# Three Query Locally Decodable Codes with Higher Correctness Require Exponential Length*

Anna Gál†
UT Austin
panni@cs.utexas.edu

Andrew Mills
UT Austin
amills@cs.utexas.edu

March 8, 2011

**Abstract**

Locally decodable codes are error correcting codes with the extra property that, in order to retrieve the correct value of just one position of the input with high probability, it is sufficient to read a small number of positions of the corresponding, possibly corrupted codeword. A breakthrough result by Yekhanin showed that 3-query linear locally decodable codes may have subexponential length.

The construction of Yekhanin, and the three query constructions that followed, achieve correctness only up to a certain limit which is $1 - 3\delta$ for nonbinary codes, where an adversary is allowed to corrupt up to $\delta$ fraction of the codeword. The largest correctness for a subexponential length 3-query binary code is achieved in a construction by Woodruff, and it is below $1 - 3\delta$.

We show that achieving slightly larger correctness (as a function of $\delta$) requires exponential codeword length for 3-query codes. Previously, there were no larger than quadratic lower bounds known for locally decodable codes with more than 2 queries, even in the case of 3-query linear codes. Our lower bounds hold for linear codes over arbitrary finite fields and for binary nonlinear codes.

Considering larger number of queries, we obtain lower bounds for q-query codes for $q > 3$, under certain assumptions on the decoding algorithm that have been commonly used in previous constructions. We also prove bounds on the largest correctness achievable by these decoding algorithms, regardless of the length of the code. Our results explain the limitations on correctness in previous constructions using such decoding algorithms. In addition, our results imply tradeoffs on the parameters of error correcting data structures.

*A preliminary version of this paper appeared in the Proceedings of STACS 2011 [10].
†Supported in part by NSF Grants CCF-0830756 and CCF-1018060

# 1 Introduction

Locally decodable codes are error correcting codes with the extra property that, in order to retrieve the correct value of just one position of the input with high probability, it is sufficient to read a sublinear or even just a constant number of positions of the corresponding, possibly corrupted, codeword. While the concept appeared in earlier work (see e.g. [3, 2, 19] the formal definition was given by Katz and Trevisan [12] in 2000.

**Definition 1.** *(Katz and Trevisan [12]) For reals $\delta$ and $\epsilon$, and a natural number $q$, we say that $\mathbf{C}$: $\Sigma^n \to \Gamma^m$ is a $(q, \delta, \epsilon)$-*Locally Decodable Code (LDC) *if there exists a probabilistic algorithm $A$ such that: in every invocation, $A$ reads at most $q$ positions of $y$; and for every $x \in \Sigma^n$ and $y \in \Gamma^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$, and for every $i \in [n]$, we have $\Pr[A^y(i) = x_i] \geq \frac{1}{|\Sigma|} + \epsilon$, where the probability is taken over the internal coin tosses of $A$.*

We will refer to the value $\frac{1}{|\Sigma|}+\epsilon$ in Definition 1 as the *correctness* associated with the given decoding algorithm $A$, while $\epsilon$ can be thought of as the *advantage* over random guessing.

Locally decodable codes have interesting applications, both in complexity theory and in practical areas. Locally decodable codes are especially useful in situations where we want to encode large amounts of data to protect against errors, but need to be able to access individual units; for example, individual patient records of a large hospital. Encoding each unit separately would give less protection against errors, and encoding the whole data set with a traditional error correcting code would require reading the whole encoded database just to access small parts of it. Locally decodable codes are closely related to private information retrieval: constructions of good locally decodable codes yield efficient protocols for private information retrieval. Private information retrieval schemes allow users to retrieve information from databases without revealing information about which data items the user is retrieving. Other applications and related structures include self correcting computations, random self-reducibility, probabilistically checkable proofs. See [20] for a survey. More recently, [8] related LDCs to polynomial identity testing for arithmetic circuits, and [6] to matrix rigidity and circuit lower bounds.

It is quite remarkable that such codes exist at all for constant number of queries. A simple example is the Hadamard code, which has the property that any input bit can be recovered with probability at least $1 - 2\delta$ from codewords possibly corrupted in up to $\delta m$ positions, by a randomized algorithm that in every invocation reads no more than 2 bits of the code. However, the code is very large: the length of the codewords is $2^n$ for encoding $n$ bit inputs.

Of course it would be desirable to have much more efficient, in particular polynomial length codes, but this seems to be currently out of reach for constant number of queries. Efficient constructions are known for large number of queries. See [20] for a survey. A recent paper of Kopparty, Saraf and Yekhanin [14] shows that with $n^\epsilon$ queries, the rate of the code can be close to 1.

It is known that for large enough $n$, 1-query locally decodable codes (that read at most one bit) cannot do better than random guessing [12].

For 2-query linear codes essentially tight bounds are known: Goldreich, Karloff, Schulman and Trevisan [11] proved exponential lower bounds for 2-query linear codes over finite fields up to a certain field size. This was later extended by Dvir and Shpilka [8] to give exponential lower bounds for 2-query linear codes over arbitrary fields. Further improvements for the 2-query linear case were given by [16, 18]. Shiowattana and Lokam [18] prove a lower bound of $\Omega(2^{4\delta n/(1-2\epsilon)})$, which is tight within a constant factor, for 2-query binary linear locally decodable codes.

Kerenidis and de Wolf [13] proved exponential lower bounds for arbitrary binary (not necessarily linear) 2-query locally decodable codes, based on quantum arguments. They also extended their lower bounds to codes over larger alphabets, but the bound decreases with the alphabet size. The strongest lower bounds so far for nonlinear codes from $\{0,1\}^n$ to $\Sigma^m = (\{0,1\}^\ell)^m$ were proved by Wehner and de Wolf [21], and are of the form $2^{\Omega(\delta\epsilon^2 n/(2^{2\ell}))}$. A proof of the 2-query lower bound for binary codes is given in [5] without using quantum arguments. It is still open to obtain nontrivial lower bounds for 2-query nonlinear codes over alphabets of size $\Omega(\sqrt{n})$.

For larger number of queries, there is still a huge gap between the known upper and lower bounds, even for binary linear codes. For codes over small (constant size) alphabets Katz and Trevisan in [12] gave a general lower bound that holds for any $q$ showing that $q$-query locally decodable codes must have length $\Omega(n^{q/(q-1)})$. This bound was slightly improved by Kerenidis and de Wolf [13] to $\Omega((n/\log n)^{\frac{q+1}{q-1}})$, and by Woodruff [23] to $\Omega(n^{\frac{q+1}{q-1}})/\log n$. Woodruff [25] proved $\Omega(n^2)$ lower bounds on the length of 3-query linear codes over any field. Prior to our work, no larger than $n^2$ lower bound was known for locally decodable codes that allow more than 2 queries, even in the case of 3-query linear codes.

A breakthrough result of Yekhanin [26] showed that subexponential length 3-query linear locally decodable codes exist, under assumptions about the existence of infinitely many Mersenne primes. Raghavendra [17] gave some simplifications to Yekhanin's codes. Building on these works, Efremenko [9] gave a construction of subexponential length 3-query linear locally decodable codes without any unproven assumptions. All these constructions have a limit on the correctness achieved by the algorithm as a function of $\delta$ where an adversary can corrupt up to $\delta$ fraction of the codeword positions. Efremenko's construction [9] gives $1-3\delta$ correctness for a 3-query nonbinary code. For 3-query binary codes, the best dependence between the parameters is achieved in a paper by Woodruff [24], which yields 3-query binary linear locally decodable codes with correctness close to, but still below, $1-3\delta$. Note that these results do not provide correctness larger than $1/2$, that is they do not give better correctness than random guessing for binary codes, if the fraction of corrupted positions $\delta$ is larger than $1/6$. The above subexponential size constructions extend to larger number of queries. But while the length of the code gets smaller, the correctness becomes weaker as the number of queries gets larger. Dvir, Gopalan and Yekhanin [7] improve the dependence between correctness and the fraction of corruption for larger number of queries, and achieve subexponential length constructions that can tolerate close to $1/8$ fraction of error for large number of queries. The results of Ben-Aroya, Efremenko, and Ta-Shma [4] give subexponential length locally decodable codes over large enough finite fields that can do better than

random guessing for $\delta$ fraction of corruption up to $\delta = 1/2 - \alpha$ for any $\alpha > 0$, but the number of queries needed and the field size get larger as $\delta$ gets closer to $1/2$.

Next, we describe our results and techniques.

## 1.1 Three query codes

Our main results show that achieving slightly larger than $1 - 3\delta$ correctness for 3-query locally decodable codes requires exponential length. We prove this for arbitrary (possibly nonlinear) binary codes and for linear codes over arbitrary finite fields. Note that larger, e.g. $1 - 2\delta$ correctness can be achieved even by 2-query linear codes: the Hadamard code is an example. With significantly larger number of queries, the correctness can be much higher as a function of $\delta$ (of the form $1 - \delta^{\Omega(q)}$): again the Hadamard code is an example. But this comes at the cost of having large length in the known constructions. Our results show that for 3-query codes, this increase in length cannot be avoided.

Here we give a somewhat simplified statement of the result for binary codes, without specifying the precise constants.

**Theorem 1.1.** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be an arbitrary (possibly nonlinear) binary* $(3, \delta, \epsilon)$*-LDC with a nonadaptive decoder, and* $n$ *large enough. If* $\frac{1}{2} + \epsilon > 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n) + \mu$*, where* $\phi(n) = O(1/n^{1/9})$*, then* $m \geq 2^{\Omega(\mu n^{1/3})}$*.*

We state the precise values hidden in the notation later in Theorem 3.1. We wanted to start with a more compact statement of our bound, showing that as soon as the correctness achieved by the code is above a certain threshold, the length of the codewords must be exponential. For binary codes this threshold is around $1 - 3\delta + 6\delta^2 - 4\delta^3$, which is just slightly larger than $1 - 3\delta$ for small values of $\delta$. The value $1 - 3\delta$ is interesting, since there are subexponential length constructions of 3-query linear LDCs that achieve correctness $1 - 3\delta$ [9] and 3-query binary linear LDCs that achieve correctness slightly below $1 - 3\delta$ [24]. The value $1 - 3\delta + 6\delta^2 - 4\delta^3$ corresponds to the probability that the number of corrupted positions in a given triple is even, where the probability is over the distribution that corrupts each bit independently with probability $\delta$.

For linear codes over arbitrary finite fields, we obtain stronger lower bounds. In our results for nonbinary codes, the value of the threshold is close to the threshold for the binary case, but slightly depends on the field size. We obtain exponential lower bounds for arbitrary finite fields, even if the field size depends on $n$.

We note that our bound holds for any $\delta \geq 0$ and any $0 < \epsilon \leq 1 - 1/|F|$, where $\delta$ and $\epsilon$ may be $o(1)$. For the bound to be nontrivial, we need $\delta \leq 1 - 1/|F|$, because of Observation 2.1. For $n$ to be large enough for our purposes, it is sufficient if $\delta > \Omega(1/n^{1/9})$ and $\epsilon > \Omega(1/n^{1/3})$ for binary nonlinear codes, and if $\delta > \Omega((|F|/n)^{1/3})$ and $\epsilon > \Omega(|F|/n)$ for linear codes over a field $F$.

## 1.2 Arbitrary number of queries

We obtain similar results for arbitrary number of queries, under some assumptions on the decoding algorithm. We note that the types of decoding algorithms we consider have been commonly used in recent constructions. Our results explain the limitations on correctness of these constructions.

Unless otherwise noted, a $q$-query decoder is allowed to use less than $q$ queries. So the correctness thresholds for requiring exponential length for $q$-query codes are never going to be smaller than the correctness thresholds for the same class of 3-query codes. In the special cases below, we show that the same thresholds to require exponential length as for 3-query codes also apply for arbitrary number of queries.

It remains open what is the correctness threshold (as a function of $\delta$) to require exponential length for general $q$-query codes. Note that it will have to be a value larger than the threshold in our 3-query results: a $q$-query code for $q > 3$ can always do at least as well, as a 3-query code. We will see below, that if we require the query sets to be exactly of size $q$, then this is not necessarily the case.

### 1.2.1 Linear Decoders

One of the starting points of our approach was the observation that using larger number of queries does not help to tolerate errors if the decoder returns a fixed linear combination of the positions read. Moreover, the probability of error increases with the number of positions used with nonzero coefficients by a linear decoder. We formalize these ideas in our results about linear decoders.

We show that linear decoders that use exactly $q$ positions cannot achieve larger correctness than $1 - q\delta + o(\delta) + O(1/n)$, *regardless of the length of the code*. Moreover, we show that the correctness of any linear decoder, for any number of positions used, is at most $1-2\delta+o(\delta)+O(1/n)$. This holds for arbitrary (possibly nonlinear) codes and over any finite field $F$. This implies that our exponential length lower bounds extend to linear decoders with arbitrary number of queries, with the same correctness threshold as for 3-query codes.

Linear decoders are commonly used in the known constructions of locally decodable codes. In fact it is noted for example in [13, 23] that any (possibly nonlinear) binary $(q, \delta, \epsilon)$-LDC has a linear decoder that achieves correctness $1/2 + \epsilon/2^q$.

In the case of linear *smooth codes* (see [12]), requiring the decoders to be linear is inconsequential: for linear codes, if any algorithm gives nontrivial advantage over random guessing when querying a given set of codeword positions $Q$, then by Lemma 2.2, $e_i \in \text{span}(Q)$ must hold. Thus, there is a fixed linear combination of the positions in $Q$ that gives the correct value of $x_i$ for any input $x$. Using the same procedure as the original decoder to choose which positions to query and then returning this fixed linear combination (if it exists) cannot violate the smoothness of the code.

However, for locally decodable codes (both linear and nonlinear), requiring to use only linear decoders may significantly reduce the correctness associated with the code. For example, taking majorities, one can obtain correctness of the from $1 - \delta^{\Omega(q)}$. Our results show

that in the recent results of [7, 4] obtaining subexponential length constructions with larger than $1 - q\delta$ correctness for larger values of $q$, the use of nonlinear operations in the decoding algorithm is important.

Our results on linear decoders imply that there is no significantly better general reduction from smooth codes to locally decodable codes than the current bounds giving at most $1 - q\delta$ correctness for $q$ query locally decodable codes. The possibility of better reductions was raised in [12].

