Electronic Colloquium on Computational Complexity, Report No. 133 (2009)

Near-optimal extractors against quantum storage

Anindya De[*] Thomas Vidick[†]

December 9, 2009

Abstract

We give near-optimal constructions of extractors secure against quantum bounded-storage adversaries. One instantiation gives the first such extractor to achieve an output length $\Theta(K-b)$, where K is the source's entropy and b the adversary's storage, depending linearly on the adversary's amount of storage, together with a poly-logarithmic seed length. Another instantiation achieves a logarithmic key length, with a slightly smaller output length $\Theta((K-b)/K^{\gamma})$ for any $\gamma > 0$. In contrast, the previous best construction [Ts09] could only extract $(K/b)^{1/15}$ bits.

Our construction follows Trevisan's general reconstruction paradigm [Tre01], and in fact our proof of security shows that essentially all extractors constructed using this paradigm are secure against quantum storage, with optimal parameters. Our argument is based on bounds for a generalization of quantum random access codes, which we call *quantum functional access codes*. This is crucial as it lets us avoid the local list-decoding algorithm central to the approach in [Ts09], which was the source of the multiplicative overhead.

Some of our constructions have the additional advantage that every bit of the output is a function of only a polylogarithmic number of bits from the source, which is crucial for some cryptographic applications.

Keywords: Extractors, Quantum storage, Random Access Codes, List decodable code

[*] Computer Science Division, University of California, Berkeley, CA, USA. anindya@cs.berkeley.edu. Supported by the "Berkeley Fellowship for Graduate Study" and Luca Trevisan's US-Israel BSF grant 2006060.

[†] Computer Science Division, University of California, Berkeley, CA, USA. vidick@cs.berkeley.edu.

ISSN 1433-8092

1 Introduction

Randomness extractors are fundamental building blocks in pseudorandomness theory, with many applications to derandomization, error-correcting codes, and expanders, among others. They are also of central importance in cryptography, where they are often used to build key generation primitives. In this context, one usually has the notion of an adversary, a malicious observer who is trying to discover a bit of the honest player's output. The most prominent model for adversaries is the bounded storage model, introduced by Maurer [Mau92], in which the adversary is allowed to store a limited amount of information about the extractor's input.

Formally, we say that a function $Ext : \{0,1\}^N \times \{0,1\}^t \rightarrow \{0,1\}^m$ is a (k,ε) strong extractor if for every distribution X with min-entropy at least k (X is called the *source*) and uniformly random Y (called the *seed*), the distribution $Ext(X,Y)$ is ε-close to uniform in statistical distance. The extractor is said to be secure against b bits of storage if $Ext(X,Y)$ is ε-close to uniform even to an adversary who has been allowed to store b bits of information about X, and has also later been revealed the seed Y.

Constructions of extractors are known that are almost-optimal in all parameters, even in the presence of the adversary (in fact, a result by Lu [Lu04] shows that any strong (k,ε) extractor is essentially a $(k+b,\varepsilon)$ extractor secure against b bits of storage). Nevertheless, in a world in which no adversary can be trusted, König *et al.* [KMR05] introduced the following interesting twist: what if the adversary is allowed *quantum* memory? In this setting, the fundamental difficulty that arises is a familiar one, with a long history: how much information can be encoded in a quantum state?

The fact that this question can admit very different answers depending on its precise formulation is reflected in the fact that not all classical extractor constructions are secure in the presence of a quantum adversary, as was shown in [GKK$^+$07]. Nevertheless, many constructions have been shown to be sound on a case-by-case basis [KMR05, KT08, FS08, Ts09]. All these constructions, however, have parameters that are far from optimal either in terms of seed length or of output length.

In order to explore this question, we first describe a construct which we call quantum functional access codes (QFAC). QFACs will be central in the proofs of correctness of our extractor constructions.

Quantum functional access codes. Holevo [Hol73] was the first to tackle the question of the information capacity of a quantum state, showing that one needs at least n qubits in order to encode n bits of information. However, this bound only holds when it is required that the whole n bits be recoverable from the quantum storage. As such, it is generally not applicable in a cryptographic context, where typically even partial information is important. Instead of asking for the whole input $x \in \{0,1\}^n$ to be recoverable from its encoding $\Psi(x)$, Ambainis *et al.* [ANTsV02] consider encodings in which it is only required that any bit of x can be recovered from $\Psi(x)$ with probability $1/2+\varepsilon$ (over the measurement's randomness), and they call such encodings 'random access codes' (RACs). Note that, since the encoding is quantum, the recovarability of any one bit does not imply the recovarability of the whole string x, so that Holevo's bound does not apply. Nevertheless, Ambainis *et al.* showed that RACs require essentially $(1 - H(1/2+\varepsilon))n$ qubits to encode n bits, providing a linear lower bound for fixed ε. These bounds have proved instrumental in many results in information theory. In fact, as pointed out in [Ts09], random access codes provide a way to construct one-bit extractors that are secure against quantum storage.

We push this question even further: what if, instead of asking that the encoding lets us recover any bit of the input, we asked that it lets us recover some fixed set of functions of the input? For example, we could ask about encodings that let us recover the XOR of any k bits of the input[1], but one can also consider more general settings.

One might ask about the relevance of such encodings, when we already know that there are strong linear lower bounds on RACs – surely, these will extend to any encoding which lets us recover more than any single bit of the input. The key point here is that, even though both Holevo's bound and the RAC lower bounds are linear in the input length when the success probability p is fixed, the two bounds scale very differently when one considers the dependence on p: while Holevo's lower bound scales as $n - \log 1/p$, the RAC bound scales as $(4\varepsilon^2/\ln 2)n$ for small $\varepsilon = 2p - 1$. So we are asking, how does the length of the code scale with the set of functions that we are trying to recover?

Define a (n, b, ε) QFAC for a set of n-bit strings A and a set of functions $\mathcal{C}$ from A to $\{0,1\}$ as a b-qubit encoding of strings $x \in A$ such that, for any function $f \in \mathcal{C}$, one can recover $f(x)$ from its encoding with success probability $1/2 + \varepsilon$.[2] Intuitively, the more the set of functions $\mathcal{C}$ is error resilient (i.e., the more spread-out the images $(f(x))_{f\in\mathcal{C}} \in \{0,1\}^{|\mathcal{C}|}$), the stronger the lower bound should be on the length of the encoding. For example, using a simple reduction to known results we can show that any (n, b, ε) QFAC for the set $\mathcal{C} = \{f_y : x \mapsto x \cdot y \mod 2,\ y \in \{0,1\}^n\}$ must have length $n - \log 1/\varepsilon$. If one simply used the fact that such a QFAC can be used to recover any bit of x with probability $1/2 + \varepsilon$, the resulting bound would be the much weaker $O(\varepsilon^2 n)$.

We believe that QFACs constitute a primitive that should be of wide interest in studying the properties of quantum states from an information-theoretic point of view. In this paper, we demonstrate the relevance of this construct by showing how good bounds on some QFACs can be used to prove the security of an extractor against quantum storage with almost-optimal parameters. In fact, many previous constructions of extractors against quantum storage can be seen as implicitly proving bounds on QFACs. For example, the construction in [KMR05] shows that any (n, b, ε) QFAC for a set of 2-universal hashing functions must have length $b \geq n - \log 1/2\varepsilon$.

Techniques. In this section we give an overview of our proof technique, explaining the connection between extractors and QFACs in the context of Trevisan's general construction paradigm [Tre01]. To describe this, let us first give a brief conceptual overview of the main steps that go into the proof of the construction by Ta-Shma [Ts09].

