SIAM J. COMPUT.
Vol. 40, No. 3, pp. 664–677


# SHORT SEED EXTRACTORS AGAINST QUANTUM STORAGE*

AMNON TA-SHMA†

**Abstract.** In this paper we show that a construction of Trevisan, solving the privacy amplification problem in the classical setting, also solves the problem when the adversary may keep quantum storage, thereby giving the first such construction with logarithmic seed length. The technique we use is a combination of Trevisan's approach of constructing an extractor from a black-box pseudorandom generator, together with locally list-decodable codes and previous work done on quantum random access codes.

**Key words.** extractors, quantum storage, random access codes, list-decodable code

**AMS subject classification.** 68Q12

**DOI.** 10.1137/09076787X

## 1. Introduction.

In the *privacy amplification* problem, Alice and Bob share information that is only partially secret with respect to an eavesdropper Charlie. A typical example is the following. Alice holds a string $x$ chosen uniformly from $\{0,1\}^n$ and sends it to Bob. Charlie listens on the line and keeps a state $\rho_x$ depending on $x$, where the only limitation imposed on Charlie is that he may keep at most $b$ bits (or qubits in the quantum setting) of memory. Alice and Bob's goal is to distill this information to a shorter string that is completely secret even toward Charlie.

The problem was introduced in the classical setting (i.e., considering a classical adversary Charlie) in [5, 4]. In the classical setting the problem can be solved almost optimally with the help of a shared, public random string. Formally, Alice and Bob use a probabilistic hash function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ that takes as input the partially secret input $x \in \{0,1\}^n$ and a public random string $y \in \{0,1\}^t$ (called the *seed*) and hashes the input (that is partially known to Charlie) to a shorter string $E(x,y)$ that looks almost uniform to Charlie. Such a function $E$ is called an *extractor*. The goal is to explicitly construct extractors with a large output length and a short seed length, and many good constructions are known today.

A major open problem that was first formally stated in König, Maurer, and Renner's work [19, 20] is whether the privacy amplification problem can be solved in the presence of a *quantum* adversary Charlie. This quantum variant of the problem naturally occurs in analyzing the security of some quantum key distribution protocols and in quantum bounded-storage cryptography. See, e.g., [6, 21, 22].

In the quantum setting we can distinguish between two types of information the quantum adversary may hold about the partially secret string $x$. First, it may hold some classical data about $x$, and this can be captured as saying that from the adversary's point of view the input is drawn from some distribution $X$ on $\{0,1\}^n$. The optimal amount of entropy we may hope to extract is upper-bounded by the min-entropy of $X$. Second, the adversary may also hold some quantum information about the source by keeping a $b$-qubit state $\rho(x)$ correlated with $x$. We say $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a $(k,b,\epsilon)$ *extractor against quantum storage* if for any

*Received by the editors August 11, 2009; accepted for publication (in revised form) March 18, 2011; published electronically June 2, 2011. This work was supported by EU grant QCS and by Israel Science Foundation grant 1090/10. A preliminary version of the paper appeared in STOC 2009.
http://www.siam.org/journals/sicomp/40-3/76787.html
†Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel (amnon@tau.ac.il).

distribution $X$ on $\{0,1\}^n$ with min-entropy $k$, and any $b$ qubit state $\rho(x)$ the adversary holds, the output distribution $E(X, U_t)$ is close to uniform. See Definition 2.2 for the precise details.

One could hope for a *generic* result showing that every extractor against classical storage is also good against quantum storage. Indeed, König and Terhal [22] showed that any extractor with a *single-bit output* that works well in the classical setting is also good against quantum storage with slightly worse parameters. They also showed that any extractor that has error $\epsilon$ in the classical setting has at most $2^{O(b)}\epsilon$ error against adversaries with $b$ quantum bits of storage. On the negative side, Gavinsky et al. [13] showed an example of an extractor that works well against classical storage but fails against quantum storage. Thus, no generic result is possible, or at the very least, any such generic result must involve penalties in some parameters.

There are several works showing that *specific* extractor constructions that work well in the classical setting also work well against quantum storage. The first such construction was given by König, Maurer, and Renner [19, 20], who showed that the pairwise independent extractor of [17] is also good (and with the same parameters) against quantum storage. Using the same techniques the result can be extended to using almost pairwise independence [29]. Fehr and Schaffner [12] show that another classical extractor for very high min-entropies is good against quantum storage (the classical version appears, e.g., in [9]).

Yet, in spite of much effort, all the methods above require seed length that is at least $\min\{\Omega(m), \Omega(b)\}$, where $m$ is the extractor's output length and $b$ is the bound on the quantum storage. In contrast, classically, there are many explicit constructions with $poly(\log n)$ seed length (where $n$ is the extractor's input length), some even with logarithmic seed length. Some of these constructions are summarized in Table 1.1. A natural question that repeatedly appears in the above-mentioned papers is whether one can show a logarithmic seed length extractor against quantum storage.

In this work we show that a celebrated extractor construction due to Trevisan [31] is also good against quantum storage, with somewhat weaker parameters. Namely, we have the following theorem.

THEOREM 1.1. *There exists a constant $c$ such that for every $\alpha > \beta > 0$, $\gamma < \frac{\alpha-\beta}{c}$, $k = n^\alpha$, $b = n^\beta$, $\epsilon \geq n^{-\gamma}$, $m = O(\frac{\epsilon}{\log n}(\frac{k}{b})^{1/c})$, and every large enough $n$, there exists an explicit $(k, b, \epsilon)$ strong extractor $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ against quantum storage, with seed length $t = O(\log n)$.*[1]

In Theorem 1.1 we use the convention that $n$ is a free parameter, tending to infinity, and that the other parameters depend on $n$. That is, for every $n$, there exists a function $E_n : \{0,1\}^n \times \{0,1\}^{t(n)} \to \{0,1\}^{m(n)}$ that is a $(k(n), b(n), \epsilon(n))$ strong extractor against quantum storage. The construction is *explicit* if there exists a polynomial time algorithm that given $x \in \{0,1\}^n$ and $y \in \{0,1\}^t$ outputs $E(x, y)$.

The seed length in the construction of Theorem 1.1 is $O(\log n)$, and as such the construction gives the first solution to the privacy amplification problem against quantum storage with logarithmic seed length. There is a well-known lower bound showing that even extractors against classical storage require $\Omega(\log n)$ seed length, and so the construction is optimal up to multiplicative factors with respect to the *seed length*.

There are, however, other aspects of the construction that are far from being optimal, with the following three major deficiencies:

[1] The constant $c$ we currently achieve is $c = 15$.