We formally define linear decoders and prove our results about them in Section 4.

### 1.2.2 Matching sum decoders

Matching sum decoders were formally defined by Woodruff [24]. A $q$-query matching sum decoder picks a set of size $q$ uniformly at random from a collection of sets that form a matching in the complete $q$-uniform hypergraph, whose vertices correspond to the positions of the codeword. Then, the decoder reads the positions corresponding to the chosen set, and returns the sum of the positions read. Most known constructions of locally decodable codes have such decoders.

Matching sum decoders are a subclass of linear decoders, thus our results on linear decoders immediately apply. However, for matching sum decoders we can prove stronger results. We show that $q$-query matching sum decoders cannot achieve larger correctness than $1 - q\delta$, *regardless of the length of the code*. This holds for arbitrary codes and over any field $F$.

Woodruff [24] proved that LDCs with 2-query matching sum decoders must have exponential length. Considering matching sum decoders where the query size is not fixed, we show that for any binary code (possibly nonlinear), and for linear codes over arbitrary finite fields, if a matching sum decoder with query sets of size at most $q$ achieves correctness more than $1 - 3\delta + O(1/n)$, then the length of the code must be exponential.

We prove our results for matching sum decoders in Section 4.1.

### 1.2.3 Query sets with large rank

For linear codes our proofs also apply to arbitrary number of queries, and possibly nonlinear decoders as long as the vectors corresponding to the positions queried are linearly independent. This is a property that holds in some of the known constructions of linear locally decodable codes. For such query sets, we show that if the correct value of $x_i$ is spanned by $q$ of the linearly independent vectors with nonzero coefficients, then the correctness of the decoder cannot be larger than $1 - q\delta + o(\delta) + O(1/n)$, *regardless of the length of the code*.

This implies, that for linear codes over arbitrary finite fields, if a $q$-query decoder (with query sets of size at most $q$) queries only linearly independent positions of the code and achieves correctness more than $1 - 3\delta + o(\delta) + O(1/n^{1/3})$, then the length of the code must be exponential. The exponential length lower bound extends to query sets that are not fully independent, but have large rank, with a correctness threshold that depends on the rank of

the query sets. The results described for query sets with large rank are direct consequences of our proofs for linear codes, specifically Lemmas 3.10 and 3.11. See also Corollary 3.12.

## 1.3 Error Correcting Data Structures

Error correcting data structures were defined by de Wolf [22]. Such data structures are a variation of the traditional bit-probe model (see e.g. [15]), where the algorithms answering questions about the data are correct with probability at least $1/2 + \epsilon$, as long as at most $\delta$ fraction of the database representing the data is corrupted, possibly by adversarial error. It is noted in [22] that error correcting data structures for the membership problem yield locally decodable codes, with the same parameters. [22] showed the existence of error correcting data structures for the membership problem and some of its variants, assuming the existence of locally decodable codes with given parameters. Because of the direct correspondence between the two models, our results rule out the existence of error correcting data structures for membership of subexponential size with larger correctness than our thresholds above, for 3-probe algorithms, as well as for algorithms with arbitrary number of probes, assuming the algorithm only uses linear operations.

## 1.4 Techniques

We start by noting why some intuition based on smooth codes would fail to explain our most general results. *Smooth codes* were defined by Katz and Trevisan [12], who also gave reductions between smooth codes and locally decodable codes. So up to changes in parameters, smooth codes and locally decodable codes are equivalent. Most of the current lower bounds for locally decodable codes have been proved via proving lower bounds for smooth codes, and the correctness of the known subexponential length constructions of 3-query linear LDCs is analyzed based on their property of having *smooth decoders*, that are correct with large probability if there is no error, and query each position of the code with not too large probability. However, the current techniques to analyze smooth decoders cannot imply larger than $1 - q\delta$ correctness for $q$-query locally decodable codes. In fact we show that no significantly better general reduction is possible.

We elaborate on a few specific points below. One could try to argue that the probability that the decoder does not query any corrupted positions is upper bounded by a function not much larger than $1 - q\delta$, thus the decoder will have to read corrupted positions. However, errors may cancel out, so the fact that some of the positions read by the decoder may contain an error, in itself does not explain our lower bounds.

If the decoder was only working with query sets that form a matching, and the decoder was linear (which is the case in several of the known constructions), then, as we show in Section 4.1, $1 - q\delta$ would in fact be a limit on correctness for decoders that query exactly $q$ positions. But these assumptions do not have to hold for every decoding algorithm, and our results cannot be explained by this simplified view.

Our proofs of the 3-query lower bounds are based on a lemma that was central in obtaining the exponential lower bounds for 2-query codes. However, we would like to emphasize that we

do not use 2-query lower bounds as a black box. We show that query sets that provide large correctness must contain subsets of size at most 2 that give nontrivial correlation with the input position we try to recover. But this does not imply that the code somehow "reduces" to a 2-query code. Consider the following simple example (many other examples are possible): query 3 positions such that each in itself has large correlation with the position $x_i$, and take the majority of the answers. Replacing this with reading only a subset of the bits, would preserve the properties of a smooth decoder, but it would reduce the correctness of the decoding algorithm. Thus, the decoder cannot be simply replaced by a 2-query decoder, if we want to preserve the correctness probabilities of the decoding algorithm.

Our approach can be summarized as follows: we show that in the case of 3-query codes, if the code is small, we can "force" the decoder to only examine query sets that are vulnerable to error. We achieve this by considering the algorithm's performance over random input $x$ and a specially constructed distribution for the corruption caused by the adversary. We show that over our distribution, the decoder cannot perform much better than a linear decoder.

In all of our results, the probability of error is estimated in terms of the probability - over appropriate random corruption - of the event that the sum of the corruption in the positions of a given query set is nonzero. Intuitively, this probability would indeed give a lower bound on the error if the decoder always returned the sum (or a fixed linear combination) of the positions read, and if this was equal to the correct answer for uncorrupted codewords. For example, this would be the case for linear decoders of a linear code. However, we also consider nonlinear codes, and arbitrary decoders that may involve nonlinear operations. In fact, we do not claim that the probability of having nonzero sum of corruption in the query set is a lower bound on the error in general. Instead, we lower bound the probability of error by a different expression, and show that this expression is lower bounded by the probability mentioned above in the case of random corruption according to our distribution.

A crucial point in our proofs for nonlinear decoders (for both linear and nonlinear codes) is comparing the conditional probabilities of error of the decoder, conditioned on the sum of the values in the corrupted positions. We show that - under appropriate assumptions on the query sets for linear codes - the sum of these conditional probabilities of the decoder being incorrect, is always $|F| - 1$. A subtle point of this argument is that the various events we work with are not always independent. Our proof for nonlinear codes is based on a similar property of conditioning on the number of corrupted positions being odd vs. even. However, for nonlinear codes instead of directly considering the conditional probabilities of incorrect decoding, we reduce estimating the probability of error to estimating the probability that the sum of the positions read gives an incorrect answer. This analysis lets us estimate the probability of incorrect decoding even if the decoding algorithm uses nonlinear operations.

# 2 Preliminaries

The definition of locally decodable codes allows the decoding algorithm to be adaptive. Lower bounds for nonadaptive decoders can be translated to lower bounds for arbitrary decoders with the same number of queries but larger correctness: it is noted in the paper by Katz and

Trevisan [12] that any adaptive $(q, \delta, \epsilon)$ decoding algorithm for a code $\mathbf{C}$: $\Sigma^n \to \Gamma^m$, can be transformed to a nonadaptive $(q, \delta, \epsilon/|\Gamma|^{q-1})$ decoding algorithm for the same code $\mathbf{C}$.

We only consider nonadaptive decoding algorithms in the rest of the paper. We will refer to the (at most $q$) positions the algorithm chooses to read in a given invocation as a *query set*. In a nonadaptive algorithm, the choice of the query set only depends on the coin flips of the algorithm.

The following simple observation means that for proving lower bounds we may assume that $\delta < 1 - \frac{1}{|\Sigma|}$, since otherwise, no algorithm can do better than random guessing for *any* of the input positions.

**Observation 2.1.** *Let $A$ be a decoding algorithm for any nontrivial code $\mathbf{C}$: $\Sigma^n \to \Sigma^m$. If $\delta \geq 1 - \frac{1}{|\Sigma|}$, then for any $i \in [n]$, $\min_{x \in \Sigma^n} \left( \min_{y \in \Gamma^m \,:\, d(y, \mathbf{C}(x)) \leq \delta m} \Pr\left[A^y(i) = x_i\right] \right) \leq \frac{1}{|\Sigma|}$.*

*Proof.* Arbitrarily choose $i \in [n]$. For each $s \in \Sigma$, let $v_s \in \Sigma^n$ be a vector such that $(v_s)_i = s$. Split $[m]$ into a partition of $|\Sigma|$ equal size subsets named $U_s$ for $s \in \Sigma$. Construct $Y \in \Sigma^m$ such that for each $s \in \Sigma$, $Y$ agrees with $\mathbf{C}(v_s)$ on the positions in $U_s$. Whenever the input to $\mathbf{C}$ is one of the vectors $v_s$, the adversary will corrupt the codeword to become equal to $Y$ by modifying at most $m(1 - \frac{1}{|\Sigma|})$ positions. We have $\sum_{s \in \Sigma} \Pr\left[A^Y(i) \text{ outputs } s\right] = 1$, where the probability is over the internal coin tosses of $A$. Thus there exists at least one $s \in \Sigma$ such that, if the adversary corrupts $\mathbf{C}(v_s)$ into $Y$, the probability of the algorithm correctly answering $s$ is at most $\frac{1}{|\Sigma|}$. Therefore, we have shown there exists an input $x$ and an adversary error pattern of size at most $\delta m$ such that the probability of error is at least $1 - \frac{1}{|\Sigma|}$. So $\min_{x \in \Sigma^n} \left( \min_{y \in \Gamma^m \,:\, d(y, \mathbf{C}(x)) \leq \delta m} \Pr\left[A^y(i) = x_i\right] \right) \leq \frac{1}{|\Sigma|}$. □

For a linear code $\mathbf{C}$: $F^n \to F^m$, it is convenient to represent the function that determines a given codeword position by a vector: for $j \in [m]$, define $a_j \in F^n$ as the vector satisfying $\forall x \in F^n$, $\mathbf{C}_j(x) = a_j \cdot x$. For vectors $a, x \in F^n$, we use $a \cdot x$ to denote their inner product over $F$. (We omit $F$ from the notation.)

For a query set $Q = \{j_1, \ldots, j_q\} \subset [m]$, we use the notation $\mathrm{span}(Q)$ to represent the linear span of the vectors $a_{j_1}, \ldots, a_{j_q}$ corresponding to the positions in $Q$. We denote the $i$'th unit vector with length $n$ by $e_i$. That is, $e_i$ has 1 in its $i$-th coordinate and 0 everywhere else.

The following lemma was stated in [11] for two query binary linear codes. Its extension to arbitrary fields and any number of queries is straightforward, but important for our arguments. For completeness, we include a proof.

**Lemma 2.2.** *(implicit in [11]) Let $\mathbf{C}$: $F^n \to F^m$ be a linear code. Let $i \in [n]$ and let $Q = \{j_1, j_2, \ldots j_q\} \subset [m]$ be a query set that the algorithm $A$ queries with nonzero probability when trying to recover the value of input position $i$. Suppose $\Pr_{x \in_U F^n}\left[A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } Q\right] > \frac{1}{|F|}$ where the probability is taken over letting $x$ be uniformly random from $F^n$ and over the internal coin tosses of $A$. Then $e_i \in \mathrm{span}(Q)$ must hold.*

*Proof.* We prove the contrapositive. Take any $i$ and $Q = \{j_1, j_2, \ldots j_q\} \subset [m]$ such that there do not exist $c_{j_1}, c_{j_2}, \ldots c_{j_q} \in F$ for which $\sum_{k=1}^{q} c_{j_k} a_{j_k} = e_i$. Say $y_{j_1}, y_{j_2}, \ldots y_{j_q}$ are the respective

values $A$ receives from querying $Q$. The algorithm's job when it queries $Q$ is to solve the following system of $q$ linear equations for $x_i$:

$$\begin{aligned} y_{j_1} &= a_{j_1} \cdot x \\ y_{j_2} &= a_{j_2} \cdot x \\ &\vdots \\ y_{j_q} &= a_{j_q} \cdot x \end{aligned}$$

Assume without loss of generality that $a_{j_1}$, $a_{j_2}$, ... $a_{j_{q'}}$ is a maximal collection of linearly independent vectors from $a_{j_1}$, $a_{j_2}$, ... $a_{j_q}$, for some $q' \leq q$. (Simply renumber the $a$'s so this is true.) Therefore, the system of $q$ linear equations above turns into a system of $q'$ independent linear equations.
Because the vector $e_i$ is not in the span of $\{a_{j_1}, a_{j_2}, ...a_{j_{q'}}\}$, there exists an $\hat{x}$ satisfying:

$$\begin{aligned} e_i \cdot \hat{x} &= 1 \\ a_{j_1} \cdot \hat{x} &= 0 \\ a_{j_2} \cdot \hat{x} &= 0 \\ &\vdots \\ a_{j_{q'}} \cdot \hat{x} &= 0 \end{aligned}$$

Because this is a system of $q' + 1$ independent linear equations, a solution $\hat{x}$ must exist. Note that $\hat{x} \neq 0$ because $\hat{x}_i \neq 0$. Using $\hat{x}$, we define the following set:

$$V_x \triangleq \{x, x + \hat{x}, x + 2\hat{x}, ..., x + (|F| - 1)\hat{x}\}$$

For any $x$ that is a solution of the original $q$ equations, every member of $V_x$ is a solution as well, but each has a different $i$'th coordinate. Notice for any $x' \notin V_x$, $V_x \cap V_{x'} = \varnothing$. This implies that the number of solutions to the original $q$ equations having $i$'th coordinate equal $d$, for any $d \in F$, is the same. Recall that we are considering uniform $x \in F^n$. So,

$$\forall d \in F : \Pr_{x \epsilon_U F^n} \left[ x_i = d \mid y_{j_1} = a_{j_1} \cdot x, y_{j_2} = a_{j_2} \cdot x, \text{ ..., and } y_{j_q} = a_{j_q} \cdot x \right] = \frac{1}{|F|}$$

This implies

$$\Pr_{x \epsilon_U F^n} \left[ A^{\mathbf{C}(x)}(i) = x_i \mid A \text{ queries } Q \right] = \frac{1}{|F|}$$

This contradicts the assumption from the theorem's statement that the probability of correctness is strictly greater than $\frac{1}{|F|}$.

