Short seed extractors against quantum storage

Amnon Ta-Shma[*]

October 10, 2008

Abstract

Some, but not all, extractors resist adversaries with limited quantum storage. In this paper we show that Trevisan's extractor has this property, thereby showing an extractor against quantum storage with logarithmic seed length.

1 Introduction

In the classical *privacy amplification* problem Alice and Bob share information that is only partially secret towards an eavesdropper Charlie. Their goal is to distill this information to a shorter string that is completely secret. The problem was introduced in [4, 3]. The classical privacy amplification problem can be solved almost optimally using extractors.[1]

An interesting variant of the problem, where the eavesdropper Charlie is allowed to keep quantum information, was introduced by Konig, Maurer and Renner [14, 15]. Let us call such an extractor *an extractor against quantum storage*.[2] This situation naturally occurs in analyzing the security of some quantum key distribution (QKD) protocols and in bounded-storage cryptography. For example, [5] show a generic way of using extractors against quantum storage to prove the security of certain QKD protocols. Using extractors for bounded-storage cryptography demands more from the extractor (it should be "locally computable"), but also allows more specific assumptions about the source distribution (e.g., [17] and [16]).

Special cases of the problem are also of great interest. The first such example appears in [1, 19, 2] where random access codes are studied. Alice and Bob share a random length n string x on which the eavesdropper Charlie knows b bits of information. If Charlie is classical, then choosing a random $i \in [n]$ and outputting x_i results in an almost uniform bit. The question studied in the above papers is wether the same also holds when Charlie is quantum and may hold b *quantum bits*. It was shown in [1, 19, 2] that the answer is positive, and this gives an extractor against quantum storage, albeit, with a *single* output bit.

Konig, Maurer and Renner [14, 15] show that the pair-wise independent extractor of [13] is also good (and with the same parameters) against quantum storage. Using the same techniques the result can also be extended to using almost pair-wise independence [22, 10]. Another classical extractor for very high min-entropies was shown to hold against quantum storage in [8] (the classical version appears, e.g., in [6]). Konig and Terhal [17] showed that any single output extractor is also good against quantum storage. They also showed that any extractor with error ϵ, has at most $2^{O(b)}\epsilon$ error against b quantum storage. Thus, if some extractor has a good dependence on the error (as is often the case) one can make the extractor good against b quantum storage by taking a longer seed (often, longer by only $O(b)$ bits).

It is tempting to conjecture that every extractor against classical storage should also be good against quantum storage. However, Gavinsky et. al. [9] show an example of an extractor that works well against classical storage but fails even against much shorter quantum storage.

[*] Department of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Supported by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, by Israel Science Foundation grant 217/05 and by USA Israel BSF grant 2004390. Email: amnon@tau.ac.il.

[1] Extractors are defined in Section 3.

[2] A formal definition is given in Section 3.

To summarize, many techniques and constructions generalize and work well against quantum storage. Yet, in spite of much effort, none of the above methods give a short seed extractor against quantum storage. [14, 15] have seed length $\Omega(n)$ and the variant with almost pair-wise independence has seed length $\Omega(m)$, where n is the length of x and m is the output length. [8] requires the seed length to be $\Omega(b)$ where b is the bound on the quantum storage. [17] show any single output bit extractor is good against quantum storage, and for m bits their method gives $m \log n$ seed length. Alternatively, they show one can do with $O(\log n + b)$ seed length, which is again not applicable if b is relatively large (say, super-polynomial). In contrast, classically, there are many explicit constructions with poly-logarithmic seed length, some even with logarithmic seed length. Some of these constructions are summarized in Table 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 Trevisan's extractor [24] is also good against quantum storage, with somewhat weaker parameters.