TABLE 1.1

*Several explicit strong $(k, b, \epsilon)$ extractors against adversaries with b-bit classical storage and the proven parameters of these schemes against quantum storage. To simplify the parameters, the error $\epsilon$ is a constant and $k = n$. The last row gives information about the classical lower bound; it is possible that a better lower bound applies for extractors against quantum storage.*

| No. of truly random bits | No. of output bits | Against classical storage | Against quantum storage |
|---|---|---|---|
| $O(n)$ | $m = n - b - O(1)$ | Pair-wise independence [17] | ✓ [19] |
| $\Theta(m)$ | $m \le n - b - O(1)$ | Almost pair-wise ind. [29, 14] | ✓, based on [19] |
| $O(b + \log n)$ | $m = n - b - O(1)$ | Fourier analysis, collision [9] | ✓ [12] |
| $O(\frac{\log^2 n}{\log(n-b)})$ | $(n-b)^{1-\zeta}$ | Designs [31] | ✓, This paper. $m \approx \epsilon(\frac{n-b}{b})^{\Omega(1)}$<br>✓ [8, 7]. $m = (n-b)^{1-\zeta}$ |
| $O(\log n)$ | $m = \Omega(n-b)$ | [23, 15, 11] | ? |
| $O(\log n)$ | $m = (1 - o(1))(n-b)$ | [10] | ? |
| $\log n + O(1)$ | $m = n - b - O(1)$ | Nonexplicit construction [28] | ? |
| $\log n + O(1)$ | $m = n - b - O(1)$ | Lower bound [27, 28] | ✓ |

- The number of output bits $m$ depends multiplicatively on the error parameter $\epsilon$. This implies that the construction *cannot* handle small error parameters and completely fails when, e.g., $\epsilon \le 1/k$. This deficiency does not appear in the classical analysis of Trevisan's extractor, where the output length does not depend on $\epsilon$; it is $(k - b - \log \epsilon^{-1})^{1-\zeta}$ for arbitrarily small $\zeta$.
  This deficiency is quite severe, as the error parameter is of crucial importance; for example, in the bounded storage model we typically want to argue that the adversary does not learn any nonnegligible amount of information about the output, and for that we need extremely small error parameters.
- The number of output bits $m$ is at most $\frac{k}{b}$. This implies that the construction gives poor results against quantum adversaries with $b = \Omega(k)$ quantum storage. In contrast, in the classical analysis of Trevisan's extractor, the term $k - b$ replaces the much inferior term $\frac{k}{b}$, and as a result the extractor works well even against adversaries with $\Omega(k)$ classical storage.
- The last deficiency has to do with the output length. The classical analysis of Trevisan's extractor shows one can output $(k - b - \log \epsilon^{-1})^{1-\zeta}$ output bits for any constant $\zeta > 0$. Our quantum analysis can, at the very best, guarantee only $k^{1/c}$ output bits for some large constant $c$ ($c = 15$ is large enough), and so we have a polynomial loss here compared to the original classical scheme.

Following our work, De and Vidick [8] gave a better analysis of the performance of Trevisan's extractor against quantum adversaries, which was further simplified and generalized by De et al. [7]. Their work shows Trevisan's extractor is essentially as good against quantum adversaries as against classical adversaries. Specifically, the extractor can output $(k - b - \log \epsilon^{-1})^{1-\zeta}$ bits for any arbitrarily small $\zeta < 1$, thus solving the above three deficiencies.[2] Even more importantly, these results hold against a more general model of quantum adversaries. We refer the interested readers to [7] for more details.

Table 1.1 summarizes the parameters of several known explicit extractors against classical storage and the proven parameters of these schemes against quantum storage. Other classical techniques also adapt well to the quantum setting; see, e.g., [21, 3]. It is our belief that more classical extractor constructions are also good against quantum storage. We conjecture that there is a construction that is good against quantum

[2] We remark that De and Vidick declare only the inferior bound $m = \frac{(k-b-\log \epsilon^{-1})}{k^{\zeta}}$ for any $\zeta > 0$. Also, the better bound is only implicitly stated in [7], because the results there are stated for a more general model of adversaries.

adversaries and matches the parameters of the currently best-known classical extractor construction, or even the parameters of the lower bound. It is our hope that the recent chain of papers on the subject will eventually lead there.

**Our technique.** We begin the discussion with a short intuitive description of the ideas behind Trevisan's extractor.[3] In a nutshell, the construction has two parts:

- The first observation is that good extractors with a single-bit output exist and, in fact, are implied by the existence of good binary error correcting codes.
- One can take $m$ *independent* copies of an extractor with a single-bit output and get an extractor outputting $m$ bits. The price of this is that the seed length becomes $\Omega(m)$. To fix this, in Trevisan's extractor a short seed of length $O(\log n)$ is used to create $m$ sets that are *pairwise nearly disjoint*. The analysis shows that in the classical setting the $m$ nearly disjoint sets can replace the $m$ independent sets, resulting with $m$ output bits but only $O(\log n)$ seed length.

We now ask whether the above approach works in the presence of quantum adversaries.

We already know the answer for the first item: König and Terhal [22] proved that *any* extractor with a single-bit output is also good against quantum adversaries, with only a small loss in parameters. The second item is trickier. As before, taking $m$ *independent* copies of an extractor with a single-bit output results in an extractor that outputs $m$ bits good against quantum storage (this was formally shown in [22]), and, as before, the main drawback of this construction is that the seed length becomes large. The main question is whether Trevisan's derandomized version, using pairwise nearly disjoint sets, also works against adversaries having quantum storage.

The analysis of Trevisan's extractor uses the fact that it is built upon a *reconstructible* pseudorandom generator (PRG). Loosely speaking, in such structures any mechanism that breaks the extractor (i.e., distinguishes its output $E(x,U)$ from uniform) can be used together with short advice to reconstruct its input $x$. At first sight, this kind of reasoning looks very well suited to generalizations to extractors against quantum storage. Assume Charlie can distinguish the extractor's output $E(x,U)$ from uniform using $b$ qubits of storage. Then, the reconstruction property tells us we should be able to reconstruct $x$ using Charlie's reconstruction procedure, his $b$ qubits of information, and short advice of $a$ classical bits. Thus, it seems we should be able to reconstruct $x \in \{0,1\}^n$ using only $a+b$ qubits. Basic quantum information theory tells us then that $a+b \geq n$, or, putting it differently, whenever $b < n-a$, the extractor's output is close to uniform.

A fundamental problem that arises in the quantum setting is that quantum advice is fragile, and using it once degrades it. We tackle this problem by taking a variant of Trevisan's extractor where the binary error correcting code used for constructing the extractor with the single-bit output is replaced with a locally list-decodable binary code. The main advantage in this variant is that the number of queries the reconstruction algorithm makes to the eavesdropper Charlie is small (poly-logarithmic in $n$ rather than linear in $n$ as in Trevisan's algorithm). As the number of queries $q$ is small, we can take as advice $q$ copies of the system Charlie holds.

The price of this trick is that the reconstruction algorithm can now learn just one bit of the input $x$, rather than the whole input $x$ as in Trevisan's extractor.

[3] We comment that the presentation here is somewhat different than that in [31].