□

We will use the following simple fact throughout our proof.

**Fact 2.3.** *(implicit in [1]) Let $a_1, \ldots, a_t$ be vectors from $F^n$. For $x$ uniformly random from $F^n$, the corresponding random values $a_1 \cdot x, \ldots, a_t \cdot x$ are $t$ independent uniformly distributed values from $F$, if and only if the vectors $a_1, \ldots, a_t$ are linearly independent over $F$.*

*Proof.* Let us consider what happens when $a_1, \ldots, a_t$ are linearly independent over $F$. Then, for any set of values $d_1, \ldots, d_t \in F$, the number of $x$ such that $\forall 1 \leq i \leq t$, $a_i \cdot d_i$ is the same: $|F|^{n-t}$. Since the distribution of $x$ is uniformly random, $a_1 \cdot x, \ldots a_t \cdot x$ are $t$ independent, uniformly random values from $F$.

If $a_1, \ldots, a_t$ are not linearly independent, then there exist $c_1, \ldots, c_t \in F$ such that at least one of them is nonzero, and $\sum_{k=1}^{t} c_k a_k = 0$ (where the sum is over $F$). This means that certain sets $d_1, \ldots, d_t \in F$, will never appear as values for $a_1 \cdot x, \ldots a_t \cdot x$, since values such that $\sum_{k=1}^{t} c_k d_k = 1$ will never appear. In particular, if $t'$ is such that $c_{t'} \neq 0$, then the set of values $\forall i \neq t'$, $d_i = 0$ and $d_{t'} = 1$ will never appear. Thus $a_1 \cdot x, \ldots a_t \cdot x$ cannot be uniformly distributed. □

The following theorem of Goldreich, Karloff, Schulman and Trevisan [11] is a crucial ingredient of our proofs.

**Theorem 2.4.** *[11] Let $a_1$, ... $a_m$ 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 $m \geq 2^{2\alpha n}$, where $\alpha \triangleq \frac{\sum_{i=1}^{n} |M_i|}{nm}$.*

This theorem was extended to arbitrary finite fields in [11]. The dependence on the field size in the bound was removed by Dvir and Shpilka in [8]. We will use the following version (see Corollary 2.9 in [8]).

**Theorem 2.5.** *[8] Let $F$ be a field. Let $a_1$, ... $a_m$ be a sequence of (not necessarily distinct) elements of $F^n$ such that for every $i \in [n]$ there is a set $M_i$ of disjoint pairs of indices $\{j_1, j_2\}$ such that $e_i \in \mathrm{span}(a_{j_1}, a_{j_2})$. Then $m \geq 2^{\alpha n - 1}$, where $\alpha \triangleq \frac{\sum_{i=1}^{n} |M_i|}{nm}$.*

A version of the theorem applicable to binary nonlinear codes is given in the "non-quantum" proof of the exponential lower bounds for 2-query binary nonlinear codes by Ben-Aroya, Regev and de Wolf [5].

**Theorem 2.6.** *(implicit in Theorem 11 of [5]) Let $0 < \epsilon, \alpha < 1/2$. Let $a_1$, ... $a_m$ be a sequence of (not necessarily distinct) functions from $\{0,1\}^n$ to $\{0,1\}$ such that for at least $\tau n$ indices $i \in [n]$ there is a set $M_i$ of disjoint pairs of indices $\{j_1, j_2\}$ such that $|M_i| \geq \alpha m$ and*

$$|\Pr_x [x_i = a_{j_1}(x) \oplus a_{j_2}(x)] - \Pr_x [x_i \neq a_{j_1}(x) \oplus a_{j_2}(x)]| \geq \epsilon$$

*where the probability is over uniform $x \in \{0,1\}^n$. Then $m \geq 2^{\tau \alpha^2 \epsilon^2 n}$.*

We also use the following theorem of Katz and Trevisan [12].

**Theorem 2.7.** *(Theorem 2 in [12]) Let $\mathbf{C} : \{0,1\}^n \to R$ be a function. Assume there is an algorithm $A$ such that for every $i \in [n]$, we have $\Pr_x [A(\mathbf{C}(x), i) = x_i] \geq \frac{1}{2} + \epsilon$, where the probability is taken over the internal coin tosses of $A$ and uniform $x \in \{0,1\}^n$. Then $\log |R| \geq (1 - H(1/2 + \epsilon))n$.*

## 2.1 Notation

Let $F$ be an arbitrary finite field. We denote by $F^*$ the set of nonzero elements of $F$. Arithmetic operations involving field elements are over $F$. This should be clear from the context, and will be omitted from the notation.

For a code $\mathbf{C}$: $F^n \to F^m$, we can represent any vector $y \in F^m$ with $d(y, \mathbf{C}(x)) \leq \delta m$ as a sum of the form $y = \mathbf{C}(x) + B$, where $B \in F^m$, such that the number of nonzero entries in $B$ is at most $\delta m$.

We will use the notation $\Pr_{x,B,A}$ to indicate probabilities over uniformly random input $x$ from $F^n$, $B$ chosen at random from a given distribution for corruption, and the random coin tosses of the given algorithm $A$.

Note that while in general the corruption may be produced by an arbitrary adversary, we will only consider distributions for $B$ that do not depend on the input $x$ or on the distribution for the coin tosses of the algorithm. This is sufficient for our purposes, since we are proving lower bounds on the length of the code.

# 3 Lower Bounds for Three Query Codes

## 3.1 Lower Bounds for Three Query Binary Codes

We state the precise version of our lower bound for arbitrary (possibly nonlinear) binary codes.

**Theorem 3.1.** *Let $\mathbf{C}$: $\{0,1\}^n \to \{0,1\}^m$ be a $(3, \delta, \epsilon)$-LDC with a nonadaptive decoder, and $n$ large enough. Let $\alpha \triangleq \delta - (\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}) - (3/n^{1/3} + \frac{36}{n})^{1/3} - \nu$, and $\nu \triangleq \frac{10}{n(1-H(1/2+1/n^{1/3}))} = O(1/n^{1/3})$. If $\alpha > 0$, then $m \geq 2^{0.225\alpha^2 n^{1/3}}$.*

**Remark 1.** *We will show in Claim 5.1 that $\alpha > 0$ when $\frac{1}{2}+\epsilon > 1-3\delta+6\delta^2-4\delta^3+\phi(n)$, where $\phi(n) = 4((3/n^{1/3}+\frac{36}{n})^{1/3}+\nu)$. Moreover, $\alpha > \frac{\mu}{4}$ when $\frac{1}{2}+\epsilon > \mu+1-3\delta+6\delta^2-4\delta^3+\phi(n)$ for some $\mu \geq 0$. This implies the version of the bound stated in Theorem 1.1 for binary codes. Note that we could also obtain a lower bound of the form $2^{\Omega(n)}$ by setting $\epsilon_2$ in the proof to a constant, but then the correctness required for the bound would be larger, roughly by $4(\epsilon_2)^{1/3}$.*

We start with an overview of the proof. For the case of binary (possibly nonlinear) codes, using the Fourier representation of Boolean functions, and properties of correlation, we show that for any decoding algorithm, and for any query set $Q$, the advantage of the algorithm over random guessing when reading the values of the query set $Q$ is at most the sum of the advantages obtained by all possible fixed linear functions over the given query set. See Claim 3.7 for a precise statement. This observation has been implicitly used also in the arguments of [13] and [25] showing the existence of linear decoders with correctness $1/2 + \epsilon/2^q$ for any binary $(q, \delta, \epsilon)$-LDC.

In all our proofs, we use a distribution for the adversary that corrupts each codeword position in a particular set $S$ independently, and chooses the corruption over the remaining set of positions so that the total fraction of corrupted positions is still below $\delta$. We construct

the set $S$ so that for query sets that do not intersect the set $S$, the contribution of sums over subsets of size at most 2 towards the advantage over random guessing is small. However, the size of $S$ has to be small to keep the total fraction of corrupted positions below $\delta$.

To achieve this, we first argue that in any LDC, the number of codeword positions that have large correlation with a given input bit $x_i$ over random input $x$ must be small for most input positions $i \in [n]$. This is straightforward for linear codes. Note however that for nonlinear codes, it is possible that a given codeword position has significant correlation with more than one input bit. We show the desired statement using Theorem 2.7.

Next we consider pairs of codeword positions, such that the sum of their values gives large correlation with a given input bit $x_i$ over random input $x$. Using Theorem 2.6, we show that if the length of the code is small, then for most $i \in [n]$, all such pairs of positions can be covered by a small number of codeword positions.

This allows us to conclude that *if the length of the code is small*, then for at least one index $i \in [n]$, there exist a set $S$ of small size, such that for query sets that do not intersect the set $S$, the contribution of sums over subsets of size at most 2 towards the advantage over random guessing the bit $x_i$ is small.

In Lemma 3.5 we show that for any LDC with correctness $1/2 + \epsilon$, there is a decoding algorithm that never reads any of the positions in $S$, but is correct with probability at least $1/2+\epsilon$ on average over random input $x$, and the random corruption of the above distribution. Note that the algorithm may not achieve the required correctness on every input and for every string within distance $\delta m$ of $\mathbf{C}(x)$. We only claim a bound on its probability of being correct over uniformly random $x$ and over random corruption according to our distribution.

This way we can argue that *if the length of the code is small* then there is a decoding algorithm that only uses query sets that either provide only small advantage over random guessing, or they involve 3 codeword positions, such that the sum of the 3 positions gives the correct value of the input bit $x_i$ with large probability over random input and the random corruption according to our distribution.

On the other hand, for this decoding algorithm we can lower bound the probability of error by the probability that the sum of a given triple of codeword positions gives an incorrect value over random input and the random corruption according to our distribution.

See Section 3.5 for a detailed proof of the theorem and Section 3.4 for the formal description of the distribution for the random corruption.

## 3.2 Lower Bounds for Three Query Linear Codes over Arbitrary Finite Fields

For linear codes over arbitrary finite fields, we obtain stronger lower bounds than our bounds for nonlinear codes.

Let $F$ be an arbitrary finite field. We denote by $F^*$ the set of nonzero elements of $F$. It is convenient to state the threshold on correctness in our bounds in terms of the probability of the event that a fixed linear combination of a given triple of coordinates of an appropriate random corruption equals to 0. More precisely, let $Q \subseteq [m]$ with $|Q| = q$ be an arbitrary

fixed subset of the coordinates. Let $c_j \in F^*$, for $j \in Q$ and let $\delta \leq 1 - 1/|F|$. For the distributions we work with, the values of $c_j$ will not make a difference, as long as they are all nonzero. Let $P(\delta, q, F) \triangleq \Pr_B \left[ (\sum_{j \in Q} c_j B_j = 0) \right]$, where the probability is over $B \in F^m$ randomly chosen according to a distribution that first chooses to corrupt each coordinate in $[m]$ independently with probability $\delta$, and then uniformly and independently assigns a value from $F^*$ to each chosen coordinate of $B$. The remaining coordinates of $B$ are set to 0. Note that an adversary using this distribution would possibly corrupt more than $\delta m$ positions with nonzero probability, so this is not the distribution we use in our proofs. But it is convenient to use the probability $P(\delta, q, F)$ in the statement of our bounds, since $P(\delta, q, F)$ only depends on $\delta$, $q$ and $|F|$, it does not depend on $m$. The theorem also holds using the distribution that chooses $\delta m$ positions uniformly (instead of independently corrupting the positions). While it is well known that these probabilities are not too far from each other in the two distributions, for our purposes we need more precise estimates than the standard ones. We carefully estimate the difference between the two probabilities using Claim 5.7 in the Appendix.

We start with a simplified statement of the result without specifying the precise constants.

**Theorem 3.2.** *Let $\mathbf{C}$: $F^n \to F^m$ be a linear $(3, \delta, \epsilon)$-LDC with a nonadaptive decoder, and $n$ large enough. If $\frac{1}{|F|} + \epsilon > P(\delta, 3, F) + \phi(n) + \mu$, where $\phi(n) = O(|F|/n^{1/3})$, then $m \geq 2^{\Omega(\mu n)}$.*

We state the precise values hidden in the notation in Theorem 3.3.

For binary linear codes we present a slightly stronger bound in the next section.

**Theorem 3.3.** *Let $\mathbf{C}$: $F^n \to F^m$ be a linear $(q = 3, \delta, \epsilon)$-LDC with a nonadaptive decoder, $\delta \leq 1 - \frac{1}{|F|}$, and $n$ large enough. Then, $m \geq 2^{.45\alpha n - 1}$ where $\alpha \triangleq \delta - (1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}) - (\frac{108|F|}{n})^{1/3} - \frac{10}{n}$.*

**Remark 2.** *We will show in Claim 5.13 that $\alpha > 0$ when $\frac{1}{|F|} + \epsilon > 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$, where $\phi(n) = 4((108|F|/n)^{1/3} + 10/n)$. Moreover, $\alpha > \frac{\mu}{4}$ when $\frac{1}{|F|} + \epsilon > \mu + 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$ for some $\mu \geq 0$. This implies the version of the bound stated in Theorem 3.2.*

Similarly to the proof for binary codes, we construct the set $S$ of positions that we corrupt independently, so that for query sets that do not intersect the set $S$, the contribution of sums (or linear combinations) over subsets of size at most 2 towards the advantage over random guessing is small. Linear codes have the very strong property that linear combinations of codeword positions are either exactly equal to a given input bit, or give no advantage over random guessing towards recovering the given input bit (see Lemma 2.2). Thus, in the case of linear codes, we can construct the distribution of the adversary so that the decoding algorithm is left with query sets of size 3, such that no subsets of size at most 2 can give any advantage over random guessing. All other query sets that the algorithm can read will give no advantage over random guessing for a given input bit. Thus we don't need the

part of the argument using Fourier representation of Boolean functions used in the binary proof to reduce estimating the probability of error to estimating the error over query sets of a special form. However, we still need to deal with the fact that the decoders can use nonlinear operations. We achieve this by considering the conditional probabilities of error of the decoder, conditioned on the sum (more precisely a fixed linear combination) of the values in the corrupted positions. In addition, we show that in the query sets we are left with the positions must correspond to linearly independent vectors in the generator matrix of the code. Based on this, we show that the probability of error (on average using our distribution) is lower bounded by $|F| - 1$ times the minimum over $k \in F$ of the probability that a fixed linear combination (with nonzero coefficients) of the corruption in the positions of the query set equals $k$.

