Lower Bounds for Linear Locally Decodable Codes and Private Information Retrieval*

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

July 7, 2005

Abstract

We prove that if a linear error-correcting code $\mathbf{C} : \{0,1\}^n \rightarrow \{0,1\}^m$ 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.

*A preliminary version of this work has appeared in the proceedings of *17th IEEE Conference on Computational Complexity*, pages 175–183, 2002.

†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, MC256-80, Pasadena CA 91125, USA. E-mail: schulman@caltech.edu. Partially supported by NSF Career grant 0049092, the Charles Lee Powell Foundation, and an Okawa Foundation Grant.

¶Computer Science Division, University of California, Berkeley. E-mail: luca@eecs.berkeley.edu. Supported by an NSF Career award CCR-9984703, a Sloan Research Fellowship, and an Okawa Foundation Grant.

Contents

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 [9], 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 recovery based on reading k bits [3] have length $2^{n^{\Theta(\log\log k/(k \log k))}}$, which is of the form $2^{n^{\Omega(1)}}$ for every constant k. 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. We note that all the known constructions of LDCs are linear codes, including constructions where the decoder makes more than two queries. See [13] for a survey on LDCs and their applications.

The study of PIR schemes was initiated by Chor, Goldreich, Kushilevitz and Sudan [6], 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. Again, we note that in all the known constructions of PIR schemes the server's answers are linear combination of the data bits. This is also true of schemes involving more than two servers. See [8] for a survey on PIR schemes.

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. That is, if the code is binary, then only two *bits* of the corrupted codeword are read. The procedure should predict each bit with a constant advantage even when there is a constant fraction of errors in the 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 a natural number n, we let $[n] \stackrel{\text{def}}{=} \{1, ..., n\}$. For $x \in \Sigma^m$ and $i \in [m]$, 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). Query $i \in [m]$ to the oracle $y \in \Gamma^m$ is answered by y_i.*

- *For every $x \in \Sigma^n$, for every $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$, and for every $i \in [n]$, we have*

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

  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 $\mathbf{C}$.*

Notice that, for small ϵ and large Σ, the definition is very weak: the decoding algorithm is only guaranteed to recover the requested entry with some advantage over a random guess. The correct answer may not even be the one with the largest probability of being output, and so the correctness probability cannot be amplified by running the algorithm several times and taking the most occurring answer. In order to prove a lower bound, it is of course desirable to state the weakest possible definition.

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 ([9]) *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 Theorem 1.3 (and in Section 3) we deal with the special case $\Sigma = \Gamma = GF(2)$, while in Theorem 1.4 (and 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 Theorem 1.5 (and in Section 5).
- $\Sigma = \Gamma = \{0, 1\}^l$, and $\mathbf{C} : \{0, 1\}^{ln} \to \{0, 1\}^{lm}$ is linear. That is, we consider codes mapping a sequence of n blocks, each being a string of length l, to a sequence of m such blocks, and algorithms that recover a desired (entire) block by making two block-queries. We refer to such codes as block-block codes, and deal with them in Theorem 1.6 (and in Section 6).

Our main result is

Theorem 1.3 *Let $\Sigma = \Gamma = \{0, 1\}$, let $\mathbf{C} : \Sigma^n \to \Gamma^m$ be a $(2, \delta, \epsilon)$-locally decodable linear code, and suppose that $n \geq 8/\epsilon\delta$. Then $m \geq 2^{\epsilon\delta n/4}$.*

In comparison, the Hadamard code, which is linear, is a $(2, \delta, 1/2 - 2\delta)$-locally decodable code for every $\delta < 1/4$, and its encoding length is $m = 2^n$. In Section 8 we show that linear $(2, \delta, \epsilon)$ codes exist with $m = 2^{O(\delta n/(1-2\epsilon)}$.

Theorem 1.3 has the following extensions to larger alphabets (corresponding to the three cases discussed above). First, we consider an extension to linear codes over arbitrary finite fields.

Theorem 1.4 *Let* $\mathbf{C} : F^n \to F^m$ *be a* $(2, \delta, \epsilon)$*-locally decodable linear code. Then* $m \geq 2^{\frac{\epsilon\delta}{8} \cdot n - 2 - \log_2 |F|}$.

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* $m \geq (1/f(k,l)) \cdot 2^{\epsilon\delta n/(4f(k,l))}$*, where* $f(k,l) = \sum_{i=0}^{k} \binom{l}{i} \leq \min\{2^l, 2l^k\}$*, provided that* $n \geq 8f(k,l)/\epsilon\delta$.

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 a linear block-block code. Then* $m \geq 2^{\frac{\epsilon\delta}{8} \cdot n - (\ell+1)2}$.

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 send 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, which responds also with a single message. All known efficient PIR schemes are one-round. 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 it 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* 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 (i.e., the statistical difference between them is at most* δ*).*

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 which have *linear answers*; that is, for every fixed query $q \in \{0,1\}^t$, the servers' answers $S_1(x,q)$ and $S_2(x,q)$ are linear functions of x (each bit of $S_1(x,q)$ and each bit of $S_2(x,q)$ is a linear combination of the bits of x). All known PIR schemes satisfy this requirement.

Our main result for PIR schemes is the following lower bound, proved in Section 7

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 only uses k predetermined bits out of the a bits it receives as answer to each query. Then*

$$t > \frac{(\epsilon - \delta)\cdot n}{6 \cdot f(k,a)} - \log_2 f(k,a) - 3,$$

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 [6, Sec. 5.2].)

- Any (secure, one-round) 2-server PIR scheme with linear answers in which the user only uses 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 only uses k bits from each answer, k a constant, must have communication complexity $\Omega(n^{1/(k+1)})$.