The good thing, though, is that the reconstruction algorithm can learn any bit of its choice. Thus, roughly speaking, if Charlie can break the extractor, then Charlie holds a $b$-qubit encoding of $x$ that lets him recover any single bit $x_i$ of his choosing. A well-known result about *random access codes* [2] shows this is impossible unless $b$ is linear in $n$. In particular, if $b$ is small enough, the extractor's output is close to uniform!

The actual analysis follows the above outline but is more involved, as it has to deal with technical details and also some substantial difficulties (e.g., the fact that the analysis requires random access codes of *subsets*).

The paper is organized as follows. In section 2 we define extractors against quantum storage. In section 3 we discuss two structures that we need for the construction and the analysis, namely, random access codes in section 3.1 and locally list-decodable codes in section 3.2. Finally, in section 4 we analyze a variant of Trevisan's extractor and show that it is good against quantum storage.

**2. Extractors against quantum storage.** We now give formal definitions. We assume familiarity with the basic notions of quantum computing (such as density matrices), and we refer the interested reader to [25] and [18] for extensive background on the subject. We also use the following notation: if $\phi$ is a (possibly entangled) state over registers $A$ and $B$, we write $\phi = \phi_A \circ \phi_B$. In the special case where $\phi$ is a product state over $A$ and $B$, we write $\phi = \phi_A \otimes \phi_B$.

DEFINITION 2.1. *An $(n, b)$ quantum encoding of some domain $\Lambda$ is a collection $\{\rho(x)\}_{x\in\Lambda}$ of density matrices $\rho(x)$ over a Hilbert space of dimension $2^b$.*

Let $X$ be some classical distribution over $\{0,1\}^n$, and let $p_x$ denote the probability $X$ assigns to $x$. Informally, an extractor is a function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ such that the extractor's output "looks uniform" to the adversary Charlie. More formally, the actual state of the system Charlie gets to see is

$$U_t \circ E(X, U_t) \circ \rho(X) \stackrel{\text{def}}{=} \sum_{x\in\{0,1\}^n, y\in\{0,1\}^t} p_x 2^{-t} \, |y, E(x,y)\rangle\langle y, E(x,y)| \otimes \rho(x) \tag{2.1}$$

and includes the public random string $y$, the extractor's output $E(x, y)$, and his own $b$-qubit information $\rho(x)$. The *ideal* state of the system, where Charlie learns no information about the output, is a *product* state where the extractor's output is *independent* of the information $\rho(x)$ Charlie knows; i.e., it is

$$U_{t+m} \otimes \rho(X) \stackrel{\text{def}}{=} \sum_{w\in\{0,1\}^{m+t}} 2^{-(m+t)} \, |w\rangle\langle w| \otimes \sum_{x\in\{0,1\}^n} p_x \rho(x). \tag{2.2}$$

Our goal is to find a function $E$ such that these two states are almost indistinguishable. One can formalize this by requiring either that the two states above are close to each other in the trace norm or, equivalently, that no observer can $\epsilon$-distinguish the two states (see, e.g., [25, Theorem 9.1]). We choose the latter option, as it is more algorithmic in nature and lends itself more easily to the quantum setting. We define the following.

DEFINITION 2.2. *A function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a $(k, b, \epsilon)$ strong extractor against quantum storage if for any classical distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$ and every $(n, b)$ quantum encoding $\{\rho(x)\}$, $U_t \circ E(X, U_t) \circ \rho(X)$ is $\epsilon$-indistinguishable from $U_{t+m} \circ \rho(X)$.*

In the definition above the min-entropy of $X$ is denoted $H_\infty(X)$ and is defined to be

$$H_\infty(X) \;=\; \min_{a:X(a)>0} 1/\log(X(a)).$$

If $H_\infty(X) \geq k$, then for all $a$ in its support $X(a) \leq 2^{-k}$. A distribution is *flat* if it is uniformly distributed over its support. Every distribution $X$ with $H_\infty(X) \geq k$ can be expressed as a convex combination $\sum \alpha_i X_i$ of flat distributions $X_i$, each with min-entropy at least $k$. As in the classical case, this implies that in Definition 2.2 we could have replaced the condition "for any classical distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$" with the condition "for any classical *flat* distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$."

We similarly define a $(k, b, \epsilon)$ strong extractor against *classical* storage, where we restrict the encoding $\rho(x)$ to storing only $b$ bits (rather than qubits) of information about $x$. If $b = 0$, we omit it and say $E$ is a $(k, \epsilon)$ strong extractor. The following well-known lemma shows that extractors that work well when $b = 0$ also work well against large *classical* storage.

LEMMA 2.3. *Let $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$. Let $k \geq b \geq 0$ and $\epsilon \geq 0$. If $E$ is a $(k - b - \log \epsilon^{-1}, \epsilon)$ strong extractor, then $E$ is a $(k, b, 2\epsilon)$ strong extractor against classical storage.*

*Proof.* Let $X$ be a flat distribution over $2^k$ elements, and let $\rho : \{0,1\}^n \to \{0,1\}^b$ be any classical encoding of $n$-bit strings to $b$ bits. Let $C$ denote the random variable containing the value $\rho(x)$. A simple calculation shows that

$$\Pr_{x \in X}[\; H_\infty(X|C = \rho(x)) \leq k - b - \log \epsilon^{-1} \;] \;\leq\; \epsilon.$$

Thus, except with probability $\epsilon$, $(X|C = \rho(x))$ has enough min-entropy, and therefore $(U_t \circ E(X, U_t) \mid C = c)$ is $\epsilon$ close to uniform. Thus $E$ is a $(k, b, 2\epsilon)$ strong extractor against classical storage. □

A $(k - b - \log \epsilon^{-1}, \epsilon)$ extractor is not necessarily a $(k, b, 2\epsilon)$ strong extractor against *quantum* storage. One formal reason the proof of Lemma 2.3 fails against quantum storage is that it is not clear how to define the conditional distribution $(X|C = \rho(x))$ when $\rho(x)$ is quantum. We know this is more than just a formal problem, as demonstrated by Gavinsky et al. [13].

Another way to look at the problem is as follows. An adversary has to fix some encoding $\rho(x)$ that determines the information he keeps about $x$. In the classical setting this encoding determines a distribution $(X|C = \rho(x))$, as in the proof of Lemma 2.3. Later, an independent random seed $y \in \{0,1\}^t$ is chosen and $E(x, y)$ is calculated. Thus, from the adversary's point of view, the seed $y$ is *independent* of the source $(X|C = \rho(x))$.

In the quantum world, however, things are not that simple. It is true that the adversary chooses the information $\rho(x)$ he keeps based solely on $x$, and $\rho(x)$ is independent of the random seed $y$. Later the adversary is shown the extractor's output $w = E(x, y)$ and is given the *choice* to decide which measurement to apply. The delicate issue is that the measurement *may depend* on the random seed $y$ and the extractor's output $w$. Each measurement $M = M_{y,w}$ and possible result $z$ define a classical distribution $X_{M,z} = (X \mid M(\rho(X) = z)$, and it is not difficult to show that almost always $X_{M,z}$ has high min-entropy. The problem is that the distributions $X_{M,z}$ depend on $y$, and so from the adversary's point of view the seed $y$ is correlated with the source $X_{M,z}$ it sees. Consequently, the extractor may fail, as indeed happens in the example shown by Gavinsky et al.