See Section 3.6 for a detailed proof of the Theorem.

## 3.3 Lower Bounds for Three Query Binary Linear Codes

For binary linear codes, we obtain a slightly stronger bound than what follows from the lower bound for linear codes over arbitrary finite fields.

**Theorem 3.4.** *Let* $\mathbf{C}$*:* $\{0,1\}^n \to \{0,1\}^m$ *be a linear* $(3, \delta, \epsilon)$*-LDC with a nonadaptive decoder, and* $n$ *large enough. Then,* $m \geq 2^{1.8\alpha n}$ *where* $\alpha \triangleq \delta - (\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}) - (\frac{36}{n})^{1/3} - \frac{10}{n}$.

**Remark 3.** *We will show in Claim 5.1 that* $\alpha > 0$ *when* $\frac{1}{2} + \epsilon > 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n)$*, where* $\phi(n) = 4((36/n)^{1/3} + 10/n)$*. Moreover,* $\alpha > \frac{\mu}{4}$ *when* $\frac{1}{2} + \epsilon > \mu + 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n)$ *for some* $\mu \geq 0$*. This implies the version of the bound stated in Theorem 3.2 for binary codes.*

The proof is almost identical to the proof in the previous section for arbitrary finite fields. The improvement comes from using Theorem 2.4, and because in the case of binary linear codes we can use a node cover of size $|M_1|$ instead of $2|M_1|$ when defining the distribution.

## 3.4 The Probability Distribution of the Adversary

We will work with probability distributions of the following general structure for the corruption. We will specify the sets $R$, $S$ and the distribution $D_R$ later in this section.

Let $\mathbf{C}$: $F^n \to F^m$ be a code. Consider two disjoint subsets $R, S \subseteq [m]$. Let $B_1 \in F^m$ be chosen according to some probability distribution $D_R$ over vectors in $F^m$ that are identically zero in coordinates outside of $R$. Let $B_2 \in F^m$ be chosen according to the following probability distribution $D_S$ over vectors in $F^m$ that are identically zero in coordinates outside of $S$: independently for each coordinate $j \in S$, with probability $\frac{1}{|F|}$ let $(B_2)_j = c$ for each $c \in F$. Note that when $|F| = 2$, $D_S$ is simply the binomial distribution with probability $1/2$ over the coordinates in $S$.

Let $B = B_1 + B_2$, generated by the product distribution of the distributions $D_R$ and $D_S$.

We will use the following lemma about probability distributions of the above structure for the corruption.

**Lemma 3.5.** *Assume there exists a $q$ query algorithm $A$ such that*

$$\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(i) = x_i\right] \geq \frac{1}{|F|} + \epsilon$$

*where the probability is over the internal coin tosses of $A$, uniform $x \in F^n$, and $B = B_1 + B_2$ chosen by the product distribution of the distributions $D_R$ and $D_S$, where $R$ and $S$ are disjoint subsets of $[m]$, $D_R$ is arbitrary over vectors in $F^m$ that are identically zero in coordinates outside of $R$, and $D_S$ is defined as above. Then there exists a $q$ query algorithm $\tilde{A}$ such that*

$$\Pr_{x,B,\tilde{A}}\left[\tilde{A}^{\mathbf{C}(x)+B}(i) = x_i\right] \geq \frac{1}{|F|} + \epsilon$$

*as well, and $\tilde{A}$ never queries any positions from $S$.*

*Proof.* Without loss of generality, let $S = [t]$ for some $1 \leq t \leq m$. For any $s'_1$, $s'_2$, ... $s'_t \in F$, there exists exactly one sequence of $s_1$, $s_2$, ... $s_t \in F$ such that $\mathbf{C}(x)_j + s_j = s'_j$ for $j \in [t]$.

Each sequence $s_1$, $s_2$, ... $s_t$ has the same probability $(\frac{1}{|F|^t})$ of occurring as the part of $B_2$ over the coordinates in $S$. So any values the algorithm receives from the positions labeled by members of $S$ are independent, uniformly random values from $F$. Therefore, we can construct a new algorithm $\tilde{A}$ that behaves exactly as $A$ except, whenever $A$ queries a member of $S$, $\tilde{A}$ samples uniformly at random from $F$ instead. Note that since we only consider nonadaptive decoding algorithms, without loss of generality, we can assume that the algorithm makes all the coin tosses before reading any values. Then over random $x$ and $B$ as above, when the coin tosses of $A$ and $\tilde{A}$ are fixed and equal, the distribution of values $A$ and $\tilde{A}$ receive as answers to their queries are the same. Thus, $\tilde{A}$ achieves the same correctness as $A$ under random $x$ and $B$. □

The distribution $D_R$ that we use will be of the following form. Let $0 \leq \beta \leq 1 - \frac{1}{|F|}$ be a value specified later. Choose a subset $Z \subseteq R$ of size $|Z| = \beta|R|$ uniformly at random from all possible subsets of $R$ of size $\beta|R|$. Next, independently for each position in $Z$ pick a value from $F^*$ uniformly with probability $\frac{1}{|F|-1}$. Let $B_1$ have the values picked this way in each position in $Z$ and the value zero in each position outside of $Z$.

Next we define the sets $R$ and $S$. Let $0 < \epsilon_1, \epsilon_2 < 1/2$ be values specified later. For $i \in [n]$, we define sets $R_i$ and $E_i$. We state two versions of the definition of the sets $R_i$ and $E_i$, one for arbitrary (possibly nonlinear) binary codes, the other for arbitrary linear codes. We could have given a common definition that includes both versions, but it is simpler to state a separate version for binary nonlinear codes. Notice that by a reasoning similar to Lemma 2.2 for binary linear codes the two definitions are equivalent.

For binary codes, for $i \in [n]$, define

$$R_i \triangleq \{j \in [m] \mid \left|\Pr_x[x_i = \mathbf{C}(x)_j] - \Pr_x[x_i \neq \mathbf{C}(x)_j]\right| \geq \epsilon_1\} .$$

For linear codes, define

$$R_i \triangleq \{j \in [m] \mid \exists c \in F^* : e_i = ca_j\}$$

Note that for binary linear codes this means that for $j \in R_i$, $\mathbf{C}(x)_j = x_i$.

For binary codes, for each $i \in [n]$, let $E_i$ be the set of pairs of indices $\{j_1, j_2\}$ such that

$$|\Pr_x[x_i = \mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2}] - \Pr_x[x_i \neq \mathbf{C}(x)_{j_1} + \mathbf{C}(x)_{j_2}]| \geq \epsilon_2 \ .$$

For linear codes over $F$, for each $i \in [n]$, let $E_i$ be the set of pairs of indices $\{j_1, j_2\}$ such that $e_i = c_1 a_{j_1} + c_2 a_{j_2}$, where $c_1, c_2 \in F^*$.

For each $i \in [n]$, let $M_i$ be a maximum set of disjoint pairs of indices $\{j_1, j_2\} \in E_i$.

Let $\alpha, \nu > 0$ be values specified later. Let

$$S_1 \triangleq \{i \in [n] \mid |R_i| \geq \nu m\} \qquad \text{and}$$
$$S_2 \triangleq \{i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m\}.$$

We will set the parameters so that we can upper bound $|S_1|$ and $|S_2|$ to ensure that $\bar{S_1} \cap \bar{S_2}$ is nonempty. Without loss of generality, assume $1 \in \bar{S_1} \cap \bar{S_2}$. That is, $|R_1| < \nu m$ and $|M_1| < \frac{\alpha}{2} m$.

Next we construct a set that covers each pair $\{j_1, j_2\} \in E_1$.

At this point, it is helpful to consider a graph over the set $[m]$ as vertices, having the pairs $\{j_1, j_2\} \in E_1$ as edges. We refer to this graph as $G$. Then, $M_1$ is a maximum matching in the graph $G$, and we are looking for a node cover of $G$. Let $\tilde{M} \subseteq [m]$ be the union of all positions that appear in a pair in $M_1$. Thus, $|\tilde{M}| = 2|M_1|$. Since $M_1$ is a maximum matching in $G$ this gives a node cover of $G$.

We can do better in the case of binary linear codes: then we can use a node cover of size $|M_1|$ instead of $2|M_1|$. This would be immediate if each vector $a \in \{0,1\}^n$ was used for at most one position of the code. In that case all pairs such that $e_1 = a_{j_1} + a_{j_2}$ would be disjoint and participate in $M_1$. In other words, if there are no vectors used for more than one position in a binary linear code, then the whole graph $G$ is a matching. However, there may be several positions in the code that correspond to the same vector $a \in \{0,1\}^n$. We get that in the case of binary linear codes, the graph $G$ described above is the union of a matching and possibly several complete bipartite graphs. Note that for nonbinary or nonlinear codes, the graph $G$ may not be a matching even if there are no vectors used for more than one position in the code. To obtain a node cover of size $|M_1|$ of $G$ for binary linear codes, we simply include into the cover one endpoint of each edge of the matching, and the positions corresponding to the smaller side of each complete bipartite graph.

To finish the definition of the distribution of the adversary, let $S = R_1 \cup \tilde{M}$. The set $R$ will be the complement $[m] \setminus S$ of the set $S$.

## 3.5 Proof of Theorem 3.1

Note that we only claim the bound when $\epsilon$ is large enough to give $\alpha > 0$. We will consider the probability distribution for corruption described in Section 3.4, using $\alpha \triangleq \delta - (\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}) - (3/n^{1/3} + \frac{36}{n})^{1/3} - \nu$. Recall that in Section 3.4 we considered

$$S_1 \triangleq \{i \in [n] \mid |R_i| \geq \nu m\} \qquad \text{and}$$
$$S_2 \triangleq \{i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m\}.$$

The sets $R_i$ and $M_i$ have been defined in Section 3.4, when describing the distribution of the adversary. For nonlinear codes, it is possible that a given index $j \in [m]$ belongs to several sets $R_i$: a nonlinear function can have nonzero correlation with more than one input bit. In the next claim we show that nevertheless, for appropriate choices of the parameters, the sets $R_i$ cannot overlap too much.

**Claim 3.6.** *Let $\nu = \frac{10}{n(1-H(1/2+\epsilon_1/2))}$. Then $|S_1| \leq 0.1n$ must hold.*

*Proof.* The claim immediately follows from Theorem 2.7, by noticing that the definition of the set $S_1$ implies that there is an index $j \in [m]$ that belongs to at least $\nu|S_1|$ of the sets $R_i$. Then applying Theorem 2.7 for the function $\mathbf{C}(x)_j$ gives $1 \geq (1 - H(1/2 + \epsilon_1/2))\nu|S_1|$. □

We let $\nu = \frac{10}{n(1-H(1/2+\epsilon_1/2))}$ and we use $\epsilon_1 = 1/n^{1/3}$ in the definition of the sets $R_i$. Note that for this setting of the parameters, $1 - H(1/2+\epsilon_1/2) \geq \frac{1}{n^{2/3}}$ for large enough $n$. This can be verified using Taylor expansion when evaluating $H(1/2 + 1/n^{1/3})$. Thus, $\nu = O(1/n^{1/3})$ holds.

If $|S_2| \geq 0.9n$, then we can use Theorem 2.6 to conclude $m \geq 2^{\frac{0.9}{4}\alpha^2\epsilon_2^2 n}$. Choosing $\epsilon_2 = 1/n^{1/3}$ gives $m \geq 2^{\frac{0.9}{4}\alpha^2 n^{1/3}}$.

If $|S_2| < 0.9n$, then $\bar{S_1} \cap \bar{S_2}$ is nonempty. Without loss of generality, assume $1 \in \bar{S_1} \cap \bar{S_2}$. Define the set $S$ as described in Section 3.4. Note that $|S| \leq (\alpha + \nu)m$ and that $\alpha + \nu < \delta$ since $\epsilon \leq \frac{1}{2}$. Define $\gamma \triangleq \frac{|[m] \setminus S|}{m}$ and let $\beta = \min(\frac{\delta - \alpha - \nu}{\gamma}, \frac{1}{2})$. We will use the distribution $D_S$ and the distribution $D_R$ with parameter $\beta$ as described in Section 3.4 to generate the corruption $B = B_1 + B_2$. Note that by our choice of parameters, $B$ contains at most $\delta m$ nonzero entries. Then, by Lemma 3.5, there is a 3-query decoding algorithm $A$ that never reads any positions from $S$ and satisfies the following:

$$\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) = x_1\right] \geq \frac{1}{2} + \epsilon \tag{1}$$

where the probability is over the internal coin tosses of $A$, uniform $x \in \{0,1\}^n$, and $B = B_1 + B_2$ chosen by the product distribution of the distributions $D_R$ and $D_S$.

We emphasize that the algorithm $A$ may not achieve the required correctness on every input and for every string within distance $\delta m$ of $\mathbf{C}(x)$. We only claim a bound on its correctness over uniformly random $x$ and over $B$ according to our distribution.

The *correlation* $Corr(f, h)$ between two Boolean functions $f$ and $h$ is defined as

$$Corr(f, h) = \Pr_x[f(x) = h(x)] - \Pr_x[f(x) \neq h(x)] \ .$$

**Claim 3.7.** *Let $g : \{0,1\}^q \to \{0,1\}$ be a Boolean function with $q$ variables. Let $h = g(h_1, \ldots, h_q)$, where $h, h_1, \ldots h_q : \{0,1\}^n \to \{0,1\}$ are Boolean functions with $n$ variables. Then*

$$|Corr(f,h)| \leq \sum_{S \subseteq [q]} |Corr(f, \sum_{i \in S} h_i)|$$

*Proof.* For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we denote by $f^*$ the function obtained from $f$ by replacing 0s by 1s and 1s by $-1$s, that is the function $f^* : \{1,-1\}^n \to \{1,-1\}$ such that for $y \in \{1,-1\}^n$, $f^*(y) = (-1)^{f((1-y)/2)}$. Then, $Corr(f,h) = Corr(f^*, h^*)$.