In one of the PIR schemes of Chor et al. [6], 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., [6, Sec. 3.1]).
- There exists a (perfectly secure, one-round) 2-server, linear-answer PIR scheme in which the user uses only one bit from each $\sqrt{n}$ bit-long answer, and the queries are also $\sqrt{n}$-bit long strings (e.g., by a minor modification of the scheme in [6, Sec. 3.2–3.3] as applied to $d = 2$).

Regarding 2-server schemes, Chor et al. [6] present a scheme where a, k and t are $O(n^{1/3})$. Our results do not yield any lower bound in this setting. In particular, it is compatible with current knowledge (but considered very unlikely) that a 2-server scheme exists in which a, k and t are all $O(\log n)$.

The best q-server PIR construction, for large q, is due to Beimel et al. [3], and it achieves $a = 1$ and $t = n^{\log q/(q \log\log q)}$.

All known constructions, including those in [6, 3] are linear and one-round.

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 [5]. 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 [6, 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 [6, Sec. 5.1], whereas communication complexity of $O(n^\epsilon)$ can be achieved for computationally-secure PIR's [10], assuming the intractability of the quadratic residuosity problem.)

1.3 Tightness of Our Bounds

In Section 8 we outline some constructions that give upper bounds which are quite close to our lower bounds for the case of binary codes (or PIR with answer size one).

Specifically, we present the following constructions:

- For every $0 < \epsilon \leq 1/2$, $c \geq 2$ and for sufficiently large n, a $(2, c, \epsilon)$ smooth linear code with encoding length $2^{O(\epsilon n/c)}$. (See Section 2 for a definition of smooth code.) This construction perfectly matches our $2^{\Omega(\epsilon n/c)}$ lower bound that we prove as an intermediate step for our lower bounds for codes and PIR schemes.

- For every $0 < \epsilon \leq 1/2$, $0 < \delta < 1/4$ and sufficiently large n, a $(2, \delta, \epsilon)$ locally decodable linear code with encoding length $2^{O(\delta n/(1-2\epsilon))}$. Our lower bound is $2^{\Omega(\epsilon\delta n)}$.

- For every $0 < \epsilon < 1/2$, $0 < \delta < 2\epsilon$, a $(1 - \delta)$-secure 2-round linear PIR with query size $O(n(2\epsilon - \delta))$ and answer size 1. Our lower bound is $\Omega(n(\epsilon - \delta))$.

1.4 Subsequent Work

Following the preliminary publication of our results, Obata [12] improves our lower bound for binary linear smooth codes to $2^{\Omega(\delta n/(1-2\epsilon))}$, for every $\delta > 0$, $\epsilon < 1/2$. (Note that in a locally decodable code that corrects up to a δ fraction of errors, the reconstruction probability cannot be arbitrarily close to 1.) In light of the construction that we mentioned above, this lower bound is tight in all parameters.

One central question left open by our work refers to general (rather than linear) binary codes. Specifically, does the exponential lower-bound on the length of binary linear codes that support 2-query decodability extend to *general* (i.e., non-linear) codes? This question has been resolved recently by Kerenidis and de Wolf [11], who proved a $2^{\Omega(\mathrm{poly}(\epsilon,\delta)\cdot n)}$ lower-bound on the length of *any* $(2, \delta, \epsilon)$ locally decodable binary code. The results of Kerenidis and de Wolf also apply to 2-server, 2-round, $(1 - \delta)$-secure PIR with answer size k and recovery probability $1 - \epsilon$, giving a communication lower bound of $\Omega(\mathrm{poly}(\epsilon, \delta, 2^{-k})$ for such schemes. We note that the dependency of their bound on ϵ and δ is worse than in our results. Interestingly, the lower bound of [11] relies on quantum information theory. For special the case of private information retrieval with 1-bit answers and recovery probability 1, the lower-bound of [11] was subsequently improved by Beigel *et al.* [2] (using a simpler "classical" argument).

Our lower-bound for the case of general fields (i.e., Theorem 1.4) was recently improved by Dvir and Shpilka [7], who removed the dependency of the lower bound on the field size. We conjecture that a similar improvement is possible for Theorem 1.6.

1.5 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 [9]) 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 [9]. 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 $n - l2$.

In Section 8 we consider the tightness of some of our lower-bounds.

2 Preliminaries

The notions and results in this section are mostly due to Katz and Trevisan [9]. 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 fixed 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 non-adaptively.*

- *For every $x \in \{0,1\}^n$ and for every $i \in [n]$, we have*

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

- *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 value, 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 [9]) *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.*

This is stated only for the case $\Sigma = \{0,1\}$ in [9], but the proof applies to the general case as well. A weak converse also holds, namely, if $\mathbf{C}$ is a (q, c, ϵ)-smooth code, then $\mathbf{C}$ is also $(q, \delta, \epsilon - q\delta)$-locally decodable, for every $\delta < \epsilon/q$.

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 which A reads in that invocation is exactly $\{q_1, q_2\}$. We say that $\{q_1, q_2\}$ is good for i if there is a non-zero probability that A reads $\{q_1, q_2\}$ and

$$\mathbf{Pr}[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ reads } \{q_1, q_2\}] > \frac{1}{|\Sigma|},$$

where the probability is taken over x uniformly chosen from $\{0,1\}^n$, and over the internal coin tosses of A. This may seem a very weak property, but we can derive interesting consequences from it when $\mathbf{C}$ is a linear code. When $\mathbf{C}$ is linear, the value of x_i can either be deduced as a linear combination of the entries q_1 and q_2 of $\mathbf{C}(x)$, or it is linearly independent from them. In the latter case, the pair (q_1, q_2) cannot possibly be good for i, because, for a random x, the value x_i is a random variable that is statistically independent of the entries q_1 and q_2 of $\mathbf{C}(x)$. Therefore, if (q_1, q_2) is good for i, and $\mathbf{C}$ is linear, it follows that x_i can be deduced without errors by looking at the entries q_1 and q_2 of $\mathbf{C}(x)$.