## 3. Some background on codes and random access codes.

**3.1. Random access codes.** A fundamental result in quantum information theory, Holevo's theorem [16], states that no more than $b$ classical bits of information can be faithfully transmitted by transferring $b$ quantum bits from one party to another. Formally, we have the following theorem.

THEOREM 3.1 (Holevo). *Let $\{\rho(x)\}$ be any $(n,b)$ quantum encoding. Let $X$ be a random variable with distribution $\{p_x\}$, and let $\rho(X) = \mathrm{E}_x\rho(x) = \sum_x p_x\rho_x$. If $Y$ is any random variable obtained by performing a measurement on the encoding, then $I(X:Y) \leq S(\rho(X)) - \mathrm{E}_x S(\rho_x) \leq S(\rho(X))$.*

In Theorem 3.1 $I(:)$ stands for the mutual information function, and $S(\cdot)$ stands for von Neumann entropy; see, e.g., [25, Chapter 11].

In view of this result, it is tempting to conclude that the exponentially many degrees of freedom latent in the description of a quantum system must necessarily stay hidden or inaccessible. However, the situation is more subtle since the recipient of the $n$-qubit quantum state can choose which measurement he uses to extract information about his state. In general, these measurements do not commute. Thus making a particular measurement will disturb the system, thereby destroying some or all of the information that would have been revealed by another possible measurement. Indeed, Ambainis et al. [1, 2] ask whether there exists an $(n,b)$ quantum encoding $\{\rho(x)\}$ such that the recipient can learn any bit $x_i$ of his choice. That is, they define the following.

DEFINITION 3.2 (see [1, 2]). *An $n \overset{p}{\mapsto} t$ quantum* random access *encoding is an $(n,t)$ encoding $\{\rho(x)\}_{x\in\{0,1\}^n}$ such that for every $1 \leq i \leq n$, there is a POVM $\mathcal{E}^i = \{\mathcal{E}^i_0, \mathcal{E}^i_1\}$ such that for all $x \in \{0,1\}^n$ we have $\mathrm{Tr}(\mathcal{E}^i_{x_i} f(x)) \geq p$.*[4]

Nayak and Ambainis et al. [24, 2] show that any quantum $n \overset{p}{\mapsto} t$ encoding must have $t \geq (1 - H(p))n$, where $H(\cdot)$ is the Shannon entropy function. In fact, this lower bound also holds if we relax the worst-case condition $\forall_x \forall_i \mathrm{Tr}(\mathcal{E}^i_{x_i} f(x)) \geq p$ and replace it with the average-case condition $\forall_x \mathbf{E}_i \mathrm{Tr}(\mathcal{E}^i_{x_i} f(x)) \geq p$.

In this paper we need random access codes that are defined for *subsets* of $\{0,1\}^n$ as follows.

DEFINITION 3.3. *Let $\mathcal{F} \subseteq \{0,1\}^n$. An $\mathcal{F} \overset{p}{\mapsto} t$ quantum* random access *encoding is an $(n,t)$ encoding $\{\rho(x)\}_{x\in\mathcal{F}}$ such that for every $1 \leq i \leq n$, there is a POVM $\mathcal{E}^i = \{\mathcal{E}^i_0, \mathcal{E}^i_1\}$ such that for all $x \in \mathcal{F}$, $i \in [n]$ we have $\mathrm{Tr}(\mathcal{E}^i_{x_i} f(x)) \geq p$.*

We prove the following theorem.

THEOREM 3.4. *Let $\mathcal{F} \subseteq \{0,1\}^n$.*

1. *For any $\delta \geq 0$, any quantum $\mathcal{F} \overset{\frac{1}{2}+\delta}{\mapsto} t$ encoding satisfies $t \geq \Omega(\frac{\delta^2}{\log n} \cdot \log|\mathcal{F}|)$.*
2. *For any $\delta \geq \frac{1}{n}$, any quantum $\mathcal{F} \overset{1-\delta}{\mapsto} t$ encoding satisfies $t \geq \Omega(\frac{\log(1/4\delta)}{\log n} \cdot \log|\mathcal{F}|)$.*

*Proof.* We use the proof technique of [1]. First, one can turn the $\mathcal{F} \overset{\frac{1}{2}+\delta}{\mapsto} t$ encoding into another $\mathcal{F} \overset{1-\epsilon}{\mapsto} t \cdot T$ encoding, with $T = O(\log \epsilon^{-1}/\delta^2)$, as follows. The new encoding is $T$ copies of the original encoding. The decoding is the majority vote

[4] We briefly recall some quantum computation definitions; for more details see, e.g., [25, section 2.2.6]. A positive operator value measure (POVM) is the most general formulation of a measurement in quantum computation. A POVM on a Hilbert space $\mathcal{H}$ is a collection $\{E_i\}$ of positive semidefinite operators $E_i : Hom(\mathcal{H},\mathcal{H}) \to Hom(\mathcal{H},\mathcal{H})$ that sum up to the identity transformation, i.e., $E_i \succeq 0$ and $\sum E_i = I$. Applying a POVM $\{E_i\}$ on a density matrix $\rho$ results in answer $i$ with probability $\mathrm{Tr}(E_i\rho)$.

over the $T$ decodings of the $T$ copies. By a standard Chernoff bound, the probability of error is at most $\epsilon$.

Fix $\epsilon = \frac{c}{n^2}$ for some constant $c$ that will be determined later. Consider some $f \in \mathcal{F}$ and its encoding $\rho = \rho(f)$. For every $i \in [n]$ the measurement $\mathcal{E}^i$ recovers $f_i$ with probability at least $1 - \epsilon$, i.e., almost with certainty. It is shown in [1][5] that sequentially applying the measurements $\mathcal{E}^1, \ldots, \mathcal{E}^n$, the result $(f_1, \ldots, f_n)$ is obtained with probability at least $1 - 4n\sqrt{\epsilon} = 1 - 4\sqrt{c}$. Taking $c$ small enough, we recover the whole string $f$ with probability at least $\frac{1}{2}$. By Holevo's theorem, $Tt \geq I(U_{\mathcal{F}} : Y) \geq \frac{1}{2}\log(|\mathcal{F}|)$.