The construction proceeds by encoding the weakly random source $x \sim X$ using a locally list-decodable code C [STV01]. This is followed by an application of the Nisan-Wigderson generator [NW94], interpreting $C(x)$ as the truth table of the "hard" function.

The proof of correctness for this construction, as the first part of ours, follows the general reconstruction framework of [Tre01]. For the sake of contradiction, assume that there is a statistical test T which uses the adversary's quantum information $\Psi(x)$ to distinguish the output from uniform with advantage ε. A Markov argument shows that for at least an $\varepsilon/2$ fraction of the samplings x from the source (call them bad samplings), T can distinguish the output (when the source is x) from uniform by at least $\varepsilon/2$. Consider any such bad sampling x. A standard hybrid argument, along with properties of the Nisan-Wigderson generator, allows us to construct a circuit T' (using little non-uniformity about x) which predicts a random position of $C(x)$ with probability $\frac{1}{2} + \delta$ where $\delta = \frac{\varepsilon}{m}$. Further, T' makes exactly one query to T.

At this stage, we have constructed a small circuit T', which uses the adversary's quantum information in order to predict the bits of $C(x)$ with some small success probability. The proof in [Ts09] follows

[1] Such codes were introduced in [BARdW08], where they are called XOR-QRACs

[2] A RAC is then simply a QFAC for the set of index functions $f_i : x \mapsto x_i$.

that in [Tre01] by showing how, from such a circuit, one can give another circuit which predicts any position of x with probability 0.99 and queries T' at most $q = (1/\delta)^c$ times ($c = 15$ for the code in [Ts09]). This gives a random access code for x; however since it makes q measurements on the quantum state $\Psi(x)$, the no-cloning theorem forces us to see it as having a length of $q \cdot b$ qubits. The main drawback of this method, when transferred to the quantum setting, is thus that the quantum state needs to be copied a large number of times in order to get a RAC – thus yielding a weaker bound than one might hope for.

Our proof departs fundamentally from the usual reconstruction paradigm at this point: instead of using a short RAC for $C(x)$ to construct a longer RAC for x, we give a direct analytical argument showing that a RAC for $C(x)$ must have large length. Note that a RAC for x is simply a QFAC for the class of functions $f_i : x \mapsto C(x)_i$. Intuitively, such a QFAC cannot be short, even though its success probability $1/2 + \varepsilon/m$ is small. If the QFAC is classical (*C*FAC), this is easy to show: assume that there existed a short CFAC for this problem. One can just repeat the recovery procedure to get a string y that agrees with $C(x)$ at a fraction $1/2 + \varepsilon/m$ of positions, and then one can use the good list-decoding properties of C to argue that the CFAC essentially lets us recover the whole input x, and hence must be long. In the quantum setting, however, it is far from obvious if this is true. The primary problem is that we cannot repeat the recovery procedure y, because quantum states are fragile.

In order to overcome this difficulty, we directly prove an analytical lower bound on the QFAC derived from the code. This lets us derive a contradiction, proving that our extractor is safe against quantum storage. The idea for the lower bound consists in seeing any good QFAC as an adversary which uses small memory, and is able to predict codeword positions. By using the fact that good list-decodable codes give rise to classical one-bit strong extractors, we can transform the adversary into a quantum adversary for a specific one-bit extractor. Finally, a result by Koenig and Terhal [KT08] shows that such an adversary would imply a classical adversary with similar storage, which we know does not exist.

Our results. We show that any extractor based on Trevisan's reconstruction paradigm [Tre01] is also safe against a quantum bounded-storage adversary, with near-optimal parameters. Rather than give the full technical result here (see Theorem 4.6), we discuss instantiations with two specific codes.

We first use a code from [GHSZ02], which is obtained through the concatenation of the Reed-Solomon code and the Hadamard code. This lets us prove the following:

Theorem 1.1 *For any $\gamma > 0$ and $K = \Omega(1/poly(N))$, $\varepsilon = 1/\text{poly}(N)$, there is a polynomial-time computable function $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ which is a $(K, 2\varepsilon)$ extractor against b qubits of quantum storage, where $t = O(\log N)$ and $m = \Omega\left(\frac{K-b}{K^\gamma}\right)$.*

We note that the construction in [Ts09] uses the concatenation of a Reed-Muller code with the Hadamard code, the parameters of the Reed-Muller code being chosen so that one can do local list-decoding. In contrast, our analysis just needs a good list-decoding radius, but no local list-decoding property. Hence our result carries over to [Ts09] and in particular implies that the construction in [Ts09] has much better output length than the one shown in that paper, which was $\Omega((K/b)^{1/15})$.

This first construction does not have the desirable property of *local computability.* By using a different code, we can also show the following:

Theorem 1.2 *For any $\alpha > 0$ and $\varepsilon = 1/\text{poly}(N)$, there is a function $Ex : \{0,1\}^N \times \{0,1\}^t \to$*

$\{0,1\}^m$ which is a $(\alpha N, 2\varepsilon)$ extractor against b qubits of quantum storage, where $t = O(\log^4 N)$ and $m = \Omega(\alpha N - b)$. Moreover, each bit output by the extractor is computable in $\operatorname{poly}\log N$ *time.*

Even though it has a slightly larger key length (note however that its output length is the optimal $\Theta(\alpha N - b)$), a major advantage of this extractor is its simplicity: each bit of the output is simply the XOR of $O(\log N)$ bits of the source, chosen based on the seed. In particular, it is locally computable[3]. On the other hand, that construction is restricted to extracting from linear entropy rates. This is inevitable, as lower bounds by Viola [Vio04] show that locally computable extractors cannot extract from sources with entropy less than $N^{0.99}$ using a polylogarithmic seed length.

The QFACs at the heart of this second construction are in fact the XOR-QRACs from [BARdW08]. A by-product of our proof is an improvement of the lower bound proved in that paper on the length of such codes (see Corollary 3.10).

A nice side feature of both these constructions, especially if one is interested in cryptographic applications, is that it is possible to achieve an arbitrary inverse polynomial statistical distance from the uniform distribution, while paying only a polylogarithmic cost in terms of output length and seed length (this will be apparent from the more detailed statement of Theorem 1.1 given in Section 4). Only the first property was known to hold for previous short seed extractor constructions against quantum storage.

Applications to cryptography. Our results are of direct applicability to the following key expansion scenario. Alice and Bob share a small secret uniformly random key K. They would like to expand it into a longer key K' in order to communicate in presence of an adversary Eve. A public source of weak randomness R (assume that R has min-entropy at least k) is available to all parties. When the string R is broadcast, Eve is allowed to compute an arbitrary function $\Psi : \{0,1\}^{|R|} \to \{0,1\}^b$ and store the result. However, once she stores $\Psi(R)$, her access to R is cut off. Indeed, Eve is assumed to have a bounded storage capacity and so she can only store a limited amount of information about R. The goal is to come up with an efficient function *Ext* which can be used by Alice and Bob to compute the shared string $K' = Ext(R, K)$. The required security condition is that K' is close to being uniformly random to Eve, even given her knowledge of $\Psi(R)$. In fact, we would like K' to remain random even if K is later revealed to Eve (after $\Psi(R)$ is computed and access to R has been cut off).

For this application, it is important that *Ext* be *locally computable*, i.e. individual bits of the output should be a function of a polylogarithmic number of bits of the source R. Indeed, since we are putting a cap on the adversary's storage it would be unreasonable not to put a similar cap on the memory used by the honest parties Alice and Bob to compute bits of their shared key.