Let $\chi_S(y) = \prod_{i \in S} y_i$. Then the Fourier representation of $g^*$ gives $g^*(y) = \sum_{S \subseteq [q]} c_S \chi_S(y)$, where the constants $c_S = \hat{g^*}(S)$ are the Fourier coefficients. Using the Fourier representation of $g^*$ and noting that $|c_S| \leq 1$ for every $S \subseteq [q]$, we get $|Corr(f^*, h^*)| \leq \sum_{S \subseteq [q]} |Corr(f^*, \prod_{i \in S} h_i^*)|$. Translating back to 0/1 values, we get the desired inequality. □

We denote by $Q$ the event that the algorithm $A$ queries the positions in the query set $Q$, and by $g$ the event that the remaining coins of $A$ are fixed so that $A$ returns the value of the function $g$ evaluated on the positions read. We use the following notation:

$$P_{Q,g} = \Pr_{x,B,A} \left[ A^{\mathbf{C}(x)+B}(1) \neq x_1 | \; Q \cap g \right] \; .$$

With this notation, we have

$$\Pr_{x,B,A} \left[ A^{\mathbf{C}(x)+B}(1) \neq x_1 \right] = \sum_{Q,g} P_{Q,g} \Pr_{x,B,A} \left[ Q \cap g \right],$$

where in the summation, $Q \subseteq [m]$, $|Q| \leq 3$, and $g : \{0,1\}^{|Q|} \to \{0,1\}$.

Notice that for $Q = \{j_1, \ldots, j_{|Q|}\}$,

$$P_{Q,g} = \Pr_{x,B} \left[ g\left( (\mathbf{C}(x)+B)_{j_1}, \ldots, (\mathbf{C}(x)+B)_{j_{|Q|}} \right) \neq x_1 \right] \; ,$$

and we have $1 - 2P_{Q,g} = Corr(x_1, h)$, where $h(x,B) = g((\mathbf{C}(x)+B)_{j_1}, \ldots, (\mathbf{C}(x)+B)_{j_{|Q|}})$. Thus, we have $2P_{Q,g} = 1 - Corr(x_1,h) \geq 1 - |Corr(x_1,h)|$. Note that here we view $x_1$ as a function from $\{0,1\}^n \times \{0,1\}^m$ to $\{0,1\}$, taking the value $x_1$ on input $(x,B)$.

Recall the property of the algorithm $A$ that we work with, that $A$ never queries any positions from the set $S$ we defined when constructing the distribution of the adversary. By Claim 3.7, for query sets $Q$ of size at most 2 such that $Q \cap S = \emptyset$ we have $|Corr(x_1,h)| \leq 3(\epsilon_1 + \epsilon_2)$, and thus $P_{Q,g} \geq \frac{1}{2} - \frac{3}{2}(\epsilon_1 + \epsilon_2)$. The values $\epsilon_1$ and $\epsilon_2$ are as defined above.

Using Claim 3.7, for query sets $Q$ of size 3 such that $Q \cap S = \emptyset$ we get

$$|Corr(x_1,h)| \leq 3(\epsilon_1 + \epsilon_2) + |Corr(x_1, h')| \; ,$$

where $h'(x,B) = \sum_{j \in Q} (\mathbf{C}(x)+B)_j$.

As noted above, $2P_{Q,g} = 1 - Corr(x_1,h) \geq 1 - |Corr(x_1,h)|$. Thus, $2P_{Q,g} \geq 1 - 3(\epsilon_1 + \epsilon_2) - |Corr(x_1,h')|$.

Note that $|Corr(x_1, h')| = 1 - 2\min\{\Pr_{x,B}[x_1 \neq h'(x,B)], \Pr_{x,B}[x_1 = h'(x,B)]\}$. This gives

$$P_{Q,g} \geq \min\{\Pr_{x,B}[x_1 \neq h'(x,B)], \Pr_{x,B}[x_1 = h'(x,B)]\} - \frac{3}{2}(\epsilon_1 + \epsilon_2) \ . \tag{2}$$

We use the following claim to get a lower bound for this probability.

**Claim 3.8.** $\Pr_{x,B}\left[x_1 \neq \sum_{j\in Q}(\mathbf{C}(x)+B)_j\right] \geq \min_{k\in\{0,1\}} \Pr_B\left[\sum_{j\in Q} B_j = k\right]$, *and* $\Pr_{x,B}\left[x_1 = \sum_{j\in Q}(\mathbf{C}(x)+B)_j\right] \geq \min_{k\in\{0,1\}} \Pr_B\left[\sum_{j\in Q} B_j = k\right]$.

*Proof.* To prove the first inequality, it is enough to show that

$$\Pr_{x,B}\left[x_1 \neq \sum_{j\in Q}(\mathbf{C}(x)+B)_j \mid \sum_{j\in Q} B_j = 0\right] + \Pr_{x,B}\left[x_1 \neq \sum_{j\in Q}(\mathbf{C}(x)+B)_j \mid \sum_{j\in Q} B_j = 1\right] = 1 \ .$$

We have $\Pr_{x,B}\left[x_1 \neq \sum_{j\in Q}\mathbf{C}(x)_j + k \mid \sum_{j\in Q} B_j = k\right] = \Pr_x\left[x_1 \neq \sum_{j\in Q}\mathbf{C}(x)_j + k\right]$. It remains to note that for each $x$ there is exactly one value $k \in \{0,1\}$ that gives

$$\sum_{j\in Q}\mathbf{C}(x)_j + k = x_i \ .$$

The second inequality follows by an analogous argument. □

Using Claim 5.7, we get that

$$\Pr_B[\sum_{j\in Q} B_j = 0] \geq (1-\beta)^3 + 3\beta^2(1-\beta) - 2\frac{9}{\gamma m}$$

and

$$\Pr_B[\sum_{j\in Q} B_j = 1] \geq 3\beta(1-\beta)^2 + \beta^3 - 2\frac{9}{\gamma m} \ .$$

Note that $\gamma m > (1-\delta)m$ by our choices of the parameters. By Observation 2.1, we can assume without loss of generality that $\delta < 1/2$. Since $\beta \leq 1/2$, and by (2) this implies for $Q$ such that $Q \cap S = \emptyset$ and any $g$ that

$$P_{Q,g} \geq 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{m} - \frac{3}{2}(\epsilon_1 + \epsilon_2) \ .$$

For large enough $n$, $m > n$ must hold, for example by using the lower bounds in [12]. This implies the following inequality:

$$P_{Q,g} > 3\beta(1-\beta)^2 + \beta^3 - \frac{36}{n} - \frac{3}{2}(\epsilon_1 + \epsilon_2) \ . \tag{3}$$

By the choice of our parameters, this expression evaluated at $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$ is $\geq \frac{1}{2} - \epsilon$. See Claim 5.2 for details, substituting $\psi(n)^3 = 36/n + 3/2(\epsilon_1 + \epsilon_2) = 36/n + 3/n^{1/3}$. This implies that if $|S_2| < 0.9n$, which allowed us to construct the above distribution for the corruption $B$, then $\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) = x_1\right] < \frac{1}{2} + \epsilon$. This however contradicts the estimate (1) about the average correctness of the algorithm $A$, which we derived based on $\mathbf{C}$ being a $(3, \delta, \epsilon)$-LDC. Thus, it must be the case that $|S_2| \geq 0.9n$, and we can use Theorem 2.6 to conclude the proof of the theorem.

## 3.6 Proof of Theorem 3.3

The statement trivially holds when $\alpha \leq 0$, thus we assume $\alpha > 0$. We will consider the probability distribution for corruption described in Section 3.4, using $\alpha \triangleq \delta - (1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}) - (\frac{108|F|}{n})^{1/3} - \frac{10}{n}$. Let $\nu = \frac{10}{n}$. Recall that in Section 3.4 we considered

$$S_1 \triangleq \{i \in [n] \mid |R_i| \geq \nu m\} \qquad \text{and}$$
$$S_2 \triangleq \{i \in [n] \mid |M_i| \geq \frac{\alpha}{2} m\}.$$

The sets $R_i$ and $M_i$ have been defined in Section 3.4, when describing the distribution of the adversary. Note that $|S_1| \leq \frac{m}{\frac{10m}{n}} = 0.1n$ must hold, since for linear codes each position $j \in m$ participates in at most one of the sets $R_i$.

If $|S_2| \geq 0.9n$, then we can use Theorem 2.5 to conclude $m \geq 2^{\frac{0.9}{2}\alpha n - 1}$.

If $|S_2| < 0.9n$, then $\bar{S_1} \cap \bar{S_2}$ is nonempty. Without loss of generality, assume $1 \in \bar{S_1} \cap \bar{S_2}$. Define the set $S$ as described in Section 3.4. Note that $|S| \leq (\alpha + \nu)m$ and that $\alpha + \nu < \delta$ since $\epsilon \leq 1 - \frac{1}{|F|}$. Define $\gamma \triangleq \frac{|[m] \setminus S|}{m}$ and let $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, 1 - \frac{1}{|F|})$. We will use the distribution $D_S$ and the distribution $D_R$ with parameter $\beta$ as described in the previous section to generate the corruption $B = B_1 + B_2$. Note that by our choice of parameters, $B$ contains at most $\delta m$ nonzero entries. Then, by Lemma 3.5, there is a 3-query decoding algorithm $A$ that never reads any positions from $S$ and satisfies the following:

$$\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) = x_1\right] \geq \frac{1}{|F|} + \epsilon \tag{4}$$

where the probability is over the internal coin tosses of $A$, uniform $x \in F^n$, and $B = B_1 + B_2$ chosen by the product distribution of the distributions $D_R$ and $D_S$.

Recall that we denote by $Q$ the event that the algorithm $A$ queries the positions in the query set $Q$. For a given set $Q$ of size $q$, we will denote by $z$ the event that the values of $\mathbf{C}(x) + B$ on the positions in $Q$ are equal to $z \in F^q$ that is, the event that $(\mathbf{C}(x) + B)_{j_i} = z_i$, for $i \in [q]$ and $j_i \in Q$.

We will estimate the probability that the algorithm returns an incorrect value, under the conditions that the algorithm queries a specific query set $Q$, that a fixed linear combination of the value of the corruption on this specific query set is equal to a fixed value, and that the values $z$ on the positions of the given query set are fixed. We use the following notation for conditional probabilities of this type. First, define

$$P_Q = \Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) \neq x_1 |\ Q\right] .$$

Next, let $c_j \in F$ for $j \in Q$, such that at least one coefficient $c_j$ is nonzero. Note that at this point we only need that at least one coefficient is nonzero, so that the events in the conditions below have nonzero probability. Define

$$P_{Q,k} = \Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) \neq x_1 |\ Q \cap (\sum_{j\in Q} c_j B_j = k)\right] ,$$

$$P_{Q,k,z} = \Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) \neq x_1 |\ Q \cap (\sum_{j\in Q} c_j B_j = k) \cap z\right] .$$

With this notation, we have

$$P_Q = \sum_{k\in F} P_{Q,k} \Pr_{x,B,A}[(\sum_{j\in Q} c_j B_j = k) \mid Q] ,$$

Furthermore,

$$P_{Q,k} = \sum_{z\in F^q} P_{Q,k,z} \Pr_{x,B,A}[z \mid Q \cap (\sum_{j\in Q} c_j B_j = k)] .$$

Note that our distribution generating the random corruption $B$, the random choices of the algorithm, and the input $x$ are independent of each other; because of the way we constructed the distribution $B$ and since $A$ is nonadaptive. Also, note that $(\sum_{j\in Q} c_j B_j = k)$ and $Q$ are independent events: the number of positions the adversary corrupts in a given set of positions is independent of whether or not the algorithm actually looks at those positions. Thus,

$$\Pr_{x,B,A}[(\sum_{j\in Q} c_j B_j = k) \mid Q] = \Pr_{B}[\sum_{j\in Q} c_j B_j = k] .$$

We get that

$$P_Q = \sum_{k\in F} P_{Q,k} \Pr_{B}[(\sum_{j\in Q} c_j B_j = k)] . \tag{5}$$

Similarly, the values of $\mathbf{C}(x) + B$ in the positions of $Q$ as well as the number of corrupted positions are independent of whether or not the algorithm looks at those positions. Thus,

$$\Pr_{x,B,A}[z \mid Q \cap (\sum_{j\in Q} c_j B_j = k)] = \Pr_{x,B}[z \mid (\sum_{j\in Q} c_j B_j = k)] .$$

For query sets $Q$ that correspond to linearly independent vectors, we can say even more. Fact 2.3 immediately gives the following.

**Claim 3.9.** *Let $Q$ be a query set such that the vectors $a_j$ for $j \in Q$ are linearly independent. Let $q = |Q|$. Then for any $z \in F^q$ we have*

$$\Pr_{x,B}[z \mid (\sum_{j\in Q} c_j B_j = k)] = 1/|F|^q .$$

The next two lemmas allow us to obtain a simple estimate on the error probability of the algorithm on certain query sets over random input $x$ and over our distribution $B$ on the corruption of the adversary. We emphasize that the lemmas hold without any assumptions on the algorithm (except its nonadaptiveness), in particular even if the algorithm uses nonlinear operations to derive its output from the values it reads.