Theorem 1.1. *There exists a constant $c > 1$, such that for every $k, b < n$ and $\epsilon > 0$ there exists an explicit (k, b, ϵ) strong extractor $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ against b quantum storage, with seed length $t = O(\frac{\log^2 n}{\log m})$ and output length $m = \Omega(\frac{\epsilon}{\log n}(\frac{k}{b})^{1/c})$.* [3]

Plugging $k = n$ which is the usual setting for privacy amplification, we get:

Corollary 1.1. *For the above constant c, for every $\beta < 1, \gamma < \frac{1-\beta}{c}$ there exists an explicit $(n, b = n^\beta, \epsilon = n^{-\gamma})$ strong extractor $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ against quantum storage, with output length $n^{\Omega(1)}$ and seed length $t = O(\log n)$.*

The seed length is $O(\log n)$ and matches classical extractor's lower bound up to constant multiplicative factors. The error ϵ is not that good, as it can not get below, e.g., $1/k$. The number of extracted bits is $n^{\Omega(1)}$. This should be compared with $n^{1-\zeta}$ for ζ arbitrarily small, in Trevisan's extractor against classical storage. Thus we have a polynomial loss here compared to the original classical scheme.

Table 1 summarizes the parameters of the known classical extractors against quantum storage. Our work gives the first solution to the privacy amplification problem against quantum storage with logarithmic seed length. We believe that other extractor constructions should also be good against quantum storage.

The technique. One way to view Trevisan's extractor is as follows. We already said a random access code is a classical extractor outputting a *single* bit. One can take m *independent* copies of this extractor 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 *pair-wise 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.

Can this also work against quantum storage? Anbainis et al. [2] show a random access code is a single-output extractor against quantum storage. Konig and Terhal [17] show taking m *independent* copies of this extractor is good against quantum storage. What about the derandomized version with pair-wise nearly-disjoint sets? Is it also good against quantum storage?

The analysis of Trevisan's extractor uses the fact that it is built upon a *reconstructible* pseudo-random 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 a short advice to reconstruct its input x. This kind of reasoning looks well suited to generalizations to extractors against quantum storage. Assume Charlie can distinguish the extractor 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 a short advice of a classical bits. Thus, we can 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 $b < n - a$, we output uniform bits.

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

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, [13]	✓[14]
$O(b + \log n)$	$m = n - b - O(1)$	Fourier analysis, collision [6]	✓[8]
$\Theta(m)$	$m \leq n - b - O(1)$	Almost pair-wise ind., [22, 10]	✓, based on [14]
$O(\frac{\log^2 n}{\log(n-b)})$	$(n-b)^{1-\alpha}$	Designs, [24]	✓, This paper. $m \approx \frac{\epsilon}{\log n}(\frac{n-b}{b})^{\Omega(1)}$
$O(\log n)$	$m = \Omega(n-b)$	[18, 11, 7]	?

Table 1: Milestones in building explicit strong extractors against b storage, in the classical and quantum setting. The error ϵ is a constant.

A fundamental problem that arises in the proof is that quantum advice is fragile, and using it once degrades it. This is exactly the main problem dealt with in [1, 19, 2]. Simplifying things, this problem forces the reconstruction algorithm to making only few queries to Charlie. Thus, a key ingredient in our solution is replacing the error correcting codes used in Trevisan's extractor with locally list-decodable codes (see Section 4). Another problem is that the analysis requires random access codes of *subsets*. We explain the technical problems we encounter and their solution (and the way this affects the parameters) in detail in the technical sections.

2 Preliminaries

We begin with some standard notation. A distribution D on Λ is a function $D : \Lambda \rightarrow [0,1]$ such that $\sum_{a \in \Lambda} D(a) = 1$. $x \in D$ denotes sampling according to the distribution D. U_t denotes the uniform distribution over $\{0,1\}^t$. We measure distance between two distributions with the variational distance $d(D_1, D_2) = \frac{1}{2}|D_1 - D_2|_1 = \frac{1}{2}\sum_{a \in \Lambda}|D_1(a) - D_2(a)| = \max_{S \subseteq \Lambda} D_1(S) - D_2(S)$, where $D(S) = \sum_{s \in S} D(s) = \Pr_{a \in D}(a \in S)$.