Our second construction has the property of being locally computable: every bit of the output is a function of polylogarithmically many bits from the source. While various constructions of classical locally computable extractors are already known [DM04, Lu04, Vad04, DT09], ours are the first to be proved secure against quantum adversaries. This makes them particularly suitable for use in the context of bounded storage cryptography.

Organization of the paper. We start with some preliminaries in Section 2. In Section 3 we introduce quantum functional access codes and give bounds for some specific families of these codes.

[3] For this to hold, we also need to check that the bits to be XOR-ed can be chosen in poly-logarithmic time, which is the case in this construction.

In Section 4 we describe our construction and state its parameters. Finally, the proof of security is given in Section 5.

2 Notation and Preliminaries

The following notations are used throughout the paper. For $x \in \{0,1\}^n$, x_i denotes the i^{th} bit of x. A tuple $(y_1, y_2, \ldots, y_k)$ is denoted by $\otimes_{i=1}^{k} y_i$, or sometimes simply $\overline{y}$. The concatenation of two strings x and y is denoted by $x \circ y$. If x and y are tuples, then $x \circ y$ represents the bigger tuple formed by concatenating x and y. For $z_1, \ldots, z_k \in \{0,1\}$, $\oplus_{i=1}^{k} z_i$ denotes the XOR of $z_1, \ldots, z_k$. We use $\Delta(x,y)$ for normalized Hamming distance between x and y. $\mathcal{D}_b$ denotes the set of all density matrices on b qubits, while $\mathcal{O}_b$ is the set of all ± 1-valued observables on b qubits. All logarithms are taken in base 2. Throughout, H will denote the binary entropy function $H(x) = -x \log x - (1-x)\log(1-x)$ for $0 < x < 1$.

Distributions. The uniform distribution on $\{0,1\}^n$ is denoted by U_n. We will manipulate random variables that have both classical and quantum parts. In general, given two random variables X, Y, $X \circ Y$ is the same as the random variable (X, Y). Given two states ρ, σ, $\rho \circ \sigma$ is just $\rho \otimes \sigma$. Finally, given a random variable $X : \Omega \to \{0,1\}^n$ and a state ρ, $X \circ \rho$ denotes the state $\mathbb{E}_{w \in \Omega}\left[|X(w)\rangle\langle X(w)| \otimes \rho\right]$. The statistical distance between two distributions D_1 and D_2 (or, more generally, the trace distance when these distributions involve quantum components) is denoted by $\|D_1 - D_2\|$.

Definition 2.1 *A (classical) distribution X is said to have min-entropy at least k (denoted $H_\infty(X) \geq k$) if $\forall x$, $Pr[X = x] \leq 2^{-k}$.*

Extractors. We first give the the formal definition of a strong extractor.

Definition 2.2 *$Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is said to be a (k, ε) strong extractor if for every distribution X with min-entropy at least k, we have that $\|U_{m+t} - Ext(X, U_t) \circ U_t\| \leq \varepsilon$. Here both U_t's in the second expression correspond to the same sampling.*

X is usually called the source *(and N its length), while the extractor's second input is called the* seed *(of length t).*

We now extend this definition to that of a strong extractor secure against a bounded-storage quantum adversary.

Definition 2.3 *$Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is said to be a (k, ε) strong extractor against b qubits of quantum storage if for every map $\Psi : \{0,1\}^N \to \mathcal{D}_b$ which maps N bits to a quantum state over b qubits and every distribution X such that $H_\infty(X) \geq k$*

$$\|U_m \circ \Psi(X) \circ U_t - Ext(X, U_t) \circ \Psi(X) \circ U_t\| \leq \varepsilon \tag{1}$$

where both U_t's in the second expression correspond to the same sampling.

We note that condition (1) above is equivalent to requiring that for any statistical test $T : \{0,1\}^m \times \mathcal{D}_b \times \{0,1\}^t \to \{0,1\}$,

$$\left| \underset{x \sim X, y \sim U_t, u \sim U_m}{\mathbb{P}} [T(u, \Psi(x), y) = 1] - \underset{x \sim X, y \sim U_t}{\mathbb{P}} [T(Ext(x,y), \Psi(X), y) = 1] \right| \leq \varepsilon$$

Since we can always assume that any measurements performed by T are done at the end, this is equivalent to saying that for any $M : \{0,1\}^{m+t} \to \mathcal{O}_b$,

$$\Big|\mathbb{E}_{x\sim X, y\sim U_t}\left[\mathrm{Tr}\left(\mathbb{E}_{u\in\{0,1\}^m}\left[M(u,y)\right]\Psi(x)\right) - \mathrm{Tr}(M(Ext(f,y),y)\Psi(x))\right]\Big| \leq 2\varepsilon$$

Quantum codes. A (n, b) quantum encoding is a map $\Psi : \{0,1\}^n \to \mathcal{D}_b$ mapping x to a density matrix $\Psi(x)$. A fundamental theorem due to Holevo essentially states that $\Psi(x)$ cannot contain more information about x than a classical string of b bits:

Theorem 2.4 *[Hol73] Let X be any distribution on $\{0,1\}^n$ and $\Psi(X) = \mathbb{E}_{x\in X}\Psi(x)$. For a particular measurement M, let Y_M denote the random variable resulting from applying the measurement on $\Psi(X)$. If $I(X : Y)$ denotes the mutual information of X and Y and $S(\Psi(X))$ denotes the von Neumann entropy of $\Psi(X)$, then $I(X : Y) \leq S(\Psi(X))$.*

Oracle circuits. Our proofs of security will involve the construction of oracle circuits. If A is an oracle circuit, we denote by A^B the circuit that uses B as the oracle. Further, let C be an oracle machine which uses A as an oracle (denoted by C^A). Then it is understood that when C calls A, then A calls the appropriate oracle B. Thus $A^C \equiv A^{C^B}$. We will also use the following easy claim:

Claim 2.5 *Let B be any oracle such that oracle circuit A can be constructed using at most t_1 bits of advice and A queries B at most q_1 times. Again let C be an oracle circuit which queries A and C can be constructed using at most t_2 bits of advice. Further, C queries A at most q_2 times. Then C can be considered as an oracle circuit which queries B at most q_1q_2 times and can be constructed using at most $t_1 + t_2$ bits of advice.*

3 Quantum Functional Access Codes

Consider the following problem from the theory of classical error-correcting codes. Let $C : \{0,1\}^n \to \{0,1\}^m$ be a code which is (ε, L) list-decodable i.e. for any $x \in \{0,1\}^m$, there are at most L codewords y such that the relative Hamming distance between x and y is less than $(\frac{1}{2} - \varepsilon)m$. Let $A = \{C(x) :\ x \in \{0,1\}^n\}$ be the set of all codewords, and consider $Enc : A \to \{0,1\}^b$, a probabilistic encoding such that for every $z \in A$, z_i can be recovered from $Enc(z)$ with probability $\frac{1}{2} + 2\varepsilon$, on average over the choice of $i \in [m]$. Given $Enc(z)$, by performing the recovery procedure for every index i, we obtain a string y which will agrees with z on at least a $\frac{1}{2} + \varepsilon$ fraction of the positions with high probability. But then the exact element z can be recovered using just an additional $\log |L|$ bits of advice (as per the list-decodability property of C). Hence, Enc can be seen as a high-probability encoding of any codeword, using only $b + \log |L|$ bits. However, the obvious information-theoretic bounds shows that this must be at least $\log |C|$ bits, implying that $b \geq \log \frac{|C|}{|L|}$. This is much better than the usual random access code bound $b \geq O(\varepsilon^2 \log |C|/ \log n)$ for small ε, that one gets if there is no guarantee on the structure of the set C.