**Lemma 3.10.** *Let $Q$ be a query set such that the vectors $a_j$ for $j \in Q$ are linearly independent, and such that $e_1 \in \mathrm{span}(Q)$. Let $q = |Q|$. Then, for any $z \in F^q$, $\sum_{k \in F} P_{Q,k,z} = |F|-1$, and $\sum_{k \in F} P_{Q,k} = |F| - 1$.*

*Proof.* Let $e_1 = \sum_{j \in Q} c_j a_j$. Note that at least one of the coefficients $c_j$ used is nonzero, since $e_1$ is a nonzero vector. Regardless of what is the decoding algorithm, this means that the algorithm is incorrect whenever it outputs a different value than $\sum_{j \in Q} c_j (\mathbf{C}(x) + B)_j - \sum_{j \in Q} c_j B_j$. Thus, for $z$ such that $\sum_{i \in [q]} c_{j_i} z_i = \ell$, where $j_i \in Q$ for $i \in [q]$, we have

$$P_{Q,k,z} = \Pr_{x,B,A} \left[ A^{\mathbf{C}(x)+B}(1) \neq \ell - k \mid Q \cap (\sum_{j \in Q} c_j B_j = k) \cap z \right] .$$

Notice that the value of the output of the algorithm $A$ is completely determined by the choice of $Q$, the remaining random choices of $A$, and the values $(\mathbf{C}(x) + B)_j$ for $j \in Q$. In addition, the event $(\sum_{j \in Q} c_j B_j = k)$ is independent of the distribution for the coin tosses of the algorithm. Thus, for $z$ such that $\sum_{i \in [q]} c_{j_i} z_i = \ell$, where $j_i \in Q$ for $i \in [q]$,

$$P_{Q,k,z} = \Pr_{x,B,A} \left[ A^{\mathbf{C}(x)+B}(1) \neq \ell - k \mid Q \cap z \right] .$$

The statement of the Lemma now follows by noting that for any $\ell \in F$,

$$\sum_{k \in F} \Pr_{x,B,A} \left[ A^{\mathbf{C}(x)+B}(1) \neq \ell - k \mid Q \cap z \right] = |F| - 1 ,$$

and using Claim 3.9. □

**Lemma 3.11.** *Let $Q$ be a query set such that the vectors $a_j$ for $j \in Q$ are linearly independent, and such that $e_1 = \sum_{j \in Q} c_j a_j$. Then*

$$P_Q \geq (|F| - 1) \min_{k \in F} \Pr_B [(\sum_{j \in Q} c_j B_j = k)] .$$

*Proof.* Immediately follows by Lemma 3.10 and equation (5). □

**Corollary 3.12.** *Let $Q$ be a query set such that the vectors $a_j$ for $j \in Q$ are linearly independent, and such that $e_1 = \sum_{j \in Q} c_j a_j$. Suppose $q'$ of the coefficients $c_j$ are nonzero, where $q' \leq |Q|$. Then*

$$P_Q \geq q' \delta - o(\delta) - O(1/n) .$$

*Proof.* Immediately follows by Lemma 3.11 and the estimates in the Appendix. □

Notice that up to this point, our argument is valid for arbitrary number of queries for linear codes. Moreover, the decoder is allowed to use nonlinear operations in the above statements. The rest of our proof for 3-query linear codes is based on Lemma 3.11, and showing that its conditions must be satisfied by the decoder $A$ that we consider. We observe that for the particular 3-query algorithm $A$ that we consider, the following claim holds.

**Claim 3.13.** *Let $Q$ be a query set that the algorithm $A$ queries with nonzero probability, such that $e_1 \in \mathrm{span}(Q)$. Then, $|Q| = 3$, and for the positions $j_1, j_2, j_3 \in Q$, we have $e_1 = c_{j_1}a_{j_1} + c_{j_2}a_{j_2} + c_{j_3}a_{j_3}$ where $c_j \in F^*$ for $j \in Q$. Moreover, the vectors $a_{j_1}$, $a_{j_2}$ and $a_{j_3}$ are linearly independent.*

*Proof.* As noted above (see the sentence of inequality (4)), the algorithm $A$ never reads any positions from $S$. Notice that $S$ was constructed so that it contains at least one index from each set $Q \subseteq [m]$ such that $|Q| \leq 2$ and $e_1 \in \mathrm{span}(Q)$. This means that a set $Q$ such that $e_1 \in \mathrm{span}(Q)$ can be queried by $A$ with nonzero probability, only when $|Q| = 3$, and $e_1$ is not spanned by any strict subsets of $Q$. This implies the statement of the claim. □

Next we note that by Lemma 2.2, $P_Q \geq 1 - 1/|F|$ for query sets $Q$ such that $e_1 \notin \mathrm{span}(Q)$. Thus, by Claim 3.13 it remains to estimate $P_Q$ for $Q$ such that $|Q| = 3$, $e_1 = c_{j_1}a_{j_1} + c_{j_2}a_{j_2} + c_{j_3}a_{j_3}$, where each $c_{j_i}$ is nonzero and the vectors $a_{j_1}$, $a_{j_2}$ and $a_{j_3}$ are linearly independent. Let $Q$ be such a set. We use Lemma 3.11 to estimate $P_Q$. Note that the probability in the bound of Lemma 3.11 is largest if all the coefficients $c_j$ used in the linear combination are nonzero. We give a precise estimate on this probability in Claim 5.8.

Since $\beta \leq 1 - 1/|F|$, the estimate in Claim 5.8 implies

$$P_Q > 3\beta(1-\beta)^2 + (1 - \frac{1}{|F|-1})3\beta^2(1-\beta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3 - 3\frac{9}{\gamma m} .$$

Note that $\gamma m > (1-\delta)m$ by our choices of the parameters. By Observation 2.1, we can assume without loss of generality that $\delta < 1 - 1/|F|$, and therefore $\gamma > 1/|F|$. Recall that for large enough $n$, by the lower bounds of [12] we can assume that $m > n$. Thus, we have

$$P_Q > 3\beta(1-\beta)^2 + (1 - \frac{1}{|F|-1})3\beta^2(1-\beta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3 - \frac{27|F|}{n} .$$

We then use Claim 5.12, substituting $\psi(n) = (108|F|/n)^{1/3}$, that is $\psi(n)^3/4 = 27|F|/n)$. This shows that by our choice of the parameters the above expression evaluated at $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, 1 - 1/|F|)$ is $\geq 1 - \frac{1}{|F|} - \epsilon$. This implies that if $|S_2| < 0.9n$, which allowed us to construct the above distribution for the corruption $B$, then

$$\Pr_{x,B,A}\left[A^{\mathbf{C}(x)+B}(1) = x_1\right] < \frac{1}{|F|} + \epsilon .$$

This however contradicts the estimate (4) about the average correctness of the algorithm $A$, which we derived based on $\mathbf{C}$ being a $(3, \delta, \epsilon)$-LDC. Thus, it must be the case that $|S_2| \geq 0.9n$, and we can use Theorem 2.5 to conclude the proof of the theorem.

# 4 Linear Decoders

**Definition 2.** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be an arbitrary (possibly nonlinear) code. We say that an algorithm* $A$ *is a* linear decoder *for* $\mathbf{C}$ *if for any fixing of the outcomes of the coin flips of* $A$*, the value it returns is a fixed linear combination of the codeword positions it reads.*

Note that this definition allows the decoder $A$ to ignore the code positions read and just return the result of a coin flip, thus the correctness of linear decoders can still be always at least $1/|F|$. Also, note that the definition allows the possibility of considering adaptive linear decoders, the decoder in principle could be adaptive when choosing the set $Q$, but the coefficients in the linear combination used cannot depend on the values read (that would allow to simulate nonlinear operations). However, for our proofs we only consider nonadaptive linear decoders. Note also that nonlinear codes may also have linear decoders.

The next lemma yields lower bounds on the error probability of nonadaptive linear decoders for arbitrary codes, and arbitrary probability distributions of the corruption $B$. We denote by $Q$ the event that the decoder reads the positions in the query set $Q$, and by $g$ the event that the remaining coins of $A$ are fixed so that $A$ returns the value of the function $g$ evaluated on the positions read.

**Lemma 4.1.** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be an arbitrary (possibly nonlinear) code, and let* $A$ *be a nonadaptive linear decoder operating on* $\mathbf{C}$*. Consider an arbitrary adversary generating the vectors* $B \in F^m$*. Let* $Q$ *be a query set, such that when* $A$ *is trying to recover* $x_i$ *there is a fixing* $g$ *of the remaining coin tosses of* $A$ *(other than the coins to choose the set* $Q$*) such that* $A$ *returns a fixed linear combination with nonzero coefficients of the positions in* $Q$*. Let* $c_j \in F^*$ *be the coefficient used on position* $j \in Q$*. Then*

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) \neq x_i \mid Q \cap g] \geq (|F|-1) \min_{k \in F} \Pr_{x,B,A}[\sum_{j \in Q} c_j B_j = k \mid Q \cap g] \, .$$

*Proof.* It is enough to show that

$$\sum_{k \in F} \Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) \neq x_i \mid Q \cap g \cap (\sum_{j \in Q} c_j B_j = k])] = |F| - 1 \, .$$

For the decoder described in the statement of the lemma, for fixed $Q$ and $g$, the output of the algorithm $A^{\mathbf{C}(x)+B}(i) = \sum_{j \in Q} c_j (\mathbf{C}(x)+B)_j = \sum_{j \in Q} c_j \mathbf{C}(x)_j + \sum_{j \in Q} c_j B_j$. Thus,

$$\begin{aligned}
&\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) \neq x_i \mid Q \cap g \cap (\sum_{j \in Q} c_j B_j = k])] \\
&\quad = \Pr_{x.B,A}[\sum_{j \in Q} c_j \mathbf{C}(x)_j + k \neq x_i \mid Q \cap g \cap (\sum_{j \in Q} c_j B_j = k])] \\
&\quad = \Pr_{x}[\sum_{j \in Q} c_j \mathbf{C}(x)_j + k \neq x_i] \, .
\end{aligned}$$

The last equation holds since the event $\sum_{j\in Q} c_j \mathbf{C}(x)_j + k \neq x_i$ does not involve $B$ or the coins of $A$. It remains to note that for each $x$ there is exactly one value $k \in F$ that gives $\sum_{j\in Q} c_j \mathbf{C}(x)_j + k = x_i$. Thus, we have

$$\sum_{k\in F} \Pr_x[\sum_{j\in Q} c_j \mathbf{C}(x)_j + k \neq x_i] = |F| - 1 \ .$$

□

We think that it is interesting that Lemma 4.1 holds for possibly nonlinear codes, and for arbitrary adversary. Moreover, under a particular distribution of the adversary, this implies that the probability of error of a linear decoder is lower bounded by the probability that the corruption sums to a nonzero value over the given linear combination used by the decoder. This is of course straightforward for linear codes for arbitrary adversary, but we show that for certain distributions of the adversary it holds even for arbitrary nonlinear codes.

**Lemma 4.2.** *Let $\mathbf{C}$: $F^n \to F^m$ be an arbitrary (possibly nonlinear) code, and let $A$ be a nonadaptive linear decoder operating on $\mathbf{C}$. Let $B$ be chosen from a distribution that first uniformly chooses a set of $\beta m$ coordinates from $[m]$, and then assigns a uniformly random value from $F^*$ independently to each chosen coordinate. The remaining $m - \beta m$ coordinates are set to 0. Let $Q$ be a query set, such that when $A$ is trying to recover $x_i$ there is a fixing $g$ of the remaining coin tosses of $A$ (other than the coins to choose the set $Q$) such that $A$ returns a fixed linear combination with nonzero coefficients of the positions in $Q$. Let $c_j \in F^*$ be the coefficient used on position $j \in Q$. Then*

$$\Pr_{x,B,A}[A^{\mathbf{C}(x)+B}(i) \neq x_i \mid Q \cap g] \geq \Pr_B[\sum_{j\in Q} c_j B_j \neq 0] \ .$$

*Proof.* First note that

$$\Pr_{x,B,A}[\sum_{j\in Q} c_j B_j = k \mid Q \cap g] = \Pr_B[\sum_{j\in Q} c_j B_j = k] \ ,$$

since the event $Q \cap g$ only depends on the coin flips of the algorithm, and the distribution of the adversary does not depend on the input or the coin flips of the algorithm. Then, the statement follows from Lemma 4.1 and Claim 5.10. □

This Lemma allows us to prove limits on the correctness achievable by linear decoders for arbitrary (possibly nonlinear) codes and regardless the length of the code.

Recall that in Section 3.2 we defined $P(\delta, q, F) \triangleq \Pr_B\left[(\sum_{j\in Q} c_j B_j = 0)\right]$, where the probability is over $B \in F^m$ randomly chosen according to a distribution that first chooses to corrupt each coordinate in $[m]$ independently with probability $\delta$, and then uniformly and independently assigns a value from $F^*$ to each chosen coordinate of $B$. The remaining coordinates of $B$ are set to 0. While we use a different distribution in the proofs, it is convenient to state our bounds in terms of $P(\delta, q, F)$.

**Theorem 4.3.** *Let $\mathbf{C}$: $F^n \to F^m$ be an arbitrary (possibly nonlinear) code, and let $A$ be a nonadaptive linear decoder operating on $\mathbf{C}$, such that it always returns a linear combination with exactly $q$ nonzero coefficients. For large enough $n$, the correctness of such decoder cannot be larger than $P(\delta, q, F) + O(1/n) = 1 - q\delta + o(\delta) + O(1/n)$, regardless of the length of the code.*

*Proof.* The proof follows by Lemma 4.2 and Claim 5.7. □

Moreover, we obtain the following bound on the correctness of arbitrary linear decoders, for any number of nonzero coefficients used.

**Theorem 4.4.** *Let $\mathbf{C}$: $F^n \to F^m$ be an arbitrary (possibly nonlinear) code. For large enough $n$, the correctness of any nonadaptive linear decoder operating on $\mathbf{C}$ is at most $\max\{1/|F|, P(\delta, 2, F) + O(1/n)\} = \max\{1/|F|, 1 - 2\delta + \frac{|F|}{(|F|-1)}\delta^2 + O(1/n)\}$, regardless of the length of the code.*

*Proof.* As in the proof of Theorem 3.1, we can show using Theorem 2.7 that in any LDC, the number of codeword positions that have large correlation with a given input bit $x_i$ over random input $x$ must be small for most input positions $i \in [n]$. We use a similar distribution for the adversary as in the proof of Theorem 3.1, letting the set $S$ be the set of codeword positions that have large correlation with $x_i$ for an index $i$ where this set is small. We let the adversary corrupt the positions in $S$ independently. Then Lemma 3.4 implies that there is a linear decoding algorithm that only uses query sets that either provide only small advantage over random guessing, or returns a linear combination of the positions read with at least 2 nonzero coefficients. Then the statement follows by Lemma 4.2 and Claim 5.7. □

Combining Theorem 4.3 with our proof of lower bounds for 3-query codes shows that for any binary code (possibly nonlinear), and for linear codes over arbitrary finite fields, if a nonadaptive $q$-query linear decoder (with query sets of size at most $q$) achieves correctness more than $P(\delta, 3, F) + O(1/n^{1/3})$, then the length of the code must be exponential.