For the second item notice that for any $\delta \geq \epsilon$ one can turn an $\mathcal{F} \overset{1-\delta}{\mapsto} t$ encoding into another $\mathcal{F} \overset{1-\epsilon}{\mapsto} O(t \cdot T)$ encoding, using $T = 2\log_{4\delta} \epsilon$, and the rest is as before. ∎

Regev showed us an example where the bound in Theorem 3.4 is tight. Partition the $n$ bits to $\sqrt{n}$ blocks, each of size $\sqrt{n}$. Take the set $\mathcal{F}$ to be all the bit strings containing exactly one 1 in each block. $\mathcal{F}$ has $\Theta(\sqrt{n} \cdot \log n)$ entropy. Yet, consider the following random access code that uses only $O(\sqrt{n} + \log n)$ bits. Given $f \in \mathcal{F}$ with indices $i_1, \ldots, i_{\sqrt{n}}$ (i.e., index $i_j$ is 1 in the $j$th block), the random access code encodes $f$ by $(h, h(i_1), \ldots, h(i_k))$, where $h : [\sqrt{n}] \to [10]$ is randomly chosen from a family of pairwise independent hash functions. When asked for a bit $t$ of the input, say, from the $j$th block, the decoder just checks whether $h(t) = h(i_j)$. It outputs 1 if yes; otherwise it outputs 0. By the pairwise independent property, we output the correct answer with probability 2/3 for each question.

We proved Theorem 3.4 with a definition that is worst-case over $i$. We remark that the average case version is false. For example, if $\mathcal{F}$ is the set of all $n$-bit strings of weight at least $\frac{2}{3}n$, there is a trivial random access code of length zero that for all $f \in \mathcal{F}$ succeeds on average over $i$ with probability at least 2/3. Thus, in this case there is a crucial difference between worst-case and average-case complexity over $i$.

**3.2. Local list-decoding.** A code is a function $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$, where $\bar{n}$ is the code *length* and $n$ is its *dimension*. We identify a code $\mathcal{C}$ with its image $\mathcal{C} = \{\mathcal{C}(x) \mid x \in \Sigma^n\}$. The distance $d$ of the code is the minimum Hamming distance between two codewords in $\mathcal{C}$. The balls of radius $\frac{d-1}{2}$ around codewords are disjoint, and therefore one can uniquely correct up to so many errors. If we allow more than $d/2$ errors, several decodings are possible. In many cases one can allow almost up to the distance errors and still get only a few possible decodings. This leads to the following definition.

DEFINITION 3.5 (list-decoding). *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. We say $\mathcal{C}$ is $(p, L)$ list-decodable if for every $z \in \Sigma^{\bar{n}}$ there are at most $L$ codewords $y$ such that $ag(z, y) \overset{\text{def}}{=} |\{i \in [\bar{n}] | z_i = y_i\}| \geq p\bar{n}$.*

We now identify elements of $y \in \Sigma^{\bar{n}}$ with functions $y : [\bar{n}] \to \Sigma$. A query to a function $y : [\bar{n}] \to \Sigma$ is a value $i \in [n]$, and the answer to the query is $y(i)$ (when using function notation) or, equivalently, $y_i$ (when using string notation). We say a decoding algorithm is *local* if it makes only a few queries to the corrupted word. Formally, we have the following definition.

DEFINITION 3.6 (local list-decoding). *Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. We say $\mathcal{C}$ has a $(p, L, q, \beta)$ local list-decoding if $\mathcal{C}$ is $(p, L)$ list-decodable, and the following hold:*

- *There exists a probabilistic, polynomial time oracle machine $A$ that on input $k \in [L]$ and $i \in [n]$ outputs a Boolean value. $A$ can make at most $q$ queries*

[5] Implicit in the proof of Lemma 4.2 in [1].

*and each query is in the range* $[\bar{n}]$.

- *For every deterministic function* $y : [\bar{n}] \to \Sigma$ *and every* $x \in \Sigma^n$ *such that* $ag(y, \mathcal{C}(x)) \geq p\bar{n}$, *there exists* $k \in [L]$ *such that for every* $i \in [n]$, $\Pr_A[A^y(k,i) = x(i)] \geq \beta$ *(where* $A^y(k,i)$ *denotes the output of the oracle machine* $A$ *on input* $(k,i)$ *when the oracle queries are answered according to the function* $y : [\bar{n}] \to \Sigma$*).*

Sudan, Trevisan, and Vadhan proved the following theorem.

THEOREM 3.7 (see [30]). *For every* $\delta = \delta(n) > 0$, *there exists an explicit binary code of dimension* $n$, *length* $\bar{n} = poly(n, \frac{1}{\delta})$, *and* $poly(\bar{n})$ *encoding time, that is* $(p = \frac{1}{2} + \delta, L = poly(\bar{n}), q = poly(\log n, \frac{1}{\delta}), \beta = 1 - \delta)$ *locally list-decodable.*[6]

In our case we have access not to a deterministic function $y : [\bar{n}] \to \Sigma$ but rather to a probabilistic procedure that has high on average success probability; i.e., we are given access to a probabilistic oracle $O : [\bar{n}] \to \Sigma$. For $y : [\bar{n}] \to \Sigma$ define $ag(O, y) \stackrel{\text{def}}{=} \Pr_{i \in [\bar{n}], O}(O(i) = y(i))$. We would like to do local list-decoding when given access to $O$. Formally, we have the following definition.

DEFINITION 3.8 (probabilistic oracle, local list-decoding). *Let* $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. *We say that* $\mathcal{C}$ *has a* $(p, L, q, \beta)$ probabilistic oracle, *local list-decoding if* $\mathcal{C}$ *is* $(p, L)$ *list-decodable, and the following hold:*

- *There exists a probabilistic, polynomial time oracle machine* $A$ *that on input* $k \in [L]$ *and* $i \in [n]$ *outputs a Boolean value.* $A$ *can make at most* $q$ *queries, and each query is in the range* $[\bar{n}]$.
- *For every probabilistic oracle* $O : [\bar{n}] \to \Sigma$ *and every* $x \in \Sigma^n$ *such that* $ag(O, \mathcal{C}(x)) \geq p\bar{n}$, *there exists* $k \in [L]$ *such that for every* $i \in [n]$, $\Pr[A^O(k,i) = x(i)] \geq \beta$.

If we are interested only in list-decoding with no restriction on the number of queries, then list-decoding a probabilistic oracle is essentially the same as list-decoding a string. This is because we can take $O$ and for every query $j \in [\bar{n}]$ sample $y_j = O(j)$. By a standard Chernoff bound, with high probability, the sampled string $y$ also has high agreement with $\mathcal{C}(x)$, and therefore the string $x$ appears somewhere in the output list of $y$.

The above argument does *not* work for *local* list-decoding. Here we need the index $k$ to depend on $O$ alone, and not on the sampled string $y$ or the index $i$. This is an essential requirement, as in local list-decoding we do not reconstruct the whole string $x$, but rather a single bit $x_i$ of it. The above argument therefore does not work, as it may happen that the index of $x$ in the list of $y$ depends on the sampled string $y$, and not just on $O$, as required by the definition.