To model this situation more precisely, note that the recovery procedure lets us recover any bit of $C(x)$ with non-trivial probability. As such, Enc can be seen as a probabilistic encoding of every $x \in \{0,1\}^n$ which lets us evaluate a class of functions $\mathcal{C} = \{g_i : x \mapsto C(x)_i,\ i \in [n]\}$. This is a generalization of the usual random access codes, introduced in [ANTsV99], for which $\mathcal{C} = \mathcal{C}_1 = \{g_i : x \mapsto x_i,\ i \in [n]\}$.

It is natural to expect that lower bounds for this more demanding kind of random access code would be tighter than more general lower bounds, in a way that depends on the structure of $\mathcal{C}$. We introduce the following definition:

Definition 3.1 *Let $A \subset \{0,1\}^n$, and $\mathcal{C} \subset \{f : A \to \{0,1\}\}$ be a set of functions defined on A. For $\varepsilon \in (0, 1/2]$, a (n, b, ε) quantum functional access code, or QFAC, for $(A, \mathcal{C})$ is a map $\Psi : A \to \mathcal{D}_b$ such that, for every $f \in \mathcal{C}$, there is an observable M_f such that for every $x \in A$, $(-1)^{f(x)}\mathrm{Tr}(M_f\Psi(x)) \geq 2\varepsilon$.*

The discussion above shows that a *classical* functional access code for a set of functions $\mathcal{C} = \{g_i : x \mapsto C(x)_i,\ i \in [n]\}$ that is derived from a good list-decodable code C will have a strong lower bound on its length. However, the classical argument cannot be extended in a straightforward manner to the quantum case, as it is dependent upon performing successive measurements on the classical encoding. If the encoding is quantum, the first such measurement will destroy the state, and we will not be able to proceed further.

Nevertheless, we are able to prove bounds in some specific cases. We start with the standard setting of random access codes, for which Theorem 4.1 in [ANTsV99] implies the following (see also Theorem 3.2 in [Ts09]):

Lemma 3.2 *Let $A \subset \{0,1\}^n, 0 < \varepsilon \leq \frac{1}{2}$ and $\Psi : \{0,1\}^n \to \mathcal{D}_b$ be a (n, b, ε) quantum functional access code for $(A, \mathcal{C}_1)$. Then $\log|A| \leq O\left(\frac{b \log n}{\varepsilon^2}\right)$.*[4]

Central to this work is the fact that functional access codes for larger classes of functions than the simple coordinate functions $\mathcal{C}_1$ enjoy much stronger lower bounds, with a weaker dependence on the success probability ε. König, Maurer and Renner [KMR05] show the following:

Theorem 3.3 ([KMR05], Theorem 12 and Corollary 13) *Let $\mathcal{C}$ be the set of all functions from $\{0,1\}^n$ to $\{0,1\}$. Then any (n, b, ε) QFAC for $(A, \mathcal{C})$ satisfies $\log|A| \leq b + 2\log 1/2\varepsilon$. Moreover, the same bound holds if $\mathcal{C}$ is any family of two-universal hash functions, and the decoding procedure is only required to be correct on average over the choice of both x and f.*

There is an obvious connection between lower bounds on the length of QFACs and lower bounds on one-way quantum communication complexity, even though results in the latter setting usually do not focus on the error dependence as much as is needed for our applications. Nevertheless, the following bound easily follows from known results in communication complexity:

Lemma 3.4 *Let $\mathcal{C} = \{g_y : x \mapsto x \cdot y \mod 2,\ y \in \{0,1\}^n\}$. If there exists a (n, b, ε) QFAC for $(A, \mathcal{C})$, then $\log|A| \leq b + 2\log(1/2\varepsilon)$.*

Proof: Note that any (n, b, ε) QFAC for $(A, \mathcal{C})$ implies a one-way quantum protocol for the communication problem in which Alice is given $x \in A$, Bob is given $y \in \{0,1\}^n$, and their goal is to output $x \cdot y \mod 2$. Using a reduction from [CDNT98], any such protocol communicating b qubits and succeeding with probability $1/2 + \varepsilon$ can be transformed into a protocol that sends any $x \in A$ to Bob, using b qubits, with success probability $4\varepsilon^2$. Theorem 1.1 in [NS06] then shows that $b \geq \log|A| - \log(1/4\varepsilon^2)$. ■

[4] As noted in [Ts09], the loss of a factor $\log n$ is inevitable. Note however that this can be removed in the case where $A = \{0,1\}^n$ by following the proof for quantum random access codes in [ANTsV02].

Families of two-universal hash functions over $\{0,1\}^n$, as well as the Hadamard code, both have size $\Omega(2^n)$, which makes them unsuitable for our purposes. Indeed, in our applications to extractors we will use the seed to select a few random functions from a family $\mathcal{C}$ and apply them to the source in order to obtain the output. However, using any of the last two function families would require a seed of size n, where n is the length of the source, whereas we would like the seed to be poly-logarithmic in the source length[5].

In order to get good extractor constructions, it is crucial to prove strong lower bounds on QFACs for smaller classes of efficiently computable functions. Our main result is based on the fact, proved below, that there are no short QFACs for families of functions that are defined from list-decodable codes. This extends the discussion introducing this section to the case of quantum encodings, and in fact we will get essentially the same bound as stated there —even though, as we argued was necessary, the proof will be very different. It will be useful to consider *approximately* list-decodable codes, which we define as follows:

Definition 3.5 *A code $C : \{0,1\}^N \to \{0,1\}^{\overline{N}}$ is (ε,δ,L) approximately list-decodable if for every $x \in \{0,1\}^{\overline{N}}$, there exists at most L strings $\{y_i\}_{i=1}^L \in \{0,1\}^N$, such that for any string $z \in \{0,1\}^N$ satisfying $\Delta(x, C(z)) < 1/2 - \varepsilon$, $\exists i \in [L]$ such that $\Delta(z, y_i) \leq \delta$. If C is $(\varepsilon, 0, L)$ approximately-list decodable then we simply say that C is (ε, L) list-decodable.*

Proposition 3.6 *Let $N, \overline{N}, L, b \in \mathbb{N}$, and $\varepsilon, \delta > 0$. Let $C : \{0,1\}^N \to \{0,1\}^{\overline{N}}$ be a (ε,δ,L) approximately list-decodable code, and $\mathcal{C} = \{f_i : x \mapsto C(x)_i,\ i \in [\overline{N}]\}$. Let $A \subseteq \{0,1\}^N$, and Ψ be a (N,b,ε) QFAC for $(A,\mathcal{C})$. Then*

$$\log|A| < H(\delta)N + b + \log L + O(\log 1/\varepsilon)$$

Moreover, this bound holds even when we only require the QFAC to have success probability $1/2+\varepsilon$ on average over the choice of $g \in \mathcal{C}$, instead of for all g.

The proof crucially relies on the result by König and Terhal [KT08] that strong one-bit extractors are automatically safe against quantum adversaries, in some range of parameters. It proceeds through the following three steps:

1. Show that the any (ε,δ,L) approximately list-decodable code C defines a good 1-bit classical strong extractor.
2. Use Theorem III.1 from [KT08] to show that the previous extractor is automatically safe against quantum adversaries that are allowed some amount of storage.
3. Conclude by showing how the security against quantum storage implies a lower bound on any QFAC for $\mathcal{C}$.

We proceed with the details.