As we noted in the introduction, in the case of linear *smooth codes* (see [12]), requiring the decoders to be linear is inconsequential: for linear codes, if any algorithm gives nontrivial advantage over random guessing when querying a given set of codeword positions $Q$, then by Lemma 2.2, $e_i \in \text{span}(Q)$ must hold. Thus, there is a fixed linear combination of the positions in $Q$ that gives the correct value of $x_i$ for any input $x$. Using the same procedure as the original decoder to choose which positions to query and then returning this fixed linear combination (if it exists) cannot violate the smoothness of the code. This means that any reduction that converts an arbitrary smooth code with given parameters to a locally decodable code with corresponding parameters, would yield a locally decodable code with a linear decoder that achieves the parameters guaranteed by the reduction. Thus, the above lemma about linear decoders also implies that there is no significantly better general reduction from smooth codes to locally decodable codes than the current bounds giving at most $1 - q\delta$ correctness for $q$ query locally decodable codes.

## 4.1 Matching Sum Decoders

Matching sum decoders were formally defined by Woodruff [24]. A $q$-query matching sum decoder picks a set of size $q$ uniformly at random from a collection of sets that form a matching in the complete $q$-uniform hypergraph, whose vertices correspond to the positions of the codeword. Then, the decoder reads the positions corresponding to the chosen set and returns a fixed linear combination with only nonzero coefficients of the positions read. Most known constructions of locally decodable codes have such decoders.

The following lemma holds for any code, regardless the length of the codewords.

**Lemma 4.5.** *Let* $\mathbf{C}$*:* $F^n \to F^m$ *be an arbitrary (possibly nonlinear) code, and let* $A$ *be a* $q$*-query matching sum decoder operating on* $\mathbf{C}$*. The correctness of the matching sum decoder* $A$ *is at most* $1 - q\delta$*.*

*Proof.* Consider an adversary that randomly chooses $q\delta$ fraction of the sets that participate in the matching, and corrupts exactly one position from each chosen set. Since the size of the matching is at most $m/q$, where $m$ is the length of the code, this way the number of corrupted positions is always at most $\delta m$. The decoder will return an incorrect value every time it picks a set that was also chosen by the adversary. This happens with probability at least $q\delta$. □

Combining this lemma with our proof of lower bounds for 3-query codes shows that for any binary code (possibly nonlinear), and for linear codes over arbitrary finite fields, if a matching sum decoder with query sets of size at most $q$ achieves correctness more than $1 - 3\delta + O(1/n)$, then the length of the code must be exponential.

**Acknowledgements** We thank David Woodruff for helpful conversations and for pointing us to [5]. We also thank the anonymous referees and Mahdi Cheraghchi for helpful comments.

# References

[1] N. Alon, L. Babai and A. Itai: A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, Vol. 7, pp. 567 - 583, 1986.

[2] L. Babai, L. Fortnow, L. Levin, M. Szegedy: Checking computations in polylogarithmic time. In *Proceedings of STOC 1991*, pp. 21-31.

[3] D. Beaver and J. Feigenbaum: Hiding instances in multioracle queries. In *Proceedings of STACS 1990*, pp. 37-48.

[4] A. Ben-Aroya, K. Efremenko, A. Ta-Shma: Local list-decoding with a constant number of queries. In *Proceedings of FOCS 2010*, pp. 715-722.

[5] A. Ben-Aroya, O. Regev and R. de Wolf: A hypercontractive inequality for matrix valued functions with applications to quantum computing and LDCs. In *Proceedings of FOCS 2008*, pp. 477 - 486.

[6] Z. Dvir: On matrix rigidity and locally self-correctable codes. In *Proceedings of IEEE Conference on Computational Complexity 2010*, pp. 291-298.

[7] Z. Dvir, P. Gopalan, S. Yekhanin: Matching Vector Codes. In *Proceedings of FOCS 2010*, pp. 705 - 714.

[8] Z. Dvir and A. Shpilka: Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *Proceedings of STOC 2005*, pp. 592 - 601.

[9] K. Efremenko: 3-query locally decodable codes of subexponential length. In *Proceedings of STOC 2009*, pp. 39-44.

[10] A. Gál, A. Mills: Three query locally decodable codes with higher correctness require exponential length. In *Proceedings of STACS 2011*, pp. 673–684.

[11] O. Goldreich, H. Karloff, L. Schulman and L. Trevisan: Lower bounds for linear locally decodable codes and private information retrieval. *Comput. Complex.*, 15(3):263–296, 2006.

[12] J. Katz and L. Trevisan: On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of STOC 2000*, pp. 80-86.

[13] I. Kerenidis and R. de Wolf: Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of STOC 2003*, pp. 106-115.

[14] S. Kopparty, A. Saraf and S. Yekhanin: High-rate codes with sublinear-time decoding. ECCC Report No. 148, 2010.

[15] P. Bro Miltersen: Cell probe complexity - a survey. In *Advances in Data Structures Workshop, 1999*.

[16] K. Obata: Optimal lower bounds for 2-query locally decodable linear codes. In *Proceedings of RANDOM 2002* pp. 39–50.

[17] P. Raghavendra: A note on Yekhanin's locally decodable codes. *ECCC TR07-016*, 2007.

[18] D. Shiowattana and S. V. Lokam: An optimal lower bound for 2-query locally decodable linear codes. *Inf. Process. Lett.*, 97(6):244–250, 2006.

[19] M. Sudan, L. Trevisan, S. Vadhan: Pseudorandom generators without the XOR lemma. In *Proceedings of STOC 1999*, pp. 537-546.

[20] L. Trevisan: Some applications of coding theory in computational complexity. *ECCC TR04-043*, 2004.

[21] S. Wehner and R. de Wolf: Improved lower bounds for locally decodable codes and private information retrieval. In *Proceedings of ICALP 2005* Vol. 3580 of LNCS, pp. 1424 - 1436.

[22] R. de Wolf: Error-correcting data structures In *Proceedings of STACS 2009* pp. 313 - 324.

[23] D. Woodruff: Some new lower bounds for general locally decodable codes. *ECCC TR07-006*, 2006.

[24] D. Woodruff: Corruption and recovery-efficient locally decodable codes. In *Proceedings of RANDOM 2008*, pp. 584–595.

[25] D. Woodruff: A Quadratic Lower Bound for Three-Query Linear Locally Decodable Codes over Any Field. In *Proceedings of RANDOM 2010*.

[26] S. Yekhanin: Towards 3-query locally decodable codes of subexponential length. In *Proceedings of STOC 2007*, pp. 266–274.

# 5 Appendix

## 5.1 Calculations

**Claim 5.1.** *Let* $0 < \epsilon \leq 1/2$ *and* $\alpha \triangleq \delta - (\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}) - \frac{\phi(n)}{4}$. *Then* $\alpha > 0$ *when* $\frac{1}{2} + \epsilon > 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n)$. *Moreover,* $\alpha > \frac{\mu}{4}$ *when* $\frac{1}{2} + \epsilon > \mu + 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n)$ *for some* $\mu \geq 0$.

*Proof.* Let $f(\beta) = 3\beta(1-\beta)^2 + \beta^3$. Let $\xi = \frac{1}{2} - (\frac{\epsilon}{4})^{1/3} = \delta - \alpha - \frac{\phi(n)}{4}$. First observe that $f(\xi) = \frac{1}{2} - \epsilon$. By Fact 5.4

$$f(\xi + \frac{\phi(n)}{4}) \leq \frac{1}{2} - \epsilon + \phi(n) \; .$$

The assumption $\frac{1}{2} + \epsilon > 1 - 3\delta + 6\delta^2 - 4\delta^3 + \phi(n)$ means that

$$f(\delta) > \frac{1}{2} - \epsilon + \phi(n) \; .$$

But $\delta = \xi + \frac{\phi(n)}{4} + \alpha$, and $f$ is a monotone increasing function, which means that $\alpha > 0$ must hold.

To see the second statement of the claim, observe that by Fact 5.4

$$f(\delta) = f(\xi + \frac{\phi(n)}{4} + \alpha) \leq \frac{1}{2} - \epsilon + \phi(n) + 4\alpha \; .$$

□

**Claim 5.2.** *Let $0 < \epsilon \leq 1/2$ and $\alpha \triangleq \delta - (\frac{1}{2} - (\frac{\epsilon}{4})^{1/3}) - \psi(n) - \nu$. Let $f(\beta) = 3\beta(1-\beta)^2 + \beta^3$. Let $n$ be large enough, so that $\psi(n)^3 < \epsilon$. Then the expression $f(\beta) - \psi(n)^3$ evaluated at $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$ is $\geq \frac{1}{2} - \epsilon$.*

*Proof.* If $\min(\frac{\delta-\alpha-\nu}{\gamma}, \frac{1}{2})$ is equal to $\frac{1}{2}$, then we have $f(\beta) = f(\frac{1}{2}) = \frac{1}{2}$. For large enough $n$, $\psi(n)^3 < \epsilon$ and thus $f(\beta) - \psi(n)^3 > \frac{1}{2} - \epsilon$.

Otherwise, since $f$ is monotone, $f(\beta) \geq f(\delta - \alpha - \nu)$. Let $\xi = \frac{1}{2} - (\frac{\epsilon}{4})^{1/3}$. Recall that $f(\xi) = \frac{1}{2} - \epsilon$ and $\delta - \alpha - \nu = \xi + \psi(n)$. Thus, by Fact 5.3 $f(\beta) \geq f(\delta - \alpha - \nu) \geq \frac{1}{2} - \epsilon + \psi(n)^3$. □

We use the following estimates on the function $f(\beta)$.

**Fact 5.3.** *Let $f(\beta) \triangleq 3\beta(1-\beta)^2 + \beta^3$. For any $\beta$ and $\rho \geq 0$, $f(\beta + \rho) \geq f(\beta) + \rho^3$.*

*Proof.* Let $x \triangleq \beta + \rho/2$ and $p \triangleq \rho/2$. So now we need to lower bound $f(x+p) - f(x-p)$. Note that $f(x)$ can be expressed as $3x - 6x^2 + 4x^3$. Therefore,

$$\begin{aligned}
f(x+p) - f(x-p) &= (3(x+p) - 6(x+p)^2 + 4(x+p)^3) - (3(x-p) - 6(x-p)^2 + 4(x-p)^3) \\
&= 3((x+p) - (x-p)) - 6((x+p)^2 - (x-p)^2) + 4((x+p)^3 - (x-p)^3) \\
&= 3(2p) - 6(4xp) + 4(6x^2p + 2p^3) \\
&= 6p - 24xp + 24x^2p + 8p^3 \\
&= 3\rho - 12x\rho + 12x^2\rho + \rho^3 \\
&= 3\rho(1-2x)^2 + \rho^3 \\
&\geq \rho^3
\end{aligned}$$

□

**Fact 5.4.** *Let $f(\beta) \triangleq 3\beta(1-\beta)^2 + \beta^3$. For any $\beta$ and $0 \leq \rho \leq 1$ such that $0 \leq \beta + \frac{\rho}{2} \leq 1$, $f(\beta + \rho) \leq f(\beta) + 4\rho$.*

*Proof.* We can use the same notation and the same first several steps of the Fact 5.3 to get

$$\begin{aligned}
f(x+p) - f(x-p) &= 3\rho(1-2x)^2 + \rho^3 \\
&\leq 3\rho + \rho^3 \\
&\leq 4\rho
\end{aligned}$$

□

The above two claims extend to the nonbinary case for the corresponding function $Z(\beta)$ as follows.

**Fact 5.5.** *Let $Z(\beta) \triangleq 3\beta(1-\beta)^2 + (1 - \frac{1}{|F|-1})3\beta^2(1-\beta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3$. For any $\beta$ and $\rho \geq 0$, $Z(\beta + \rho) \geq Z(\beta) + \frac{\rho^3}{4}$.*

*Proof.* For easier notation, let $c \triangleq \frac{1}{|F|-1}$. We can rewrite $Z(\beta)$:

$$\begin{aligned} Z(\beta) &= 3\beta(1-\beta)^2 + (1-c)3\beta^2(1-\beta) + (1-c+c^2)\beta^3 \\ &= 3\beta - 6\beta^2 + 3\beta^3 + (1-c)3\beta^2 - (1-c)3\beta^3 + (1-c+c^2)\beta^3 \\ &= 3\beta + (-3-3c)\beta^2 + (1+2c+c^2)\beta^3 \\ &= 3\beta - 3d\beta^2 + d^2\beta^3 \qquad \text{where } d \triangleq 1+c \end{aligned}$$

Let $x \triangleq \beta + \rho/2$ and $p \triangleq \rho/2$. So now we need to lower bound $Z(x+p) - Z(x-p)$. Therefore,

$$\begin{aligned} &Z(x+p) - Z(x-p) \\ &\quad = (3(x+p) - 3d(x+p)^2 + d^2(x+p)^3) - (3(x-p) - 3d(x-p)^2 + d^2(x-p)^3) \\ &\quad = 3((x+p)-(x-p)) - 3d((x+p)^2 - (x-p)^2) + d^2((x+p)^3 - (x-p)^3) \\ &\quad = 3(2p) - 3d(4xp) + d^2(6x^2p + 2p^3) \\ &\quad = 6p - 12dxp + 6d^2x^2p + 2d^2p^3 \\ &\quad = 3\rho - 6dx\rho + 3d^2x^2\rho + \frac{d^2}{4}\rho^3 \\ &\quad = 3\rho(1-dx)^2 + \frac{d^2}{4}\rho^3 \\ &\quad \geq \frac{\rho^3}{4} \end{aligned}$$

□

**Fact 5.6.** *Let* $Z(\beta) \triangleq 3\beta(1-\beta)^2 + (1-\frac{1}{|F|-1})3\beta^2(1-\beta) + (1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\beta^3$. *For any* $\beta$ *and* $0 \leq \rho \leq 1$ *such that* $0 \leq \beta + \frac{\rho}{2} \leq \frac{2}{1+\frac{1}{|F|-1}}$, $Z(\beta+\rho) \leq Z(\beta) + 4\rho$.