For every $i \in [n]$, we consider the graph with edge set consisting of the set of good pairs. We call this graph the *recovery graph* for i. We may assume without loss of generality that the decoding procedure makes two distinct queries, and so the recovery graph has no self-loop.

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 ([9]) *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 $\epsilon m|\Sigma|/(2c \cdot (|\Sigma| - 1))$.*

This is essentially Lemma 4 in [9], but, since we slightly changed the definition of the recovery graph (from [9]), 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 $j_1, j_2 \in [m]$,

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

where the probability is taken uniformly over $x \in \Sigma^n$ and A's internal coin tosses. (For example, we can modify A so that it outputs a random element of Σ whenever $i \in [n]$ and $j_1, j_2 \in [m]$ do not satisfy Eq. (1).) It follows from Markov's inequality that, with probability at least $|\Sigma|\epsilon/(|\Sigma| - 1)$, on input $i \in [n]$, algorithm A generates a pair that is good for i. In other words, with probability at least $|\Sigma|\epsilon/(|\Sigma| - 1)$, 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 $|\Sigma|\epsilon/(|\Sigma| - 1)$. On the other hand, no element of $[m]$ is queried by A with probability greater than c/m, and so it follows that $|C| \geq (|\Sigma|\epsilon/(|\Sigma| - 1)/(c/m) = |\Sigma|\epsilon m/(|\Sigma| - 1)c$. 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 $|\Sigma|\epsilon m/2(|\Sigma| - 1)c$. ■

3 The Boolean Case – 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 (always) equal to a particular bit of the input. We can accommodate this condition essentially by removing the bits of the input that are identical to too many bits in the output. This gives the following lemma, which is stated and proven here only for the case of linear codes.[1]

Lemma 3.1 *For $n > 4c/\epsilon$ and $m < 2^{n/2}/n$, let $\mathbf{C} : \{0,1\}^n \to \{0,1\}^m$ be a linear (q, c, ϵ)-smooth code. Then, for some $n' \geq n/2$, there is a linear code $\mathbf{C}' : \{0,1\}^{n'} \to \{0,1\}^{2m}$ that has a $(q, c, \epsilon/2)$-smooth reconstruction procedure such that each of the output bits of $\mathbf{C}'$ is neither identically zero nor equal to a single input bit.*

[1] We conjecture that a similar lemma hold for general codes.

Thus lower bounds on the length of smooth codes satisfying the conclusion of the lemma yield lower bounds on general smooth codes. Needless to say, if $m \geq 2^{n/2}/n$ then we are done anyhow.

Proof: We say that the output position $j \in [m]$ is a projection of the input position $i \in [n]$ if it holds that $\mathbf{C}(x)_j = x_i$ for every $x \in \{0,1\}^n$. We denote by P_i the set of output positions that are projections of the input position $i \in [n]$, and consider the set I of locations in the input that occur in more than a fraction $2/n$ of the bits of the output; that is, $I \stackrel{\text{def}}{=} \{i \in [n] : |P_i| \geq 2m/n\}$. Clearly, $|I| \leq n/2$. On the other hand, for each $i \in [n] \setminus I$, on input i, the reconstruction procedure queries a location in P_i with probability at most $2c/n$, which is less than $\epsilon/2$ (provided that $n > 4c/\epsilon$). Thus, if we modify $\mathbf{C}$ in locations $\cup_{i\in[n]\setminus I} P_i$, then we may decrease the recovery probability by at most $\epsilon/2$, so the recovery condition is met.

Without loss of generality, suppose that $I = \{n'+1, ..., n\}$, where $n' \geq n/2$. We construct the code $\mathbf{C}' : \{0,1\}^{n'} \to \{0,1\}^m$ by replacing the values of all projected output bits that correspond to inputs in $[n']$ by $x_1 + x_2$, and replacing each input bit x_i for $i \in I$ by some function $f_i(x_1, ..., x_{n'})$ to be determined later. That is, $\mathbf{C}'(x_1, ..., x_{n'})_j = x_1 + x + 2$ for $j \in \cup_{i\in[n']} P_i$, and $\mathbf{C}'(x_1, ..., x_{n'})_j = \mathbf{C}(x_1, ..., x_{n'}, f_{n'+1}(x'), ..., f_n(x'))_j$ otherwise, where $x' = (x_1, ..., x_{n'})$. Note that $\mathbf{C}'$ essentially maintains the decodability properties of the original n' variables; that is, the recovery algorithm of $\mathbf{C}$ recovers each of the bits of $\mathbf{C}'$ with probability at least $(1+\epsilon)/2$.

Recall that we need to show that each of the output bits of $\mathbf{C}'$ is neither identically zero nor equal to a single input bit. This is obvious for $j \in \cup_{i\in[n']} P_i$, and showing it for the other j's requires an adequate choice of the functions f_i's (for $i \in I$). This yields $n'+1$ linear inequalities for each of the m output bits, yielding a system on $(n'+1)m$ inequalities in the formal variables $x_1, ..., x_{n'}$ and the undetermined linear functions $f_{n'+1}, ..., f_n$. Using the probabilistic method and $(n'+1)m < 2^{n'}$, it follows that there exists a choice of these functions such that all $(n'+1)m$ inequalities are satisfied (as formal inequalities between linear expressions in the formal variables $x_1, ..., x_{n'}$).[2] The lemma follows. ■

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 with $\mathbf{C}'$ 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 is not possible to gain any advantage in predicting x_i by looking at the q_1-th 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

[2] Specifically, selecting each f_i uniformly among all possible $2^{n'}$ linear functions, each inequality is violated with probability at most $2^{-n'}$. Thus, a random choice of these functions satisfies all inequalities with probability at least $1 - (n'+1)m \cdot 2^{-n'} > 0$.