Proof: Let $t = \log \overline{N}$ (assume it an integer for notational simplicity) and consider the following 1-bit extractor

$$\begin{array}{rccc} E: & \{0,1\}^N \times \{0,1\}^t & \to & \{0,1\} \\ & (x,y) & \mapsto & C(x)_y \end{array}$$

The following claim proves item 1 above.

[5] Note that usually we use N to denote the length of the source.

Claim 3.7 *$E : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}$ as defined above is a (K, ε) strong extractor for any $K > H(\delta)N + \log L + \log \frac{2}{\varepsilon}$.*

Proof: Assume for the sake of contradiction that E is not a (K, ε) strong extractor. Then there is a distribution D with min-entropy K, and a statistical test T such that the following holds.

$$| \mathop{\mathbb{P}}_{y \sim U_t, x \sim D}[T(y) = C(x)_y] - \frac{1}{2}| \geq \varepsilon$$

With a possible flip in the output of circuit T, we get a new test T' such that

$$\mathop{\mathbb{P}}_{y \sim U_t, x \sim D}[T'(y) = C(x)_y] \geq \frac{1}{2} + \varepsilon$$

By a Markov argument, there is a set $BAD \subseteq \{0,1\}^N$ such that for every $x \in BAD$,

$$\mathop{\mathbb{P}}_{y \sim U_t}[T'(y) = C(x)_y] \geq \frac{1}{2} + \frac{\varepsilon}{2}$$

and $\mathbb{P}_{x \sim D}[x \in BAD] \geq \varepsilon/2$. Evaluating T' on every possible $y \in \{0,1\}^t$ results in a string x' such that

$$\mathop{\mathbb{P}}_{y \in \{0,1\}^t}[x'_y = C(x)_y] \geq \frac{1}{2} + \frac{\varepsilon}{2} \tag{2}$$

We can now use the (ε, δ, L) list-decodability properties of C. For any x' satisfying (2) we can get a set of $k \leq L$ strings $x^1, \ldots, x^k$ such that at least one of them satisfies that

$$\mathop{\mathbb{P}}_{y \sim \mathcal{U}_N}[x^i_y = x_y] \geq 1 - \delta \tag{3}$$

Note that process of finding $x^1, \ldots, x^k$ need not be polynomial time, but we only require existence here; the important point is that the list of x^i is uniquely determined by x' (take the lexicographically smallest list satisfying the conditions in the fact). If $x^1, \ldots, x^k$ are known, then we require at most $\log L$ bits to specify $i \in [t]$ such that x^i satisfies (3). Once x^i is specified, we know that x must be among one of the at most $2^{H(\delta)N}$ possible N-bit strings which are δ-close to x. Hence we require an additional $H(\delta)N$ bits to fully specify x. Thus, the total amount of bits used to specify x is $\log L + H(\delta)N$, which in turn implies that the size of the set BAD is bounded by $L \cdot 2^{H(\delta)N}$.

To conclude the argument, observe that every element in BAD is sampled with probability at most 2^{-K} and hence $\mathbb{P}_{X \in D}[X \in BAD] \leq (L \cdot 2^{K + H(\delta)N})$. However, this is a contradiction if

$$L \cdot 2^{-K+H(\delta)N} < \frac{\varepsilon}{2} \qquad \text{i.e.} \qquad K > H(\delta)N + \log L + \log \frac{2}{\varepsilon}$$

which gives the bound stated in the claim. ■

Let $\eta > 0$ be an error parameter, $A \subseteq \{0,1\}^N$, and $\mathcal{U}_A$ the uniform distribution on A. Theorem III.1 in [KT08] implies that, as long as

$$\log |A| - b \geq K + \log 1/\eta, \tag{4}$$

the function E is automatically a $(\log |A|, 3\sqrt{\eta})$ extractor that is secure against b qubits of quantum storage (see Definition 2.3). This means that, for any collection of quantum states $\Psi(x) \in \mathcal{D}_b$, knowledge of y and $\Psi(x)$ cannot help distinguish $E(x, y)$ from a uniformly random bit with advantage more than $3\sqrt{\eta}$ (over the choice of x in A, and uniform y). In particular, we have that for any collection of ± 1 observables $(M_y)_{y \in \{0,1\}^t}$ on $\mathcal{D}_b$,

$$\mathbb{E}_{x \in A,\, y \in \{0,1\}^t} \left[(-1)^{C(x)_y} \mathrm{Tr}(M_y \Psi(x)) \right] \leq 3\sqrt{\eta}$$

By definition, any (N, b, ε) QFAC for $(A, \mathcal{C})$, even one that is only correct on average over the choice of y, contradicts this conclusion for $\eta = 4\varepsilon^2/9$. Hence our assumption (4) on the size of A must be contradicted, i.e. any such QFAC must be such that $\log |A| < K + b + \log 9/4\varepsilon^2$. Setting K to be the smallest possible value satisfying the condition in Claim 3.7, we get

$$\log |A| < H(\delta)N + b + \log L + O(\log 1/\varepsilon)$$

■

We will use two instantiations of this proposition, for specific families of codes. The first one, which will let us get an extractor with optimal seed length, is based on the following from [GHSZ02]:

Fact 3.8 *For any $N \in \mathbb{N}$, $\varepsilon > 0$, there exists a polynomial-time computable binary code $C_R : \{0,1\}^N \to \{0,1\}^{\overline{N}}$, where $\overline{N} = O(N/\varepsilon^4)$, that is $(\varepsilon, O(1/\varepsilon^2))$ list-decodable.*

These codes lead to the following, the proof of which follows immediately from Proposition 3.6:

Corollary 3.9 *Let C_R be the code from Fact 3.8, and $\mathcal{C}_R = \{f_i : x \mapsto C(x)_i,\ i \in [\overline{N}]\}$. Then any (N, b, ε) QFAC for $(A, \mathcal{C}_R)$ is such that*

$$\log |A| < b + O(\log 1/\varepsilon)$$

Moreover, this bound holds even when we only require the QFAC to have success probability $1/2 + \varepsilon$ on average over the choice of $g \in \mathcal{C}_R$, instead of for all g.

Our second main construction uses a QFAC for the class $\mathcal{C}_k = \{g : x \mapsto \bigoplus_{j=1}^k x_{i_j},\ (i_1, \ldots, i_k) \in [n]\}$. QFACs for this class of functions were introduced in [BARdW08], where they are called XOR-QRACs. That paper shows a bound on the length of such codes using a generalization of the hypercontractive inequality to matrix-valued functions. We improve their result by showing the following:

Corollary 3.10 *Let k, N be integers, and $\varepsilon > 2k^2/2^N$. Let $A \subset \{0,1\}^N$. If there exists a (N, b, ε) QFAC for $(A, \mathcal{C}_k)$, then*

$$\log |A| < b + H\left(\frac{1}{k} \ln \frac{2}{\varepsilon}\right) N + O\left(\log \frac{1}{\varepsilon}\right)$$

Moreover, this bound holds even when we only require the QFAC to have success probability $1/2 + \varepsilon$ on average over the choice of $g \in \mathcal{C}_k$, instead of for all g.

By generalizing the proof of Theorem 7 in [BARdW08] (which is only stated for $A = \{0,1\}^N$ in that paper), we can get the bound $\log |A| \leq b + \left(1 - \frac{1}{2 \ln 2}\right) N + O\left(\log \frac{1}{\varepsilon}\right)$. This would lead to an extractor construction which only works for sources with min-entropy γN for $\gamma > 0.28$, and our improvement on their bound gets rid of this constraint.

Proof: The following fact (for a reference, see [IJK06], Lemma 42) shows that the XOR code is $(\varepsilon, (1/k)\ln(2/\varepsilon), 4/\varepsilon^2)$ approximately list-decodable for any $\varepsilon > 2k^2/2^N$.