The entropy of D is $H(D) = \mathrm{E}_{a \in D} \log(1/D(a))$. The min-entropy of D is $H_\infty(D) = \min_{a: D(a) > 0} 1/\log(D(a))$. If $H_\infty(D) \leq k$, then for all a in its support $D(a) \geq 2^{-k}$. A distribution is flat if it is uniformly distributed over its support. For flat distributions $H_\infty(X) = H(X)$. 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.

A superposition is a vector in some Hilbert space. $\mathcal{H}_{2^b}$ denotes a Hilbert space of dimension 2^b. A general quantum system is in a *mixed state*—a probability distribution over superpositions. Let $\{p_i, |\phi_i\rangle\}$ denote the mixed state where superposition $|\phi_i\rangle$ occurs with probability p_i. The behavior of the mixed state $\{p_i, |\phi_i\rangle\}$ is completely characterized by its *density matrix* $\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|$ in the sense that two mixed states with the same density matrix have the same behavior under any physical operation. Notice that a density matrix over a Hilbert space $\mathcal{H}$ belongs to $Hom(\mathcal{H}, \mathcal{H})$, the set of linear transformation from $\mathcal{H}$ to $\mathcal{H}$. Density matrices are positive semi-definite operators and have trace 1.

A POVM (Positive Operator Value Measure) 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 semi-definite operators $E_i : Hom(\mathcal{H}, \mathcal{H}) \rightarrow Hom(\mathcal{H}, \mathcal{H})$ 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 ρ results in answer i with probability $\mathrm{Trace}(E_i \rho)$.

3 Extractors against quantum storage

3.1 Extractors and privacy amplification

Alice holds a string x drawn from the uniform distribution. An adversary C is given some partial information about x in two ways:

- First, C is told a small subset $X \subseteq \{0,1\}^n$ from which the input x is taken.
- Second, we let C keep b bits of information about x.

In the classical world we model the second item by two arbitrarily correlated random variables X and C, with the constraint that C is distributed over $\{0,1\}^b$. In the quantum world, we say an (n,b) quantum encoding is a collection $\{\rho(x)\}_{x \in \{0,1\}^n}$ of density matrices $\rho(x) \in \mathcal{H}_{2^b}$, and we let C hold any (n,b) quantum encoding of X.

Our goal is to find a function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ such that $E(X, U_t)$, which is the distribution obtained by picking $x \in X, y \in U_t$ and outputting $E(x,y)$, "looks uniform" to the adversary C. We define this as follows. We say a boolean test T ϵ–distinguishes D_1 from D_2 if $|\Pr_{x_1 \in D_1}[T(x_1) = 1] - \Pr_{x_2 \in D_2}[T(x_2) = 1]| \geq \epsilon$. We say D_1 is ϵ-indistinguishable from D_2 if no boolean POVM can ϵ distinguish D_1 from D_2. We define:

Definition 3.1. *A function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a (k,b,ϵ) strong extractor against quantum storage, if for any 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 ϵ-indistinguishable from $U_{t+m} \circ \rho(X)$.*[4]

In the definition we could have replaced the condition "for any distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$" with the condition "for any *flat* distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$", as any distribution $X \subseteq \{0,1\}^n$ with $H_\infty(X) \geq k$ can be expressed as a convex combination of flat distributions with min-entropy k.