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 $a_1, \ldots, a_m$ be a sequence of* (not necessarily distinct) *elements of $\{0,1\}^n$ such that for every $i \in [n]$ there is a set M_i of disjoint pairs of indices $\{j_1, j_2\}$ such that $e_i = a_{j_1} \oplus a_{j_2}$. Then $m \geq 2^{2\gamma n}$, where $\gamma \stackrel{\text{def}}{=} \sum_{i=1}^{n} |M_i|/nm$.*

Indeed, a special case of interest is where $|M_i| \geq \gamma m$, for each i. Below, we will present two alternative proofs of Lemma 3.3 (the first being "combinatorial" and the second "information theoretic"). 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, and suppose that $n \geq 4c/\epsilon$. Then $m \geq 2^{\epsilon n/(2c)}$.*

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

Proof: Ignoring the case of $m \geq 2^{n/2}/n$, 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 $\epsilon' = \epsilon/2$ such that no bit in the codeword equals a bit of the plaintext. 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 $m' \geq 2^{2\epsilon' n'/c} \geq 2^{\epsilon n/(2c)}$, and using $m \geq m'$ the claim follows. ■

For sake of future reference, we also state the following direct corollary to Lemma 3.3:

Corollary 3.5 *Let $a_1, \ldots, a_m$ and $M_1, \ldots, M_n$ be as in Lemma 3.3. Then, $m \log_2 m \geq 2 \cdot \sum_{i=1}^{n} |M_i|$.*

Proof: Just let $\gamma \stackrel{\text{def}}{=} \frac{1}{nm} \sum_{i=1}^{n} |M_i|$, and apply Lemma 3.3 (which yields $\log_2 m \geq 2\gamma n$). ■

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 3.6 (See Appendix) *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 $\frac{1}{2}|S| \log_2 |S|$.*

Let us see that in this special case, Lemma 3.3 follows. Recall that the aforementioned (distinct) vectors $a_1, \ldots, a_m$ are all vertices of a hypercube, and the hypothesis (of Lemma 3.3) implies that the total number of edges between these vertices is at least γmn. But Lemma 3.6 implies that this number it at most $\frac{1}{2} m \log_2 m$, and so it follows that $m \geq 2^{2\gamma n}$.

[3] We note that the lower-bound of Lemma 3.3 is tight and implies Lemma 3.6 as a special case. Specifically, in the special case, the set of edges $E(S,S)$ with both endpoints in S can be partitioned into matchings M_i's as in Lemma 3.3. Letting $\gamma = (\sum_i |M_i|)/(n|S|)$, and applying Lemma 3.3, we get $|S| \geq 2^{2\gamma n} = 2^{2\sum_i |M_i|/|S|}$. Thus, $\log_2 |S| \geq 2|E(S,S)|/|S|$, which implies $|E(S,S)| \leq (1/2)|S| \log_2 |S|$.

To complete the proof of Lemma 3.3, we have to consider the case in which $a_1, \ldots, a_m$ are not all different. Note that an analogue of Lemma 3.6 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]

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:

$$|M_i| \leq \sum_{a,b : a \oplus b = e_i} \min\{\nu_a, \nu_b\} = \frac{1}{2} \sum_a \min\{\nu_a, \nu_{a \oplus e_i}\} \tag{5}$$

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 that 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 first part of the bound of (5). Notice that the sum is over all *unordered* pairs $\{a, b\}$, so that if we enumerate all ordered pairs of the form $(a, a \oplus e_i)$ we are actually counting each unordered pair twice. This explains the factor 1/2 in the second part of (5).

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

$$\begin{aligned}
\gamma m n &= \sum_{i=1}^n |M_i| \\
&\leq \frac{1}{2} \sum_{i=1}^n \sum_{a \in \{0,1\}^n} \min\{\nu_a, \nu_{a \oplus e_i}\} \\
&= \frac{1}{2} \sum_{i=1}^n \sum_{a \in \{0,1\}^n} \sum_{k \geq 1} \chi(\nu_a, k)\chi(\nu_{a \oplus e_i}, k)
\end{aligned}$$

[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, Lemma 3.3 implies $m \geq 2^{2\gamma n}$, for $\gamma \leq 1/nm \leq 1/2n$, which indeed holds (because $m \geq 2$).

and so

$$\gamma mn \;\le\; \frac{1}{2}\sum_{k\ge 1}\sum_{i=1}^{n}\sum_{a\in\{0,1\}^n}\chi(\nu_a,k)\chi(\nu_{a\oplus e_i},k). \qquad (6)$$

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 3.6, we have, for every k, that

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

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

$$\gamma mn \le \frac{1}{2}\sum_k s_k\cdot\log_2 m = \frac{1}{2}m\cdot\log_2 m,$$

from which it follows that $m \ge 2^{2\gamma n}$. This completes the proof of Lemma 3.3.

3.4 An Information-Theoretic Proof of Lemma 3.3

The "information-theoretic" 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 J be an integer chosen uniformly at random from $\{1,2,...,m\}$ and let $X = a_J$. 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, $H(X)\le\log_2 m$. On the other hand, we will prove that $H(X)\ge 2\gamma n$, and $m\ge 2^{2\gamma n}$ will follow immediately.