Fact 3.11 *For every $\varepsilon > 2k^2/2^N$ and $z' \in (\{0,1\}^N)^k$, there is a list of $t \le 4/\varepsilon^2$ elements $x^1, \ldots, x^t \in \{0,1\}^N$ such that the following holds: for every $z \in \{0,1\}^N$ which satisfies*

$$\mathop{\mathbb{P}}_{\{y_1,\ldots,y_k\}\in\binom{N}{k}} [z'_{(y_1,\ldots,y_k)} = \oplus_{i=1}^k z_{y_i}] \ge \frac{1}{2} + \varepsilon$$

there is an $i \in [t]$ such that

$$\mathop{\mathbb{P}}_{y\sim\mathcal{U}_N} [x^i_y = z_y] \ge 1 - \delta$$

with $\delta = (1/k)\ln(2/\varepsilon)$.

Note that in [IJK06] the lemma is proved for tuples instead of sets, and has a $t \le 1/\varepsilon^2$. However, since most tuples are sets, it is straightforward to get the above version for sets. Plugging the list-decoding parameters from this Fact in the bound of Proposition 3.6 immediately gives the result. ■

4 Overview of the construction

Our construction is inspired by the construction by Trevisan [Tre01] and its subsequent adaptation against quantum storage by Ta-shma [Ts09]. However, our proof technique differs from that of [Ts09] in that it avoids constructing random access codes by copying the adversary's storage many times. Rather, we use the much stronger bounds on QFACs proved in Section 3. This is crucial in allowing us to prove an additive, rather than multiplicative, dependence of the output on the adversary's storage.

We first describe a few standard tools that are used in the construction, before giving it in detail. Its correctness will be proved in Section 5.

4.1 Preliminaries

Definition 4.1 *A collection of subsets $S_1, \ldots, S_m \subset [l]$ is called a (l, n, m, ρ) weak design if for all i, $|S_i| = n$ and for all j, $\sum_{i<j} 2^{|S_i \cap S_j|} \le \rho(m-1)$.*

The following theorem is due to Raz, Reingold and Vadhan [RRV99].

Theorem 4.2 *For every $m, l \in \mathbb{N}$ and $\rho \ge 2$, there is a (l, n, m, ρ) design which is computable in time $(ml)^{O(1)}$ with $l = O(\frac{n^2}{\log \rho})$.*

Note that the value of l blows up when ρ approaches 1. In order to keep l bounded even as ρ approaches 1, we can use a construction given in [RRV99]. Even though the construction is computable in polynomial time, it does not meet many finer notions of efficiency which are of interest to us. Hartman and Raz [HR03] achieved similar parameters with a better efficiency:

Theorem 4.3 *For every $m, l \in \mathbb{N}$ such that $m > n^{\log n}$ and $0 < \gamma < \frac{1}{2}$, there is a $(l, n, m, 1+\gamma)$ design such that $l = O(n^2 \log \frac{1}{\gamma})$. Further, each individual design can be output in time polynomial in l and n.*

For the purposes of this paper, let $l(n, \rho)$ denote the smallest value of l for which Theorems 4.2 or 4.3 guarantee the existence of a weak (l, n, m, ρ) design. Whether we use Theorem 4.2 or 4.3 depends on how small we want ρ to be.

Our last tool is the Nisan-Wigderson generator with respect to a function $f : \{0,1\}^n \to \{0,1\}$.

Definition 4.4 *Let $S_1, \ldots, S_m$ be a (l, n, m, ρ) weak-design guaranteed by Theorem 4.2 or 4.3. Let $x \in \{0,1\}^{2^n}$. Then $NW^x : \{0,1\}^l \to \{0,1\}^m$ is defined as*

$$NW^x(y) = x_{y_{S_1}} \circ \ldots \circ x_{y_{S_m}}$$

Here y_{S_j} denotes the restriction of y to the indices in S_j.

4.2 Description of the construction

Let $C : \{0,1\}^N \to \{0,1\}^{\overline{N}}$ be a code with good (possibly approximate) list-decoding capabilities, t the extractor's seed length, m its ouptut length, and $\rho > 0$ a parameter of the construction. Let $(S_1, \ldots, S_m)$ be a $(t, \log \overline{N}, m, \rho)$ design as discussed above. Define a *key-expansion* function g as the function implicit in the Nisan-Wigderson generator: g maps $y \in \{0,1\}^t$ to the m-tuple $(y_{S_1}, \ldots, y_{S_m}) \in ([\overline{N}])^m$. Then the extractor can be succintly described as

$$\begin{aligned} Ext_C : \{0,1\}^N \times \{0,1\}^t &\to \{0,1\}^m \\ (x, y) &\mapsto NW^{C(x)}(y) \end{aligned}$$

Remark 4.5 The local computability properties of the extractor will depend on those of the code C. Even if every bit of $C(x)$ can be computed in time $\mathrm{poly}(\log N)$, one additionally requires that any individual design be computable in time $\mathrm{poly}(\log m, \log N)$. While this is true for the design in Theorem 4.3 when ρ is very close to 1, it is not true for the design from Theorem 4.2, which we will nevertheless use in order to get a smaller key length. However, the cost of computing the designs can be considered as a one-time cost, as they can be pre-computed if desired. Further, the entire design can be computed in time in polynomial in m. So, irrespective of whether or not individual bits can be computed in polylogarithmic time, the extractor remains polynomial-time computable.

4.3 Main theorem

Our main result is the following:

Theorem 4.6 *Let $\delta > 0$, $N, \overline{N}, K, b \in \mathbb{N}$, $1 < \rho < N^{0.02}$, and $\varepsilon = 1/poly(N)$. Define*

$$m = \frac{K - b - H(\delta)N - \Omega(\log(1/\varepsilon) + \log^2 \overline{N})}{1 + \rho}$$

If $C : \{0,1\}^N \to \{0,1\}^{\overline{N}}$ is a $(\varepsilon/m, \delta, L)$ approximately list-decodable code such that $L = poly(m/\varepsilon)$, then the function $Ext_C : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a $(K, 2\varepsilon)$ extractor secure against b qubits of quantum storage, where $t = l(\log \overline{N}, \rho)$.

We give two instantiations of this result. The first one uses the codes from Fact 3.8, and lets us achieve optimal seed length. We obtain it by setting $\rho = K^\gamma$, for any $\gamma > 0$, and using the combinatorial designs guaranteed by Theorem 4.2:

Corollary 4.7 *Let $0 < \gamma < 1$ be any constant, $N \in \mathbb{N}$, and $\varepsilon = 1/\mathrm{poly}(N)$. Let C_R be the code from Fact 3.8. The function $Ext_{C_R} : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a $(K, 2\varepsilon)$ extractor against b qubits of quantum storage for any $K = \Omega(1/poly(N))$, where $t = O(\log N)$ and $m = \Omega\left(\frac{K-b}{K^\gamma}\right)$.*

An inconvenient of this construction, particularly relevant in cryptography, is that, even though the extractor is polynomial-time computable, it is not *locally computable*. Indeed, any bit of the output may require polynomial time to be computed, whereas one might wish for it to be computable in polylogarithmic time. We achieve such an extractor by taking $C = C_k : \{0,1\}^N \to \{0,1\}^{\binom{N}{k}}$ the XOR code $C_k(x)_{y_1,\ldots,y_k} = x_{y_1} \oplus \ldots \oplus x_{y_k}$. By using these codes together with the designs from Theorem 4.3, the bound from Corollary 3.10 gives the following:

Corollary 4.8 *Let $\alpha, \delta > 0$ be any constants, $N, b \in \mathbb{N}$, and $\varepsilon = 1/\mathrm{poly}(N)$. For all large enough N, the function $Ext_{C_k} : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a $(\alpha N, 2\varepsilon)$ extractor against b qubits of quantum storage where $t = O(\log^4 N)$, $m = \frac{1}{2}((\alpha - 2\delta)N - b)$, and we have set $k = O(\log(m/\varepsilon)/\delta^2)$.*

Note that this extractor is locally computable, and every individual bit of the output can be computed in polylogarithmic time, as the designs in Theorem 4.3 are locally computable. Note also that the extractor only works for linear entropy rates: this is inevitable, as lower bounds by Viola [Vio04] show that locally computable extractors cannot extract from sources with entropy less than $N^{0.99}$ using a polylogarithmic seed length.

