{C}$.*

Proof: Let us first develop some intuition about the proof. By enumerating all possible answers from either server, we can view the PIR system as encoding the database $x \in \{0,1\}^n$ as a string $PIR(x) \in (\{0,1\}^a)^l$, where $l = 2 \cdot 2^t$. The user can reconstruct one bit x_i of the database with advantage ϵ by looking at two entries of the encoded string $PIR(x)$. For any i and j, the distribution of the first entry read into $PIR(x)$ when reconstructing x_i is δ-close to the distribution of the first entry read into $PIR(x)$ when reconstructing x_j (and similarly for the second entry). Instead of this closeness property, we would like to have a smoothness property, that is, we would like each entry to be read with low probability. We are willing to make the encoding be slightly longer in order to achieve this goal. We will achieve this goal by duplicating entries that have a high probability of being read.

Suppose, to start, that $\delta = 0$. Then, for every j, the probability that entry j is queried by the reconstruction algorithm (as a first query or as a second query) is a fixed value p_j (independent of which bit of the database the user wants to reconstruct); note that $\sum_j p_j = 2$. We will replicate entry j of the encoding $n_j = \lceil p_j \cdot l \rceil$ times, denoting by $\mathbf{C}(x)$ this new encoding (with repetitions) of x. Recall that $PIR(x) \in (\{0,1\}^a)^l$ (and we will show that $\mathbf{C}(x) \in (\{0,1\}^a)^{O(l)}$).

A reconstruction algorithm for x_i from $\mathbf{C}(x)$ will generate queries j_1, j_2 as in the reconstruction algorithm that accesses $PIR(x)$. The algorithm then picks at random one of the n_{j_1} copies of the j_1th entry and one of the n_{j_2} copies of the j_2th entry, and then accesses these selected two entries in $\mathbf{C}(x)$. Clearly, the advantage in decoding x_i remains the same. Regarding smoothness, let us consider an entry j in $PIR(x)$. If $p_j \leq 1/l$, then the corresponding (unique) bit in $C(x)$ is accessed with probability $p_j \leq 1/l$. Otherwise (i.e., $p_j > 1/l$), the jth entry is replicated $n_j = \lceil p_j l \rceil > 1$ times, and each copy is accessed with probability p_j/n_j, which is

$$\frac{p_j}{\lceil p_j l \rceil} \leq \frac{p_j}{p_j l} = \frac{1}{l}.$$

The length of the new encoding is $m = \sum_{j=1}^{l} n_j$, and we have

$$\begin{aligned} m &= \sum_{j:p_j \leq 1/l} \lceil p_j l \rceil + \sum_{j:p_j > 1/l} \lceil p_j l \rceil \\ &\leq \sum_{j:p_j \leq 1/l} 1 + \sum_{j:p_j > 1/l} (1 + p_j l) \\ &\leq l + \sum_j p_j l \\ &= 3l \;=\; 6 \cdot 2^q. \end{aligned}$$

Recall that no entry is queried with probability higher than $1/l$, which (using $m \leq 3l$) is bounded above by $3/m$.

Consider now the general case in which the query distributions for x_{i_1} and x_{i_2} are only guaranteed to be δ-close. We apply the previously described construction using the distribution of queries for x_1. When we want to reconstruct x_i we proceed as follows. For every j, let p_j be the probability that j is queried when reconstructing x_1 and let q_j be the probability that j is queried when reconstructing x_i. Note that $\sum_j p_j = \sum_j q_j = 2$ and that $\sum_j |p_j - q_j| \leq 4\delta$, and so $\sum_{j:q_j > p_j}(q_j - p_j) \leq 2\delta$. We sample queries j_1, j_2 as in the original algorithm for x_i (modified so as to choose a random copy, if the required entry has multiple copies), and then if $q_{j_1} \leq p_{j_1}$, we proceed to make query j_1. If $q_{j_1} > p_{j_1}$, then we read query j_1 with probability p_{j_1}/q_{j_1} and we enter a "failure mode" with the remaining probability. In failure mode, bit x_i is just guessed randomly. Query j_2 is handled similarly.

Observe that the smoothness requirement is satisfied as before (since each bit corresponding to the original query j is accessed with probability $\min\{q_j, p_j\}/n_j \leq p_j/n_j \leq 1/l$). The probability of entering failure mode is $\sum_{j:q_j > p_j}(q_j - p_j) \leq 2\delta$, and when the failure mode is entered, the probability of guessing x_i correctly is exactly one half. Thus, in the worst case, failures subtract δ of the probability of guessing x_i correctly, and so the overall probability of guessing x_i right is at least $1/2 + \epsilon - \delta$. ■

7.2 Consequences

Theorem 1.8 follows by combining Lemma 7.1 and Corollary 5.2. Specifically, using $m \leq 6 \cdot 2^t$, a smoothness bound of $c = 3$ and recovery advantage $\epsilon - \delta$, we have $6 \cdot 2^t \geq \frac{1}{f(k,a)} \cdot 2^{\frac{(\epsilon-\delta)\cdot n}{4 \cdot 3 \cdot f(k,a)}}$, and Theorem 1.8 follows.

Acknowledgments

We are grateful to Alex Samorodnitsky for suggesting to us (and allowing us to include) the information-theoretic proof of Lemma 3.3 which is presented in Section 3.4.

References

[1] A. Ambainis. An Upper Bound On The Communication Complexity of Private Information Retrieval. In *24th ICALP*, Springer, Lecture Notes in Computer Science, Vol. 1256, pages 401–407, 1997.

[2] B. Bollobás. *Combinatorics.* Cambridge University Press, 1986.

[3] B. Chor and N. Gilboa. Computationally-Private Information Retrieval. In *29th STOC*, pp. 304–313, 1997.

[4] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan, Private Information Retrieval. *Journal of the ACM*, Vol. 45, No. 6, pages 965–982, November 1998.

[5] J. Katz and L. Trevisan. On The Efficiency Of Local Decoding Procedures For Error-Correcting Codes. In *32nd STOC*, 2000.

[6] E. Kushilevitz and R. Ostrovsky. Replication is Not Needed: Single Database, Computationally-Private Information Retrieval. In *38th FOCS*, pages 364–373, 1997.