We can express the entropy of X as

$$H(X) = H(X_1) + H(X_2|X_1) + \cdots + H(X_n|X_1\cdots X_{n-1}). \qquad (7)$$

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:

$$H(X_i|X_{1,i-1}) = \sum_{b\in\{0,1\}^{i-1}}\mathbf{Pr}[X_{1,i-1}=b]\cdot H(X_i|X_{1,i-1}=b). \qquad (8)$$

Observe that for any 0-1 random variable Y with $p \stackrel{\text{def}}{=} \mathbf{Pr}(Y=1)$ (in our case $Y = (X_i|X_{1,i-1}{=}b)$), we have $H(Y) = H_2(p)$, where $H_2(x) = x\log_2(1/x) + (1-x)\log_2(1/(1-x)) \ge 2\cdot\min\{x,1-x\}$ is the binary entropy function.[5] So Eq. (8) is at least

$$\sum_{b\in\{0,1\}^{i-1}}\mathbf{Pr}[X_{1,i-1}=b]\cdot 2\min\{\mathbf{Pr}[X_i=0|X_{1,i-1}=b],\mathbf{Pr}[X_i=1|X_{1,i-1}=b]\}. \qquad (9)$$

Let us say $X = a_J$ *is an endpoint of an edge of* M_i if J is any one of the $2|M_i|$ indices in the pairs of M_i. Now, conditioning on the values of any bits other than the ith, the probability that X is an

[5] We claim that, for $x\in[0,0.5]$, it holds that $H_2(x)\ge 2x$ (whereas a bound of $H_2(x)\ge x$ is obvious). The claim can be verified by noting that $f(x)\stackrel{\text{def}}{=}H_2(x)-2x$ is convex in that interval, and that $f(0)=0=f(1/2)$.

endpoint of an edge of M_i equals the sum, over $\sigma \in \{0,1\}$, of the probability that $X_i = \sigma$ and X is an endpoint of an edge of M_i, under the same conditioning. We prove below that

$$\begin{aligned} &\mathbf{Pr}[X_i = \sigma \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ &\quad \le \min\{\mathbf{Pr}[X_i = 0 | cond], \mathbf{Pr}[X_i = 1 | cond]\} \end{aligned} \tag{10}$$

for $\sigma = 0,1$. Applying inequality (10) for $\sigma = 0,1$, we have

$$\begin{aligned} &\sum_{\sigma=0}^{1} \mathbf{Pr}[X_i = \sigma \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ &\quad \le 2\min\{\mathbf{Pr}[X_i = 0 | cond], \mathbf{Pr}[X_i = 1 | cond]\}, \end{aligned}$$

and therefore

$$\begin{aligned} &\mathbf{Pr}[X \text{ is an endpoint of an edge of } M_i | cond] \\ &\le 2\min\{\mathbf{Pr}[X_i = 0 | cond], \mathbf{Pr}[X_i = 1 | cond]\}. \end{aligned} \tag{11}$$

Now we prove inequality (10). We use the fact that the conditioning *cond* doesn't involve bit i of X. Given an edge e of M_i, either both endpoints satisfy condition *cond* or neither. Hence

$$\begin{aligned} &\mathbf{Pr}[X_i = 0 \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ &= \mathbf{Pr}[X_i = 1 \text{ and } X \text{ is an endpoint of an edge of } M_i | cond]. \end{aligned} \tag{12}$$

Clearly

$$\begin{aligned} & \mathbf{Pr}[X_i = 0 \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ \le\ & \mathbf{Pr}[X_i = 0 | cond] \end{aligned}$$

and

$$\begin{aligned} & \mathbf{Pr}[X_i = 1 \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ \le\ & \mathbf{Pr}[X_i = 1 | cond]. \end{aligned}$$

By equation (12), for $\sigma \in \{0,1\}$ we have

$$\begin{aligned} & \mathbf{Pr}[X_i = \sigma \text{ and } X \text{ is an endpoint of an edge of } M_i | cond] \\ \le\ & \min\{\mathbf{Pr}[X_i = 0 | cond], \mathbf{Pr}[X_i = 1 | cond]\}. \end{aligned}$$

This is inequality (10).

By inequality (11) with *cond* replaced by "$X_{1,i-1} = b$" for varying b, expression (9) and hence $H(X_i | X_{1,i-1})$ are bounded below by

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

Therefore, by equation (7),

$$H(X) \ge \sum_{i=1}^{n} \frac{2|M_i|}{m} = \frac{2}{m} \cdot \gamma m n = 2\gamma n.$$

Recalling that $H(X) \le \log m$, we obtain $m \ge 2^{2\gamma n}$, and establish Lemma 3.3.

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. Our aim here is to establish Theorem 1.4, which asserts that if we let $\mathbf{C} : F^n \to F^m$ be a $(2, \delta, \epsilon)$-locally decodable linear code, then $m \geq 2^{\frac{\epsilon\delta}{8} \cdot n - 2 - \log_2 |F|}$.

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 $\vec{e}_i$. Indeed the technical contents of this section is captured by the following lemma.

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*

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

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

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 Lemma 4.1, 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

$$S' \stackrel{\text{def}}{=} S \setminus \{\alpha\vec{e}_i : \alpha \in F \,\&\, i = 1, ..., n\} \tag{13}$$

$$E'_i \stackrel{\text{def}}{=} E_i \cap (S' \times S') \tag{14}$$

$$M'_i \stackrel{\text{def}}{=} M_i \cap (S' \times S'). \tag{15}$$

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* i*th entry of the message 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

$$S'' \stackrel{\text{def}}{=} \{\langle\vec{u}, \alpha\rangle : \vec{u} \in S' \,\&\, \alpha \in F \setminus \{0\}\}, \tag{16}$$

$$E''_i \stackrel{\text{def}}{=} \{(\langle\vec{u}, \alpha\rangle, \langle\vec{v}, \beta\rangle) \in S'' \times S'' : \exists\gamma \in F \setminus \{0\} \text{ s.t. } \alpha\vec{u} - \beta\vec{v} = \gamma\vec{e}_i\}. \tag{17}$$

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* i*th coordinate (and they differ on their* i*th 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* $|S'''| \log_2 |S'''| \geq \sum_{i=1}^{n} |M'''_i|$.