Remark 4.9 The seed length of this construction can be improved to $t = O(\log^3 N)$ by using designs from Theorem 4.2. In this case, however, the output length is slightly reduced to $m = \Omega\left(\frac{(\gamma-\delta)N-b-\log(1/\varepsilon)}{N^\gamma}\right)$ for any $\gamma > 0$. In that case, even though the extractor is locally computable, individual bits are not computable in $\mathrm{poly}\log N$ time because the computation of even one element of the design in Theorem 4.2 requires $\mathrm{poly}(m)$ time.

5 Proof of security

We give the proof of security of our construction. The first steps of the proof follow the general reconstruction paradigm from [Tre01], and we give them first.

5.1 Proofs in the reconstruction paradigm

We start with the following standard observation.

Observation 5.1 *In order to prove that $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a $(K, 2\varepsilon)$ strong extractor against b qubits of storage, it suffices to prove that for any function $M : \{0,1\}^{m+t} \to \mathcal{O}_b$ and $\Psi : \{0,1\}^N \to \mathcal{D}_b$, there are at most $\varepsilon 2^K$ strings $x \in \{0,1\}^N$ such that*

$$\left|\mathbb{E}_{y\in\{0,1\}^t}\left[\mathrm{Tr}(\mathbb{E}_{u\in\{0,1\}^m}\left[M(y,u)\right]\Psi(x)) - \mathrm{Tr}(M(y, Ext(x,y))\Psi(x))\right]\right| > 2\varepsilon \qquad (5)$$

where, for every x, $\Psi(x)$ is a state on b qubits representing the adversary's memory after having seen the input x.

Proof: Assume for contradiction that $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is not a $(K, 2\varepsilon)$ strong extractor against b qubits of quantum storage. By definition, there exists $M : \{0,1\}^{m+t} \to \mathcal{O}_b$ such that

$$\left|\mathbb{E}_{x\sim X, y\sim U_t}\left[\mathrm{Tr}\left(\mathbb{E}_{u\in\{0,1\}^m}\left[M_{u,y}\right]\Psi(x)\right) - \mathrm{Tr}(M_{Ext(x,y),y}\Psi(x))\right]\right| > 4\varepsilon$$

where X is the source's distribution. Since it has min-entropy at least K, it must be true that for at least $\varepsilon 2^K$ inputs x,

$$\left|\mathbb{E}_{y\sim U_t}\left[\mathrm{Tr}(\mathbb{E}_{u\in\{0,1\}^m}\left[M_{y,u}\right]\Psi(x)) - \mathrm{Tr}(M_{Ext(x,y),y}\Psi(x))\right]\right| > 2\varepsilon$$

■

Fix $M : \{0,1\}^{t+m} \to \mathcal{O}_b$. The previous observation shows that, in order to show that Ext is a strong extractor, it suffices to bound the number of strings x such that (5) holds. For this, we use the reconstruction approach in [Tre01]. For a fixed x, define $M_x : \{0,1\}^{m+t} \to \{0,1\}$ as the probabilistic procedure which, on input $(y,u) \in \{0,1\}^{t+m}$, outputs 1 with probability $1/2 + \mathrm{Tr}(M(y,u)\Psi(x))/2$, where $\Psi(x)$ is the state of the adversary's storage on x. M_x implicitly depends on the adversary's storage $\Psi(x)$, but for the most part our proofs will be oblivious to this fact, and will simply treat M_x as a probabilistic oracle. Moreover, all probabilities that we write involving M_x, or other oracle circuits making calls to M_x, will implicitly be taken over M_x's internal randomness. We will however carefully keep track of the number of calls made to M_x, as the quality of the final bound will depend crucially on it.

Once such a M_x has been fixed, the first step is to use the standard hybrid argument followed by Yao's distinguisher versus predictor lemma to get an oracle circuit T which queries M_x exactly once, and is such that T predicts $Ext(x,y)_i$ with some advantage over a random guess when y as well as the value of x on some related points are given as input. We skip the (by now, standard) argument and state the final result (see [Tre01] for details).

Lemma 5.2 *Let M, x, ε be such that (5) is satisfied, and $Ext(x,y)_i$ be the i^{th} bit of the extractor's output on (x,y). Then using $m + \log m + 3$ bits of classical advice, we can get a circuit T which makes one query to M_x and is such that for some $1 \le i \le m$, T satisfies:*

$$Pr_{y\in U_t}[T^{M_x}(y, \otimes_{j=1}^{i-1} Ext(x,y)_j) = Ext(x,y)_i] > \frac{1}{2} + \frac{\varepsilon}{m} \qquad (6)$$

Our next step is to construct a small circuit R_x which predicts the value of $C(x)$ at any position y with some non-negligible correlation, leading to the following technical lemma:

Lemma 5.3 *Let $M : \{0,1\}^{m+t} \to \mathcal{O}_b$, $x \in \{0,1\}^N$, and $\varepsilon > 0$ such that (5) holds. Then using classical advice s of length $m(1+\rho) + \log m + t + O(1)$, we can construct an oracle circuit R_x, which makes one call to M_x and predicts $C(x)_z$ with probability $1/2 + \varepsilon/m$, on average over the choice of $z \in \{0,1\}^{\overline{N}}$.*

Proof: By Lemma 5.2, for any M, x satisfying (5) for $\varepsilon > 0$, using $m + \log m + 3$ bits of advice, we can get an oracle circuit T which makes exactly one query to M_x and for some $1 \le i \le m$ satisfies

$$\mathbb{P}_y[T^{M_x}(y, C(x)_{y_{S_1}}, \ldots, C(x)_{y_{S_{i-1}}}) = C(x)_{y_{S_i}}] \ge \frac{1}{2} + \frac{\varepsilon}{m}$$

Let us split y into two parts $z = y_{S_i}$ and $w = y_{[l]-S_i}$. Let y_{S_j} now be denoted by $h_j(z,w)$. The above probability can then be rewritten as

$$\underset{zow}{\mathbb{P}}[T^{M_x}(z,w,C(x)_{h_1(z,w)},\ldots,C(x)_{h_{i-1}(z,w)}) = C(x)_z] \geq \frac{1}{2} + \frac{\varepsilon}{m}$$

By an averaging argument, we can fix a w (using at most t bits of advice) such that the above inequality holds with the probability taken over z. Let us hardwire all the possible values of $C(x)_{h_j(z,w)}$ (for the fixed value of w), as z varies over $\{0,1\}^N$ and j varies between 1 and $i-1$, into the circuit T. By the definition of a weak design, there are at most $(m-1)\rho$ bits that need to be hardwired. Let R_x be the circuit with all the hardwired values. R_x satisfies the following equation

$$\underset{z}{\mathbb{P}}[R_x^{M_x}(z) = C(x)_z] \geq \frac{1}{2} + \frac{\varepsilon}{m} \tag{7}$$

The total classical advice taken so far is $m + \log m + t + m\rho + O(1)$. ■

5.2 Security against quantum storage from lower bounds on QFACs

Assume for contradiction that there is an adversary to *Ext*, which can distinguish its output from uniform given access to the seed y and some partial quantum information $\Psi(x) \in \mathcal{D}_b$ about the source. Such an adversary can be described by the mapping Ψ, together with a function $M : \{0,1\}^{t+m} \to \mathcal{O}_b$ describing the adversary's measurement on his quantum information $\Psi(x)$, when provided with the seed and the extractor's output[6].