Luckily, going back to the construction of [30], one can check that essentially the same analysis shows the following.[7]

THEOREM 3.9 (based on [30]). *For every* $\delta = \delta(n) > 0$, *there exists an explicit binary code, of dimension* $n$, *length* $\bar{n} = poly(n, \frac{1}{\delta})$, *and encoding time* $poly(\bar{n})$, *that*

[6] The code in [30] is a Reed–Muller code concatenated with the Hadamard code. The list-decoding algorithm first list-decodes the Hadamard code and then uses the result to list-decode the Reed–Muller code. Working out the parameters, we get that the field size is $|F| = O(\frac{\log^2 n}{\delta^5})$. The local list-decoding algorithm presented in [30] has $|F|^3$ queries and solves the local list-decoding problem worst-case over $i$. We remark that using a better inner code the query complexity can be reduced.

[7] This is because the advice for $x$ is a point $v$ and a value $\sigma$ such that $\widehat{x}(v) = \sigma$, where $\widehat{x}$ is the low-degree extension of $x$, and with high probability such advice separates, for *most* of the sampled strings $y$, the true codeword $\mathcal{C}(x)$ from the other codewords that arise from $y$.

*is* $(p = \frac{1}{2} + \delta, L = poly(\bar{n}), q = poly(\log n, \frac{1}{\delta}), \beta = 1 - \delta)$ *probabilistic oracle, local list-decodable.*

**4. Extractors against quantum storage from black-box PRGs.** We now reach Trevisan's argument and its refinement to the quantum setting. Roughly speaking, Trevisan shows that the Nisan–Wigderson PRG implies a corresponding extractor. We need, of course, to define what a PRG is, and, in fact, it will turn out that we need two variants of the notion which are PRGs with a good on average reconstruction algorithm and black-box PRGs. Indeed, we will define these notions soon. However, for the time being we continue with a high-level description, and we first give a schematic description of the work presented in this section.

Trevisan showed the following two claims:

- A PRG with a good-on-average reconstruction algorithm can be converted to a black-box PRG, and
- black-box PRGs give rise to good classical extractors.

Trevisan then used the Nisan–Wigderson PRG, which has a good-on-average reconstruction algorithm, to derive a corresponding extractor.

Using the above terminology we show that

1. PRGs with a good-on-average reconstruction algorithm can be converted to black-box PRGs with *a few queries*, and
2. black-box PRGs with *a few queries* give rise to good extractors against *quantum storage*.

Then, using the Nisan–Wigderson PRG that has a good-on-average reconstruction algorithm, we derive a corresponding extractor that is good against quantum storage.

The claim in item 1 above is purely classical, and we prove it by replacing the list-decodable, binary error correcting codes that Trevisan uses with locally list-decodable binary error correcting codes. The claim in item 2 above constructs extractors against quantum storage, and its proof uses the lower bound on random access codes presented in section 3.1.

The section is organized as follows: in section 4.1 we define the various variants of PRGs that we need. We then prove the first claim in section 4.2; namely, we prove that PRGs with a good-on-average reconstruction algorithm can be converted to black-box PRGs with *a few queries*. Finally, in section 4.3 we prove that black-box PRGs with *a few queries* give rise to good extractors against *quantum storage*, which is perhaps the central idea underlying the paper.

**4.1. PRGs.** A PRG is a mapping $G : \{0,1\}^t \to \{0,1\}^m$ that extends a short string to a longer one. A PRG "$\epsilon$-fools" a boolean test $T$ if

$$\left| \Pr_{x \in \{0,1\}^t}[T(G(x)) = 1] - \Pr_{y \in \{0,1\}^m}[T(y) = 1] \right| \leq \epsilon.$$

One way of constructing such PRGs is by starting with some function $f : [n] \to \{0,1\}$ that is "hard" for some model of computation and showing that if $G^f$ does not fool some test $T$, then the test $T$ can be used for efficiently computing the hard function $f$, thus leading to a contradiction. More formally, we have the following definition.

DEFINITION 4.1. *We say* $(G, R)$ *is a pair consisting of a black-box PRG* $G$ *and a reconstruction algorithm* $R$ *with* $a$ *advice bits and* $q$ *queries if the following hold:*

- $G : \{0,1\}^t \to \{0,1\}^m$ *is a classical oracle machine with oracle calls to a function* $f : [n] \to \{0,1\}$.

- *$R$ is a classical oracle circuit with inputs $adv \in \{0,1\}^a$ and $i \in [n]$, making at most $q$ queries to its oracle.*

A pair $(G,R)$ is $(\epsilon,p)$ pseudorandom if any test $T$ that $\epsilon$-breaks the PRG $G^f$ can be used for efficiently computing $f(i)$ with success probability $p$ (this is the "reconstruction" step). This definition comes in two varieties: in one $f(i)$ is computed worst-case over $i$, and in the other $f$ is correctly computed on most inputs $i$, but not necessarily on all of them. Formally, we have the following definition.

DEFINITION 4.2. *Let $(G,R)$ be as above.*

- *We say $(G,R)$ is a black-box $(\epsilon,p)$-PRG if, for every Boolean function $f : [n] \to \{0,1\}$ and every probabilistic oracle $T : \{0,1\}^{t+m} \to \{0,1\}$ that $\epsilon$-distinguishes $U_t \circ G^f(U_t)$ from uniform, there exists advice $adv = adv(T,f) \in \{0,1\}^a$ such that for all $i \in [n]$, $\Pr[R^T(adv;i) = f(i)] \geq p$, where the probability is over the internal coins of $R$ and $T$.*
- *We say $(G,R)$ is a black-box $(\epsilon,p)$-PRG with* average-case *reconstruction if, for every Boolean function $f : [n] \to \{0,1\}$ and every probabilistic oracle $T : \{0,1\}^{t+m} \to \{0,1\}$ that $\epsilon$-distinguishes $U_t \circ G^f(U_t)$ from uniform, there exists advice $adv = adv(T,f) \in \{0,1\}^a$ such that $\Pr[R^T(adv;i) = f(i)] \geq p$, where the probability is over a uniform $i \in [n]$ and the internal coins of $R$ and $T$.*

Sometimes we omit $R$ and say that $G$ is a black-box $(\epsilon,p)$-PRG, meaning that there exists some reconstruction algorithm $R$ such that $(G,R)$ is an $(\epsilon,p)$-PRG.

**4.2. A black-box PRG with a few queries.** Nisan and Wigderson [26] constructed a black-box PRG with average-case reconstruction. Specifically, for every $\epsilon > 0$, $\mathrm{NW}^{f:[n]\to\{0,1\}} : \{0,1\}^t \to \{0,1\}^m$ has $(\epsilon, p = \frac{1}{2} + \frac{\epsilon}{2m})$ average-case reconstruction with $a = O(m^2)$ advice bits and $t = O(\frac{\log^2 n}{\log m})$ seed length. The Nisan–Wigderson reconstruction algorithm uses exactly one oracle call to the distinguishing algorithm.

Trevisan showed how to combine a PRG with average-case reconstruction together with a good list-decodable binary code to get a black-box PRG. We need a similar result, except that we have the additional requirement that the reconstruction algorithm should make only a few queries. We achieve this goal by replacing the list-decodable codes Trevisan uses with probabilistic oracle, locally list-decodable code. We prove the following lemma.