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, $\mu_\chi(\vec{u},\alpha) = \chi(v_1)\cdots\chi(v_n)$, 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 Corollary 3.5 to B_χ and the $M_{i,\chi}$'s, and conclude that $|B_\chi|\log_2|B_\chi| \geq 2 \cdot \sum_{i=1}^n |M_{i,\chi}|$. Thus

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

■

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 $|S''|\log_2|S''| \geq \sum_{i=1}^n |M_i''|$. Applying the other items of Claim 4.3, we get

$$\begin{aligned}
(|F|-1)\cdot|S'|\log_2(|F|\cdot|S'|) &> (|F|-1)\cdot|S'|\cdot\log_2((|F|-1)|S'|) \\
&= |S''|\cdot\log_2|S''| \\
&\geq \sum_{i=1}^n |M_i''| \\
&= (|F|-1)\cdot\sum_{i=1}^n |M_i'|.
\end{aligned}$$

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

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).

There are two cases to be considered, depending on whether or not x_i can be reconstructed as a linear combination of the 2ℓ bits $\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}$.

1. If x_i can be reconstructed as a linear combination of the bits $\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}$, then this means that there are vectors $a_1, a_2 \in \{0,1\}^\ell$ such that $x_i = a_1 \cdot \mathbf{C}(x)_{j_1} \oplus a_2 \cdot \mathbf{C}(x)_{j_2}$. (We use the notation $a \cdot b$ to denote the inner product of the vectors a and b.) In this case, algorithm A' can reconstruct x_i by looking at two bits of $\mathbf{C}'(x)$, that is, the entries (j_1, a_1) and (j_2, a_2).

2. If x_i cannot be reconstructed as a linear combination the bits $\mathbf{C}(x)_{j_1}, \mathbf{C}(x)_{j_2}$, then, for a random x, the random variable x_i is independent of the random variables $\mathbf{C}(x)_{j_1}$ and $\mathbf{C}(x)_{j_2}$. In this case, algorithm A' outputs 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, we are in case (1) and A' correctly reconstructs x_i. Combining these two observations, we bound the reconstruction probability of A' below 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 simulates A, and gets two queries (j_1, j_2) as well as two corresponding sets of bit locations $S_1, S_2 \subseteq [\ell]$. If x_i can be reconstructed as a linear combination 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 using such a linear combination, 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 (j_1, j_2) (made by A) is good for i, it must be the case that A'' correctly reconstructs x_i when (j_1, j_2) are the queries selected in the simulation step. ■

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* $m \geq (1/2^l) \cdot 2^{\epsilon n/2 \cdot 2^l \cdot c}$*, provided that* $n \geq 2^{l+2}c/\epsilon$*. 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* $m \geq (1/t) \cdot 2^{\epsilon n/2 \cdot t \cdot c}$*, where* $t = \sum_{i=0}^{k} \binom{\ell}{i}$*, provided that* $n \geq 4ct/\epsilon$*.*

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.)[6] 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 4. Our aim here is to establish Theorem 1.6, which asserts 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 $m \geq 2^{\frac{\epsilon\delta}{8} \cdot n - (\ell+1)2}$.

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. Indeed the technical contents of this section is captured by the following lemma.

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* i*th block. Then*

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

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

[6] 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 $GF(2) = \{0,1\}$.

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 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 Lemma 6.1, 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

$$E_i' \stackrel{\text{def}}{=} \{(U,V) \in S \times S : \exists A, B \in F \text{ s.t. } AU + BV = I_i\}, \tag{18}$$

$$M_i' \stackrel{\text{def}}{=} M_i \cap E_i'. \tag{19}$$

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

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 $\{e_{(i-1)\ell+j} : j = 1, ..., \ell\}$. Consider now the set of indices i such that V belongs to a pair in $M_i \setminus M_i'$; for each such index i, V spans a vector in the set $\{e_{(i-1)\ell+j} : j = 1, ..., \ell\}$, and for different indices i the sets $\{e_{(i-1)\ell+j} : j = 1, ..., \ell\}$ are disjoint, and their elements are all linearly independent. Considering that V can span at most ℓ linearly independent vectors, we conclude that there are at most ℓ indices i such that V belongs to a pair in $M_i \setminus M_i'$. 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

$$S'' \stackrel{\text{def}}{=} \{\langle U, A\rangle : U \in S \,\&\, A \in F\}, \tag{20}$$

$$E_i'' \stackrel{\text{def}}{=} \{(\langle U, A\rangle, \langle V, B\rangle) \in S'' \times S'' :\ A, B \in F \,\&\, \exists C \in F\ AU + BV = CI_i\}. \tag{21}$$

That is, if U occurs with multiplicity t in S, then (for every A) $\langle U, A\rangle$ occurs with multiplicity t 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 $|S'''| \log_2 |S'''| \geq \sum_{i=1}^n |M'''_i|$.*

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 $M \stackrel{\rm def}{=} \{0,1\}^{\ell \times \ell}$, denoted χ, and $\langle U, A\rangle \in Q \times F$ is mapped to $\chi(U_1) \cdots \chi(U_n)$, 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, $\mu_\chi(U, A) = \chi(U_1) \cdots \chi(U_n)$, 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 $\mu_\chi(U, A) = \mu_\chi(V, B)$. To justify the key observation, let $(U_1, \ldots, U_n) = AU$ and $(V_1, \ldots, V_n) = BV$. Then, $U_j = V_j$ for $j \neq i$, and $U_i \neq V_i$. For every choice of the coloring χ, we have $\chi(U_j) = \chi(V_j)$. With probability $\frac{1}{2}$, we have $\chi(U_i) = \chi(V_i)$, and with probability $\frac{1}{2}$ we have $\chi(U_i) \neq \chi(V_i)$. The first case gives $\mu_\chi(U, A) = \mu_\chi(V, B)$, and the second case gives $\mu_\chi(U, A) \oplus \mu_\chi(V, B) = e_i$. 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 Corollary 3.5 to B_χ and the $M_{i,\chi}$'s, and conclude that $|B_\chi| \log_2 |B_\chi| \geq 2 \cdot \sum_{i=1}^n |M_{i,\chi}|$. Thus