For a fixed x, let $M_x : \{0,1\}^{m+t} \to \{0,1\}$ as in Section 5.1. By Observation 5.1, to prove that *Ext* is a $(K, 2\varepsilon)$ strong extractor secure against b qubits of quantum storage, it suffices to prove that there are at most $\varepsilon 2^K$ strings x such that

$$\left|\mathbb{E}_{y\in\{0,1\}^t,\, u\in\{0,1\}^m}\left[M_x(y,u) - M_x(y, Ext(x,y))\right]\right| > \varepsilon \tag{5}$$

At this point, the key conceptual step in our proof is to observe that from the circuit R_x given by Lemma 5.3, we can construct a QFAC for the family $\mathcal{C} = \{f_i : x \mapsto C(x)_i,\ i \in [\overline{N}]\}$ of codeword positions, and the set A of all x satisfying (5). The strong lower bounds we proved in Section 3 then let us bound the set A, as a function of the adversary's storage and the list-decoding properties of C. The following claim makes this connection formal.

Claim 5.4 *Let $A \subseteq \{0,1\}^N$ be such that, for any $x \in A$, using only c bits of classical advice and b qubits of quantum advice, we can construct a circuit R_x which is such that for a random y, it predicts $C(x)_y$ with probability $1/2 + \varepsilon$. Then the cardinality of A is at most $s \cdot 2^c$, where s is the maximum size of a set B such that there exists a (N, b, ε) QFAC for $(B, \mathcal{C})$.*

Proof: The c advice bits partition the set A into 2^c sets A_s, for $s \in \{0,1\}^c$. Fix such a s and consider the set A_s. Since s has been fixed, all $x \in A_s$ have the same circuit R_x; only the b-qubit quantum state $\Psi(x)$ on which it operates depends on x. Hence there is a fixed set of measurements such that, for a random y, the measurement M_y on $\Psi(x)$ outputs $C(x)_y$ with probability $1/2 + \varepsilon$. This means we have a (N, b, p) QFAC for $(A_s, \mathcal{C})$. Hence the size of A_s is bounded by the maximum size of any set for which such a code exists. This gives us the promised bound on A. ■

[6] This describes the most general situation, as we can always assume that any measurement made by the adversary is done at the end of his recovery procedure.

To finish the proof of Theorem 4.6, note that by Proposition 3.6 and our assumption that C is $(\varepsilon/m, \delta, poly(m/\varepsilon))$ list-decodable, any $(N, b, \varepsilon/m)$ QFAC for $(A, \mathcal{C})$ satisfies

$$\log|A| \leq b + H(\delta)\,N + O(\log m/\varepsilon)$$

Applying Claim 5.4 to the advice circuit promised by Lemma 5.3, we deduce that the number of strings x such that (5) holds is at most $2^{b+H(\delta)N+O(\log(m/\varepsilon))} \cdot 2^{m(1+\rho)+\log m+t+O(1)}$. and using $t = O(\log^2 \overline{N})$ and $\log(m) = O(\log N)$, this expression can be upper-bounded by $2^{b+H(\delta)N+m(1+\rho)+O(\log(1/\varepsilon)+\log^2 \overline{N})}$. Using the bound on m given in Theorem 4.6, we immediately get that this expression is upper-bounded by $\varepsilon 2^K$, finishing the proof of the Theorem.

Acknowledgements. We are grateful to Falk Unger and Umesh Vazirani for helpful comments concerning the presentation of this manuscript.

References

[ANTsV99] Andris Ambainis, Ashwin Nayak, Amnon Ta-shma, and Umesh V. Vazirani. Dense quantum coding and quantum finite automata. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 376–383, 1999.

[ANTsV02] Andris Ambainis, Ashwin Nayak, Amnon Ta-shma, and Umesh V. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, 2002.

[BARdW08] Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A Hypercontractive Inequality for Matrix-Valued Functions with Applications to Quantum Computing and LDCs . In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pages 477–486, 2008. Full version at arXiv:0705.3806.

[CDNT98] Richard Cleve, Wim van Dam, Michael Nielsen, and Alain Tapp. Quantum entanglement and the communication complexity of the inner product function. In *QCQC '98: Selected papers from the First NASA International Conference on Quantum Computing and Quantum Communications*, pages 61–74, London, UK, 1998. Springer-Verlag.

[DM04] Stefan Dziembowski and Ueli Maurer. Optimal randomizer efficiency in the bounded-storage model. *Journal of Cryptology*, 17(1):5–26, 2004. Preliminary version in *Proc. of STOC'02*.

[DT09] Anindya De and Luca Trevisan. Extractors using hardness amplification. In *APPROX-RANDOM*, pages 462–475, 2009. Full version available at *http://www.cs.berkeley.edu/∼anindya/exthardfull.pdf*.

[FS08] Serge Fehr and Christian Schaffner. Randomness extraction via *delta* -biased masking in the presence of a quantum attacker. In Ran Canetti, editor, *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 465–481. Springer, 2008.

[GHSZ02] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1034, 2002.

[GKK$^+$07] Dmitri Gavinsky, Julia Kempe, Iordanis Kerendis, Ran Raz, and Ronald de Wolf. Exponential separations for one-way quantum communication complexity with applications to cryptography. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 516–525, 2007.

[Hol73] Alexander Holevo. Information-theoretic aspects of quantum measurement. *Problems of Information Transmission*, 9(2):31–42, 1973.

[HR03] Tzvika Hartman and Ran Raz. On the distribution of the number of roots of polynomials and explicit weak designs. *Random Structures and Algorithms*, 23(3):235–263, 2003.

[IJK06] Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Approximately List-Decoding Direct Product Codes and Uniform Hardness Amplification. In *Proceedings of the 47th IEEE Symposium on Foundations of Computer Science*, pages 187–196, 2006. Full version at `http://www1.cs.columbia.edu/∼rjaiswal/`.

[KMR05] Robert König, Ueli Maurer, and Renato Renner. On the power of quantum memory. *IEEE Transactions on Information Theory*, 51(7):2391–2401, 2005.

[KT08] Robert König and Barbara Terhal. The bounded storage model in presence of a quantum adversary. *IEEE Transactions on Information Theory*, 54(2):749–762, 2008.

[Lu04] Chi-Jen Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *Journal of Cryptology*, 17(1):27–42, 2004.

[Mau92] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *J. Cryptology*, 5(1):53–66, 1992.

[NS06] Ashwin Nayak and Julia Salzman. Limits on the ability of quantum states to convey classical messages. *J. ACM*, 53(1):184–206, 2006.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994. Preliminary version in *Proc. of FOCS'88*.

[RRV99] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 149–158, 1999.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. Preliminary version in STOC-Complexity 99.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.

[Ts09] Amnon Ta-shma. Short seed extractors against quantum storage. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 401–409, 2009.

[Vad04] Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, 2004.

[Vio04] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2004.