LEMMA 4.3 (worst-case to average-case reduction for black-box PRG using only a few queries). *Assume $(G,R)$ is a black-box $(\epsilon, \frac{1}{2}+\delta)$-PRG with average-case reconstruction using $a$ advice bits and one query. Let $\mathcal{C}$ be a $(p = \frac{1}{2}+\delta, L, q, \beta)$ probabilistic oracle, locally list-decodable binary code. Define $\mathrm{TR}^f(y) = \mathrm{NW}^{\mathcal{C}(f)}(y)$. Then* TR *is a black-box $(\epsilon,\beta)$-PRG with $a + \log L$ advice bits and $q$ queries.*

*Proof.* Let $\bar{f} = \mathcal{C}(f)$. Suppose $T$ $\epsilon$-breaks the PRG $\mathrm{TR}^f = \mathrm{NW}^{\bar{f}}$. As $\mathrm{NW}^{\bar{f}}$ has average-case reconstruction, given the right advice $adv = adv(f,T)$ to $R$, $R^T(adv;i)$ computes $\bar{f}_i$ with average success probability $p = \frac{1}{2} + \delta$ over $i \in [\bar{n}]$ using a single query to $T$. We can view $R^T(adv;\cdot)$ as a probabilistic oracle from $[\bar{n}]$ to $\{0,1\}$ having $\frac{1}{2}+\delta$ agreement with $\bar{f}$.

As $\mathcal{C}$ is a $(p = \frac{1}{2}+\delta, L, q, \beta)$ probabilistic oracle, locally list-decodable binary code, there exists a value $k \in [L]$ such that the local list-decoding algorithm computes $f$ worst-case over $i$, given $k$ and the probabilistic oracle $R^T(adv;\cdot)$. The advice to the new reconstruction algorithm $\overline{R}$ includes the string $adv$ and the value $k \in [L]$.

Now assume that we ask $\overline{R}$ for the value of $f_i$, $i \in [n]$, i.e., that we wish to compute $\overline{R}^T(adv,k;i)$. We do this as follows. We apply the probabilistic oracle, locally list-

decoding algorithm of $\mathcal{C}$ and evaluate the $q$ queries $i_1, \ldots, i_q \in [\bar{n}]$ to $\bar{f} = \mathcal{C}(f)$ that it is going to make. We answer the $j$th query with the probabilistic oracle $R^T(adv; i_j)$, and we output the local decoding result. By the probabilistic oracle, locally list-decoding property, for every $i \in [n]$ the reconstruction oracle $\overline{R}^T$ outputs the right answer with probability at least $\beta$. □

Plugging in the parameters of the Nisan–Wigderson PRG, we get the following theorem.

THEOREM 4.4. *Let $\epsilon > 0$, $m \leq n$. There exists an explicit black-box $(\epsilon, 1 - \frac{1}{m})$ PRG $G^{f:[n]\to\{0,1\}} : \{0,1\}^t \to \{0,1\}^m$ with $a = O(m^2 + \log \frac{n}{\epsilon})$ advice bits, $t = O(\frac{\log^2 \frac{n}{\epsilon}}{\log m})$ seed length, and $q = \mathrm{poly}(\log n, \frac{m}{\epsilon})$ queries.*

*Proof.* Let $\epsilon > 0$, $m \leq n$. Let $\mathrm{NW}^{f:[n]\to\{0,1\}} : \{0,1\}^{t'} \to \{0,1\}^m$ be the Nisan–Wigderson PRG with $a = O(m^2)$ advice bits and $t' = O(\frac{\log^2 n}{\log m})$ seed length. Nisan and Wigderson showed that $\mathrm{NW}^f$ is a black-box $(\epsilon, \frac{1}{2} + \delta)$ PRG with average reconstruction and $\delta = \frac{\epsilon}{2m}$.

Let $\mathcal{C}$ be the $(p = \frac{1}{2} + \delta, L = poly(\bar{n}), q = poly(\log n, \frac{1}{\delta}), \beta = 1 - \delta)$ probabilistic oracle, locally list-decodable binary code of Theorem 3.9, where $\bar{n} = \mathrm{poly}(n/\delta)$. Define $\mathrm{TR}^{f:[n]\to\{0,1\}} : \{0,1\}^{\bar{t}} \to \{0,1\}^m$ by $\mathrm{TR}^f(y) = \mathrm{NW}^{\mathcal{C}(f)}(y)$ with $\bar{t} = O(\frac{\log^2 \bar{n}}{\log m}) = O(\frac{\log^2 \frac{n}{\delta}}{\log m})$ seed length. By Lemma 4.3 TR is a black-box $(\epsilon, 1 - \delta)$ PRG with $a = O(m^2 + \log \frac{n}{\epsilon})$ advice bits and $q$ queries. Finally, note that a black-box $(\epsilon, 1 - \delta)$ PRG is in particular a black-box $(\epsilon, 1 - \frac{1}{m})$ PRG. □

**4.3. Black-box PRGs yield extractors against quantum storage.** Trevisan [31] showed that black-box PRGs give rise to extractors. We show that they actually give rise to extractors against quantum storage; alas their quality depends on the number of oracle calls the reconstruction algorithm makes.

PROPOSITION 4.5 (generalizing [31]). *Suppose $(G, R)$ is a black-box $(\epsilon, p = 1 - \frac{1}{m})$-PRG with $a$ advice bits and $q$ queries. For $n > m$ define $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ by*

$$E(f, y) = G^f(y).$$

*Then $E$ is a $(k, b, 2\epsilon)$ strong extractor against quantum storage for $k = \Omega(\frac{\log n}{\log m}(a + qb)) + \log \epsilon^{-1}$.*

*Proof.* Let $T$ be a quantum test using $b$ qubits of side information $\rho$. Let $\mathcal{F}$ be the set of all functions $f \in \{0,1\}^n$ for which $T$ $\epsilon$-distinguishes

$$U_t \circ E(f, U_t) \circ \rho(f) \stackrel{\mathrm{def}}{=} \sum_{y \in \{0,1\}^t} 2^{-t} |y, E(f,y)\rangle\langle y, E(f,y)| \otimes \rho(f)$$

from

$$U_{t+m} \otimes \rho(f) \stackrel{\mathrm{def}}{=} \sum_{w \in \{0,1\}^{m+t}} 2^{-(m+t)} |w\rangle\langle w| \otimes \rho(f).$$

Clearly,

$$|\Pr[T(U_t \circ E(X, U_t) \circ \rho(X)) = 1] - \Pr[T(U_{t+m} \otimes \rho(X)) = 1]| \leq \epsilon + \Pr_{x \in X}[x \in \mathcal{F}],$$

where we have used the notation set in (2.1) and (2.2).