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

■

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 $|S''| \log_2 |S''| \geq \sum_{i=1}^n |M''_i|$. Applying the other item of Claim 6.3, we get

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

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

$$\sum_{i=1}^n |M_i| \;\leq\; \ell \cdot |S| + \sum_{i=1}^n |M'_i|$$

$$\leq \ell \cdot |S| + |S| \log_2(2^{\ell 2} \cdot |S|)$$
$$\leq (\ell + \ell 2) \cdot |S| + |S| \log_2 |S|.$$

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 into 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 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 $\mathbf{C} : \{0,1\}^n \to (\{0,1\}^a)^m$, 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

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

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 the 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 $6 \cdot 2^t \geq \frac{1}{f(k,a)} \cdot 2^{\frac{(\epsilon-\delta)\cdot n}{2\cdot 3 \cdot f(k,a)}}$, and Theorem 1.8 follows.[7]

[7]Note that, in order to use Corollary 5.2, we need to assume $n \geq 12 f(k,a)/(\epsilon - \delta)$. If the condition is not satisfied, however, the conclusion of Theorem 1.8 is still true, because it reduces to the trivial statement that t is larger than a negative number.

8 Tightness of our Bounds

In this section we show that, for the case of binary alphabets, our bounds are almost tight in their dependency on all parameters. Some of the bounds presented in this section also appear in [12], credited to Trevisan.

We begin by recalling that the Hadamard code is a locally decodable and smooth code and how it gives a private information retrieval system.

The Hadamard code is a linear code $C : \{0,1\}^n \to \{0,1\}^{2^n}$. We index the 2^n entries of a codeword by using elements of $\{0,1\}^n$ instead of using integers in $[2^n]$, and for a message x and an entry a we have $C(x)[a] = x \cdot a$, where the dot product $x \cdot a$ is defined as $\sum_i x_i a_i \pmod 2$.

Such a code is $(2, 2, 1/2)$ smooth. The decoding algorithm, when given a codeword $C(x)$ and an index i, picks at random a string $a \in \{0,1\}^n$ and it computes $C(x)[a + e_i] - C(x)[a]$, where all the operations are done modulo 2 and e_i is the vector with a 1 in the ith position and zeroes everywhere else. The algorithm is always correct, because the output of the algorithm is $x \cdot (a + e_i) - x \cdot a$ which, by linearity, is equal to $x \cdot e_i = x_i$. Both queries are uniformly distributed, and so every location is queried with probability exactly $2/2^n$.

The same algorithm also shows that the Hadamard code is $(2, \delta, 1/2 - 2\delta)$ locally decodable for every $\delta < 1/4$.

Since the encoding length of the Hadamard code is 2^n, this implies that for constant ϵ and δ, our lower bound of $2^{\Omega(\epsilon\delta n)}$ on the encoding length of a $(2, \delta, \epsilon)$ locally decodable linear code is tight, and so is, for constant c, our $2^{\Omega(\epsilon n/c)}$ lower bound on the encoding length of a $(2, c, \epsilon)$ smooth linear code.

Finally, we note that the algorithm also implies the existence of a 1-secure 2-server one-round linear PIR scheme with recovery probability 1, query size n and answer size 1. We proved a $\Omega((\epsilon - \delta)n)$ lower bound on the query size of a $(1 - \delta)$-secure 2-server one-round linear PIR scheme with answer size 1, and so our bound is tight for constant ϵ and constant $\delta < \epsilon$.

We are now going to consider some other constructions based on the Hadamard code that show that bound are tight (or almost tight) also in their dependency on the other parameters.

8.1 Smooth Codes

For a given parameter t, that we will set later, we define a code $C : \{0,1\}^n \to \{0,1\}^{t \cdot 2^{n/t}}$ as follows:[8] we divide the input message into t blocks of length n/t each. We encode each block using the Hadamard code, and then we concatenate the encodings together. The length of the encoding is therefore $t \cdot 2^{n/t}$.

The decoding algorithm, given $C(x)$ and i, applies the Hadamard decoding algorithm to the portion of $C(x)$ that contains the encoding of the block of x to which x_i belongs. The decoding procedure makes two queries, and each entry of $C(x)$ has a probability of being queried which is either zero or $2/2^{n/t}$. The decoding procedure is then $(2, 2t, 1/2)$ smooth, and by setting $t = c/2$, we get a $(2, c, 1/2)$-smooth code with encoding length $(c/2) \cdot 2^{2n/c} = 2^{O(n/c)}$.

Consider now the following decoding algorithm: with probability 2ϵ run the above decoding procedure; with probability $1 - 2\epsilon$ make two queries uniformly at random among the entries that would not be queried by the decoding procedure, and then output a random bit. This algorithm has a probability $2\epsilon + (1 - 2\epsilon)/2 = 1/2 + \epsilon$ of outputting the right answer. Regarding smoothness, each

[8] We assume for simplicity that n/t is an integer.

entry has either a probability $2\epsilon/2^{n/t}$ or $2(1-\epsilon)/((t-1)\cdot 2^{n/t})$ of being queried. By setting $t = c/2\epsilon$, we get that the decoding procedure is $(2, c, \epsilon)$-smooth and the encoding length is $(c/2\epsilon) \cdot 2^{2\epsilon n/c} = 2^{O(\epsilon n/c)}$. Our $2^{\Omega(\epsilon n/c)}$ lower bound is thus tight in its dependency on all the parameters.