*Proof.* We can use the same notation and the same first several steps of the Fact 5.5 to get

$$\begin{aligned} Z(x+p) - Z(x-p) &= 3\rho(1-dx)^2 + \frac{d^2}{4}\rho^3 \\ &\leq 3\rho + \rho^3 \\ &\leq 4\rho \end{aligned}$$

□

**Claim 5.7.** *For large enough* $m$,

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} > \binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$$

*Proof.* When $1 \leq k < q$:

$$\frac{\binom{q}{k}\binom{m-q}{\delta m-k}}{\binom{m}{\delta m}} = \binom{q}{k}\frac{(\delta m)!(m-\delta m)!}{m!}\frac{(m-q)!}{(\delta m-k)!(m-\delta m-q+k)!}$$
$$= \binom{q}{k}\frac{\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)}{m(m-1)...(m-q+1)}$$

Let us start by considering just the numerator:

$$\delta m(\delta m-1)...(\delta m-k+1)(m-\delta m)(m-\delta m-1)...(m-\delta m-q+k+1)$$

Expand it out, and collect like powers of $m$. This gives:

$$\delta^k(1-\delta)^{q-k}m^q - ((\sum_{i=0}^{k-1} i)\delta^{k-1}(1-\delta)^{q-k} + (\sum_{i=0}^{q-k-1} i)\delta^k(1-\delta)^{q-k-1})m^{q-1} + O(m^{q-2})$$

Note that in the sum above, the $m^{q-j}$ terms have positive coefficients when $j$ is even and negative coefficients when $j$ is odd. Therefore, for large enough $m$, we can bound the numerator by just two terms:

$$> \delta^k(1-\delta)^{q-k}m^q - ((\sum_{i=0}^{k-1} i)\delta^{k-1}(1-\delta)^{q-k} + (\sum_{i=0}^{q-k-1} i)\delta^k(1-\delta)^{q-k-1})m^{q-1}$$
$$= \delta^k(1-\delta)^{q-k}m^q - (\frac{(k-1)k}{2}\delta^{k-1}(1-\delta)^{q-k} + \frac{(q-k-1)(q-k)}{2}\delta^k(1-\delta)^{q-k-1})m^{q-1}$$
replacing the summations
$$> \delta^k(1-\delta)^{q-k}m^q - \frac{q^2}{2}(\delta+1-\delta)\delta^{k-1}(1-\delta)^{q-k-1}m^{q-1}$$
because $k(k-1)$ and $(q-k-1)(q-k)$ are both less than $q^2$
$$> \delta^k(1-\delta)^{q-k}m^q - q^2m^{q-1}$$

The denominator, $m(m-1)...(m-q+1)$, is upper bounded by $m^q$, so the overall expression is bounded by

$$> \binom{q}{k}\frac{\delta^k(1-\delta)^{q-k}m^q - q^2m^{q-1}}{m^q}$$
$$= \binom{q}{k}\delta^k(1-\delta)^{q-k} - \frac{q^2}{m}$$

When $k = q$:

$$\begin{aligned}\frac{\binom{m-q}{\delta m-q}}{\binom{m}{\delta m}} &= \frac{\delta m(\delta m-1)...(\delta m-q+1)}{m(m-1)...(m-q+1)} \\ &= \frac{\delta^q m^q - \frac{(q-1)q}{2}\delta^{q-1}m^{q-1} + O(m^{q-2})}{m(m-1)...(m-q+1)} \\ &> \frac{\delta^q m^q - q^2 m^{q-1}}{m(m-1)...(m-q+1)} \\ &> \frac{\delta^q m^q - q^2 m^{q-1}}{m^q} \\ &= \delta^q - \frac{q^2}{m}\end{aligned}$$

When $k = 0$, the bound is straightforward by a similar calculation as in the case $k = q$. □

**Claim 5.8.** *Let $Q \subseteq [m]$ with $|Q| = 3$. Let $k \in F$ and $\beta \leq 1 - 1/|F|$. Let $c_j \in F^*$ for $j \in Q$. Let $B$ be chosen from a distribution that first uniformly chooses a set of $\beta m$ coordinates from $[m]$, and then assigns a uniformly random value from $F^*$ independently to each chosen coordinate. The remaining $m - \beta m$ coordinates are set to 0. Then*

$$\begin{aligned}(|F|-1)\min_{k\in F}\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\right] \geq \\ 3\beta(1-\beta)^2 + (1-\frac{1}{|F|-1})3\beta^2(1-\beta) + (1-\frac{1}{|F|-1}+\frac{1}{(|F|-1)^2})\beta^3 - 3\frac{9}{m}\,.\end{aligned}$$

*Proof.* We use the following decomposition:

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\right] = \sum_{i=0}^{3}\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\;\middle|\;(|B\cap Q| = i)\right]\Pr_B\left[|B\cap Q| = i\right]\,,$$

where $(|B\cap Q| = i)$ denotes the event that $B$ has $i$ nonzero coordinates among the 3 positions in $Q$.

We use Claim 5.7 to get an estimate on $\Pr_B[|B\cap Q| = i]$, that does not depend on $m$, except in the term $9/m$. It remains to estimate the probabilities

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\;\middle|\;(|B\cap Q| = i)\right]\,. \tag{6}$$

First note that this probability is 0 when $k \neq 0$ and $i = 0$ or when $k = 0$ and $i = 1$; it is 1 when $k = 0$ and $i = 0$, and it is $\frac{1}{|F|-1}$ when $k \neq 0$ and $i = 1$ or when $k = 0$ and $i = 2$. We use that a position in $B \cap Q$ always contributes a nonzero value, since the adversary assigns values from $F^*$ to the chosen positions, and each coefficient $c_j$ is nonzero. Thus, to estimate

(6) when $k = 0$ and $i = 3$, note that the linear combination involving all 3 positions can only be 0 if the part over any 2 positions is nonzero. This gives $\frac{1}{|F|-1}(1 - \frac{1}{|F|-1})$. For the remaining cases, note that when $k \neq 0$,

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)|(|B\cap Q| = i)\right] = \frac{1}{|F|-1}\left(1 - \Pr_B\left[(\sum_{j\in Q} c_j B_j = k)|(|B\cap Q| = i-1)\right]\right) .$$

Thus, for every nonzero $k$ we have:

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\right] \geq$$
$$\frac{1}{|F|-1}\left(3\beta(1-\beta)^2 - \frac{9}{m}\right) + \frac{1}{|F|-1}\left(1 - \frac{1}{|F|-1}\right)\left(3\beta^2(1-\beta) - \frac{9}{m}\right) +$$
$$\frac{1}{|F|-1}\left(1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2}\right)\left(\beta^3 - \frac{9}{m}\right) .$$

The estimate follows by Claim 5.9, which shows that for $\beta \leq 1 - 1/|F|$, and $k \in F^*$

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = 0)\right] \geq \Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\right] .$$

□

**Claim 5.9.** *Let $Q \subseteq [m]$ with $|Q| = q$. Let $\beta \leq 1 - 1/|F|$. Let $c_j \in F^*$ for $j \in Q$. Let $B$ be chosen from a distribution that first uniformly chooses a set of $\beta m$ coordinates from $[m]$, and then assigns a uniformly random value from $F^*$ independently to each chosen coordinate. The remaining $m - \beta m$ coordinates are set to 0. Then for any $k \in F^*$,*

$$\Pr_B\left[(\sum_{j\in Q} c_j B_j = 0)\right] \geq \Pr_B\left[(\sum_{j\in Q} c_j B_j = k)\right] .$$

*Proof.* We prove the statement by induction on $|Q|$. For $|Q| = 1$, the statement is equivalent to the claim that $1 - \beta \geq \beta/(|F| - 1)$. This holds since we assumed $\beta \leq 1 - 1/|F|$.

Assume the statement holds for $|Q| = q - 1$. Let $Q$ be a query set with $|Q| = q$, and let $j_1$ denote the first position in $Q$. For $\ell \in F$, denote by $E_\ell$ the event that $(\sum_{j\in Q} c_j B_j = \ell)$. Let $p_1 = \Pr_B[j_1 \in B \cap Q]$ and $p_0 = \Pr_B[j_1 \notin B \cap Q]$. Then,

$$\Pr_B[E_\ell] = \Pr_B[E_\ell | j_1 \in B \cap Q]p_1 + \Pr_B[E_\ell | j_1 \notin B \cap Q]p_0$$

.

Notice that

$$\Pr_B[E_0 | j_1 \in B \cap Q] = \sum_{k\neq 0} \frac{1}{|F|-1} \Pr_B[\sum_{j\in Q\setminus\{j_1\}} c_j B_j = k)] ,$$

$$\Pr_B[E_0|j_1 \notin B \cap Q] = \Pr_B[\sum_{j \in Q \setminus \{j_1\}} c_j B_j = 0)] \ ,$$

and for $k \in F^*$

$$\Pr_B[E_k|j_1 \in B \cap Q] = \sum_{\ell \neq k} \frac{1}{|F|-1} \Pr_B[\sum_{j \in Q \setminus \{j_1\}} c_j B_j = \ell)] \ ,$$

$$\Pr_B[E_k|j_1 \notin B \cap Q] = \Pr_B[\sum_{j \in Q \setminus \{j_1\}} c_j B_j = k)] \ .$$

Notice that for the distribution over $B$ that we consider, $p_1 = 1 - \beta$ and $p_0 = \beta$. Using the induction hypothesis and that $\beta \leq 1 - 1/|F|$, this implies the statement. □

**Claim 5.10.** *Let $Q \subseteq [m]$ with $|Q| = q$. Let $k \in F$ and $\beta \leq 1 - 1/|F|$. Let $c_j \in F^*$ for $j \in Q$. Let $B$ be chosen from a distribution that first uniformly chooses a set of $\beta m$ coordinates from $[m]$, and then assigns a uniformly random value from $F^*$ independently to each chosen coordinate. The remaining $m - \beta m$ coordinates are set to 0. Then*

$$(|F|-1) \min_{k \in F} \Pr_B \left[ (\sum_{j \in Q} c_j B_j = k) \right] = \Pr_B \left[ (\sum_{j \in Q} c_j B_j \neq 0) \right]$$

*Proof.* First notice that since $\beta \leq 1 - 1/|F|$, for $k \in F^*$ by Claim 5.9

$$\Pr_B \left[ (\sum_{j \in Q} c_j B_j = 0) \right] \geq \Pr_B \left[ (\sum_{j \in Q} c_j B_j = k) \right] \ .$$

Now the claim follows since for nonzero $k$, the probabilities $\Pr_B \left[ (\sum_{j \in Q} c_j B_j = k) \right]$ are the same. See the proof of Claim 5.8 for their value when $q = 3$. □

**Claim 5.11.** *Let $Q \subseteq [m]$ with $|Q| = 3$ be an arbitrary fixed subset of the coordinates. Let $c_j \in F^*$ for $j \in Q$, and let $\beta \leq 1 - 1/|F|$. Let $P(\beta, 3, F) \triangleq \Pr_B \left[ (\sum_{j \in Q} c_j B_j = 0) \right]$, where the probability is over $B \in F^m$ randomly chosen according to a distribution that first chooses to corrupt each coordinate in $[m]$ independently with probability $\beta$, and then uniformly and independently assigns a value from $F^*$ to each chosen coordinate of $B$. The remaining coordinates of $B$ are set to 0. Then*

$$P(\beta, 3, F) = 1 - 3\beta(1-\beta)^2 - (1 - \frac{1}{|F|-1}) 3\beta^2(1-\beta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3 \ .$$

*Proof.* As in Claim 5.8 we use the decomposition:

$$\Pr_B \left[ (\sum_{j \in Q} c_j B_j = k) \right] = \sum_{i=0}^{3} \Pr_B \left[ (\sum_{j \in Q} c_j B_j = k) | (|B \cap Q| = i) \right] \Pr_B [|B \cap Q| = i] \ ,$$

where $(|B \cap Q| = i)$ denotes the event that $B$ has $i$ nonzero coordinates among the 3 positions in $Q$. For the current distribution, we have $\Pr_B[|B \cap Q| = i] = \binom{3}{i}\beta^i(1-\beta)^{3-i}$.

We estimate the probabilities $\Pr_B\left[(\sum_{j\in Q} c_j B_j = k)|(|B \cap Q| = i)\right]$ as in the proof of Claim 5.8. The bound follows using Claim 5.10. □

**Claim 5.12.** *Let* $0 < \epsilon \leq 1 - 1/|F|$ *and* $\alpha \triangleq \delta - (1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}) - \psi(n) - \nu$. *Let* $Z(\beta) = 3\beta(1-\beta)^2 + (1 - \frac{1}{|F|-1})3\beta^2(1-\beta) + (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\beta^3$ . *Let* $n$ *be large enough, so that* $\psi(n)^3/4 < \epsilon$. *Then the expression* $Z(\beta) - \frac{\psi(n)^3}{4}$ *evaluated at* $\beta = \min(\frac{\delta-\alpha-\nu}{\gamma}, 1 - 1/|F|)$ *is* $\geq 1 - 1/|F| - \epsilon$.

*Proof.* If $\min(\frac{\delta-\alpha-\nu}{\gamma}, 1 - 1/|F|)$ is equal to $1 - 1/|F|$, then we have $Z(\beta) = Z(1 - 1/|F|) = 1 - 1/|F|$. For large enough $n$, $\psi(n)^3/4 < \epsilon$ and thus $Z(\beta) - \psi(n)^3/4 > 1 - 1/|F| - \epsilon$.

Let $\xi = 1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}$ and note that $Z(\xi) = 1 - 1/|F| - \epsilon$. Since $Z$ is monotone, $Z(\beta) \geq Z(\delta - \alpha - \nu)$. We have $\delta - \alpha - \nu = \xi + \psi(n)$. Thus, the statement follows by Claim 5.5. □

**Claim 5.13.** *Let* $0 < \epsilon \leq 1 - 1/|F|$ *and* $\alpha \triangleq \delta - (1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}) - \frac{\phi(n)}{4}$. *Then* $\alpha > 0$ *when* $\frac{1}{|F|} + \epsilon > 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$. *Moreover,* $\alpha > \frac{\mu}{4}$ *when* $\frac{1}{|F|} + \epsilon > \mu + 1 - 3\delta(1-\delta)^2 - (1 - \frac{1}{|F|-1})3\delta^2(1-\delta) - (1 - \frac{1}{|F|-1} + \frac{1}{(|F|-1)^2})\delta^3 + \phi(n)$ *for some* $\mu \geq 0$.

*Proof.* The proof is analogous to the proof of Claim 5.1, using Claim 5.6, and noting that $Z(\xi) = 1 - 1/|F| - \epsilon$ for $\xi = 1 - \frac{1}{|F|} - \epsilon^{1/3}(1 - \frac{1}{|F|})^{2/3}$. □