We will show $|\mathcal{F}| = 2^{O((a+qb)\frac{\log n}{\log m})}$. It will then follow that for any $X \subseteq \{0,1\}^n$, $E$ is a $(\log \frac{|\mathcal{F}|}{\epsilon}, b, 2\epsilon)$ strong extractor against quantum storage, as promised.

We are left to show that $\mathcal{F}$ is indeed small. Recall that $G$ is an $(\epsilon, p)$ PRG and for any $f \in \mathcal{F}$, $T$ $\epsilon$-distinguishes $U_t \circ E(f, U_t) \circ \rho(f)$ from $U_{t+m} \otimes \rho(f)$. Thus, by Definition 4.2, for every $f \in \mathcal{F}$ there exists an advice $adv = adv(T, f) \in \{0,1\}^a$ such that the reconstruction circuit $R^T(adv; \cdot)$ computes $f : [n] \to \{0,1\}$ with $q$ queries to $T$ and worst-case (over $i$) success probability $p$. We replace each of the $q$ queries to $T$ with a quantum circuit acting on its classical input and an independent $b$-qubit state that is initialized to $\rho(f)$. Altogether, the new circuit uses $qb$ qubits of side information. Notice that because the inputs to the different queries are in product state, the answers to the $T$ queries are independent. The resulting quantum circuit recovers the bits of $f : [n] \to \{0,1\}$ with probability $p$, *worst-case* over $i$. Thus, $\mathcal{F}$ has a random access code of length $a + qb$ and worst-case success probability $p = 1 - \frac{1}{m}$. By Theorem 3.4, item 2, $a + qb = \Omega(\frac{\log m}{\log n} \log |\mathcal{F}|)$, and so $|\mathcal{F}| = 2^{O((a+qb)\frac{\log n}{\log m})}$, as promised. □

Plugging Theorem 4.4 into Proposition 4.5, we get Theorem 1.1.

**Acknowledgments.** I would like to thank Avraham Ben-Aroya, Ashwin Nayak, Oded Regev, and Pranab Sen for stimulating talks on the subject. I also thank Oded for the example showing that the bound of Theorem 3.4 is tight.

## REFERENCES

[1] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, *Dense quantum coding and quantum finite automata*, in STOC, ACM, New York, 1999, pp. 376–383.

[2] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, *Dense quantum coding and quantum finite automata*, J. ACM, 49 (2002), pp. 496–511.

[3] A. Ben-Aroya and A. Ta-Shma, *Better Short-Seed Extractors against Quantum Knowledge*, preprint, http://arxiv.org/abs/1004.3737, 2010.

[4] C. H. Bennett, G. Brassard, C. Crepeau, and U. Maurer, *Generalized privacy amplification*, IEEE Trans. Inform. Theory, 41 (1995), pp. 1915–1923.

[5] C. H. Bennett, G. Brassard, and J.-M. Robert, *Privacy amplification by public discussion*, SIAM J. Comput., 17 (1988), pp. 210–229.

[6] M. Christandl, R. Renner, and A. Ekert, *A Generic Security Proof for Quantum Key Distribution*, preprint, http://arxiv.org/abs/quant-ph/0402131, 2004.

[7] A. De, C. Portmann, T. Vidick, and R. Renner, *Trevisan's Extractor in the Presence of Quantum Side Information*, preprint, http://arxiv.org/abs/0912.5514, 2009.

[8] A. De and T. Vidick, *Near-optimal extractors against quantum storage*, in STOC, ACM, New York, 2010, pp. 161–170.

[9] Y. Dodis and A. Smith, *Correcting errors without leaking partial information*, in STOC, ACM, New York, 2005, pp. 654–663.

[10] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan, *Extensions to the method of multiplicities, with applications to Kakeya sets and mergers*, in FOCS, IEEE, Washington, DC, 2010, pp. 181–190.

[11] Z. Dvir and A. Wigderson, *Kakeya sets, new mergers and old extractors*, in FOCS, IEEE, Washington, DC, 2008, pp. 625–633.

[12] S. Fehr and C. Schaffner, *Randomness Extraction via Delta-Biased Masking in the Presence of a Quantum Attacker*, preprint, http://arxiv.org/abs/0706.2606, 2007.

[13] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf, *Exponential separations for one-way quantum communication complexity, with applications to cryptography*, in STOC, ACM, New York, 2007, pp. 516–525.

[14] O. Goldreich and A. Wigderson, *Tiny families of functions with random properties: A quality-size trade-off for hashing*, Random Structures Algorithms, 11 (1997), pp. 315–343.

[15] V. Guruswami, C. Umans, and S. Vadhan, *Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes*, in Computational Complexity, IEEE, Washington, DC, 2007, pp. 96–108.

[16] A. S. Holevo, *Some estimates of the information transmitted by quantum communication channels*, Probl. Inf. Transm., 9 (1973), pp. 177–183.
[17] R. Impagliazzo, L. Levin, and M. Luby, *Pseudo-random generation from one-way functions*, in STOC, ACM, New York, 1989, pp. 12–24.
[18] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation*, AMS, Providence, RI, 2002.
[19] R. König, U. Maurer, and R. Renner, *On the Power of Quantum Memory*, preprint, http://arxiv.org/abs/quant-ph/0305154, 2003.
[20] R. König, U. Maurer, and R. Renner, *On the power of quantum memory*, IEEE Trans. Inform. Theory, 51 (2005), pp. 2391–2401.
[21] R. König and R. Renner, *Sampling of Min-Entropy Relative to Quantum Knowledge*, preprint, http://arxiv.org/abs/0712.4291, 2007.
[22] R. König and B. Terhal, *The bounded-storage model in the presence of a quantum adversary*, IEEE Trans. Inform. Theory, 54 (2008), pp. 749–762.
[23] C. Lu, O. Reingold, S. Vadhan, and A. Wigderson, *Extractors: Optimal up to constant factors*, in STOC, ACM, New York, 2003, pp. 602–611.
[24] A. Nayak, *Optimal lower bounds for quantum automata and random access codes*, in FOCS, IEEE, Washington, DC, 1999, pp. 369–376.
[25] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.
[26] N. Nisan and A. Wigderson, *Hardness vs. randomness*, J. Comput. System Sci., 49 (1994), pp. 149–167.
[27] N. Nisan and D. Zuckerman, *Randomness is linear in space*, J. Comput. System Sci., 52 (1996), pp. 43–52.
[28] J. Radhakrishnan and A. Ta-Shma, *Bounds for dispersers, extractors, and depth-two superconcentrators*, SIAM J. Discrete Math., 13 (2000), pp. 2–24.
[29] A. Srinivasan and D. Zuckerman, *Computing with very weak random sources*, SIAM J. Comput., 28 (1999), pp. 1433–1459.
[30] M. Sudan, L. Trevisan, and S. Vadhan, *Pseudorandom generators without the xor lemma*, J. Comput. System Sci., 62 (2001), pp. 236–266.
[31] L. Trevisan, *Extractors and pseudorandom generators*, J. ACM, 48 (2001), pp. 860–879.