8.2 Locally Decodable Codes

The $(2, c, 1/2)$-smooth code that we described in the previous section is also a $(2, \delta, 1/2 - c \cdot \delta)$-locally decodable code (whereas its length is $(c/2) \cdot 2^{2n/c}$). (Use the first decoding procedure and note that the decoding error is $c \cdot \delta$.) In particular, setting $c = (1 - 2\epsilon)/2\delta \geq 2$, we can have a $(2, \delta, \epsilon)$ locally decodable code with encoding length $2^{O(n\delta/(1-2\epsilon))}$. Note that this holds only for $\epsilon \leq (1/2) - 2\delta$, and that for very small ϵ this upper-bound does not not quite match our $2^{\Omega(\epsilon\delta n)}$ lower bound. Obata [12] has recently closed this gap by proving an improved, and tight, lower bound of $2^{\Omega(n\delta/(1-2\epsilon))}$.

8.3 Private Information Retrieval

We consider again the encoding $C : \{0,1\}^n \rightarrow \{0,1\}^{t \cdot 2^{n/t}}$ where the input is divided into t blocks, each block is encoded using the Hadamard code, and then the blocks are concatenated together. In the 2-server PIR schemes described in this section, a query is an entry $j \in [t \cdot 2^{n/t}]$ and the reply of a server to query j for database x is the j-th bit of $C(x)$. The query size is thus $n/t + \log t = O(n/t)$. The protocols differ in the choice of t and in the algorithm for the user.

As a first scheme, the user, given index i, runs with probability $1/t$ the Hadamard decoding algorithm to recover the bit i; with probability $1 - 1/t$ it makes two random queries that are uniformly distributed among the queries that are never made by the decoding algorithm for bit i, and then produces a random answer. Each server sees a uniform query regardless of i, so the protocol is 1-secure. The probability of success for the user is $1/t + (1 - 1/t)/2 = 1/2 + 1/2t$. By setting $t = 1/(2\epsilon)$ we get a PIR system with query size $O(\epsilon n)$, perfect security and recovery probability $1/2 + \epsilon$.

Suppose now that we just want to construct a $(1 - \delta)$-secure system. Then the user can run the Hadamard decoding algorithm with probability $\delta + 1/t$ and make random other queries with probability $1 - \delta - 1/t$. This still guarantees $(1 - \delta)$-security. The recovery probability is now $1/2 + \delta/2 + 1/2t$, which is $1/2 + \epsilon$ if we set $t = 1/(2\epsilon - \delta)$. With such a setting we get a $(1 - \delta)$-secure system with recovery probability $1/2 + \epsilon$ and query size $O(n(2\epsilon - \delta))$, which is a close, if imperfect, match for our $\Omega(n(\epsilon - \delta))$ lower bound.

Acknowledgments

We are grateful to Alex Samorodnitsky for suggesting to us the information-theoretic proof of Lemma 3.3 and allowing us to present it in Section 3.4. Thanks also to Noga Alon for helpful discussions, to Yan Zhong Ding for suggesting an improvement in Section 3.3, and to the anonymous referees for their valuable comments.

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] R. Beigel, L. Fortnow and W. Gasarch. A Nearly Tight Lower Bound for Private Information Retrieval Protocols. Technical Note 2002-L001N, NEC Laboratories America, 2002. See also *ECCC*, TR03-087, 2003.

[3] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Rayomnd. Breaking the $O(n^{1/(2k-1)})$ Barrier for Information-Theoretic Private Information Retrieval. In *43rd FOCS*, pages 261–270, 2002.

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

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

[6] 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.

[7] Z. Dvir and A. Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *37th STOC*, pages 592–601, 2005.

[8] W. Gasarch. A Survey on Private Information Retrieval. Bulletin of the EATCS, Vol. 82, 2004.

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

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

[11] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 106–115, 2003.

[12] K. Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *Proc. of RANDOM'02*, pages 39–50. Springer-Verlag, 2002.

[13] L. Trevisan. Some Applications of Coding Theory in Computational Complexity. *ECCC* TR04-043, 2004.

Appendix: Proof of Lemma 3.6

Theorem 2 in [4, Sec. 16] establishes a tighter upper-bound on the number of internal edges; that is, the upper-bound is tight for any possible value of $|S|$. The said upper-bound has a more cumbersome form, which does imply the upper-bound of Lemma 3.6 (i.e., $\frac{1}{2}|S|\log_2|S|$), but Exercise 1 in [4, Sec. 16] only asserts an upper-bound of $\frac{1}{2}|S|\lceil\log_2|S|\rceil$. Here, we present a direct proof of Lemma 3.6, which is simpler than the proof of Theorem 2 in [4, Sec. 16]. This proof seems to be folklore.

The proof is by induction on the size of the set $S \subseteq \{0,1\}^n$. The base case of a singleton set is trivial. In the induction step (i.e., for $|S| \geq 2$), we consider any bit that is not fixed over the strings in S, and denotes by S_0 and S_1 the partition of S according to the value of this bit. Using the induction hypothesis, the number of internal edges in S is at most

$$\frac{1}{2}|S_0|\log_2|S_0| + \frac{1}{2}|S_1|\log_2|S_1| + \min(|S_0|,|S_1|), \tag{22}$$

where the last term upper-bounds the number of edges between S_0 and S_1. Assuming, without loss of generality, that $|S_0| \leq |S_1$ and denoting $m = |S|$ and $x = |S_0|/|S|$, the expression in Eq. (22) is captured by the function $f_m(x) \stackrel{\text{def}}{=} \frac{1}{2}[(xm\log_2 xm) + (1-x)\log_2(1-x)m] + xm$. Clearly, $f_m(x) = (m/2)\cdot(2x - H_2(x)) + \frac{1}{2}m\log_2 m$, where $H_2(x) = x\log_2(1/x) + (1-x)\log_2(1/(1-x))$, and the claim follows by noting that $H_2(x) \geq 2x$ for $x \in (0,1/2]$. The latter fact follows by ($H_2(0) = 0$ and $H_2(1/2) = 1$ and) the convexity of H_2 in the interval $[0,1/2]$, which in turn follows by the fact that $H_2(x) \geq x\log_2(1/x) \geq x$ in this interval.