We similarly define a (k,b,ϵ) strong extractor against *classical* storage, where we allow the adversary C two types of information: first we tell C that x is drawn from a small subset $X \subseteq \{0,1\}^n$, and second, we let C store b bits of information about x. However, classically, these two types of information are redundant. Formally,

Lemma 3.1. *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. Assume C keeps b bits of information. Except for probability ϵ, C gets a value c such that $\Pr[C = c] \geq \epsilon 2^{-b}$ and so $H_\infty(X|C = c) \geq k - b - \log \epsilon^{-1}$ and therefore $(U_t \circ E(X,U_t) \mid C = c)$ is ϵ 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 is that it is not clear how to define the conditional distribution $(X|C = \rho)$ when C may be quantum. Renner [21] defines *smooth min-entropy* for this case, but still it is not clear how to define the marginal distribution itself as it depends on which measurement C chooses to take later.

Another way to look at the problem is as follows. In the classical world, C has to first choose c bits of information about x which already determines a distribution $(X|C = c)$, and only then an independent random seed $y \in \{0,1\}^t$ is chosen and $E(x,y)$ is calculated. In the quantum world, however, things are not that simple. C first chooses c qubits of information about x. This by itself does not determine any classical distribution X on $\{0,1\}^n$. Next, an independent random seed $y \in \{0,1\}^t$ is chosen and $E(x,y)$ is calculated. Finally, C may choose which measurement to make based on x and y. The problem is that it may be possible for C to make a measurement that will correlate the distribution X with the seed y, making the extractor useless. This point of view is further explained in [17].

[4] $U_t \circ E(X,U_t) \circ \rho(X)$ denotes the mixed state obtained by sampling $x \in X, y \in \{0,1\}^t$ and outputting $|y, E(x,y)\rangle \otimes \rho(x)$. Similarly, $U_{t+m} \times \rho(X)$ denotes the mixed state obtained by sampling $w \in \{0,1\}^{t+m}, x \in X$ and outputting $|w\rangle \otimes \rho(x)$.

3.2 Random access codes

A similar problem to the one above appears in *random access codes*. We now explain what random access codes are, as this will turn out to be a basic building block in our result. A fundamental result in quantum information theory, Holevo's theorem [12], 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,

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 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 has a choice of measurement he can make to extract information about their state. In general, these measurements do not commute. Thus making a particular measurement will disturb the system, thereby destroying some or all the information that would have been revealed by another possible measurement. Indeed, Ambainis et. al. [1] ask whether there exists an (n, b) quantum encoding $\{\rho(x)\}$ such that the recipient can learn any bit x_i of his choice. I.e., they define:

Definition 3.2. *[2] A $n \overset{p}{\mapsto} t$ 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 $\mathcal{E}^i = \{\mathcal{E}^i_0, \mathcal{E}^i_1\}$ (i.e., $\mathcal{E}^i_0 + \mathcal{E}^i_1 = I, \mathcal{E}^i_j \succeq 0$) such that for all $x \in \{0, 1\}^n$ we have $\mathrm{Trace}(\mathcal{E}^i_{x_i} f(x)) \geq p$.*

[19, 2] show that any quantum $n \overset{p}{\mapsto} t$ encoding must have $t \geq (1 - H(p))n$. In fact, this lower bound also holds if we relax the worst-case condition $\forall_x\forall_i\mathrm{Trace}(\mathcal{E}^i_{x_i} f(x)) \geq p$ and replace it with the average-case condition $\forall_x\mathbb{E}_i\mathrm{Trace}(\mathcal{E}^i_{x_i} f(x)) \geq p$.

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

Definition 3.3. *Let $\mathcal{F} \subseteq \{0, 1\}^n$. A $\mathcal{F} \overset{p}{\mapsto} t$ 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 $\mathcal{E}^i = \{\mathcal{E}^i_0, \mathcal{E}^i_1\}$ (i.e., $\mathcal{E}^i_0 + \mathcal{E}^i_1 = I, \mathcal{E}^i_j \succeq 0$) such that for all $x \in \mathcal{F}, i \in [n]$ we have $\mathrm{Trace}(\mathcal{E}^i_{x_i} f(x)) \geq p$.*

We prove:

Theorem 3.2. *Let $\delta \geq 0$, $\mathcal{F} \subseteq \{0, 1\}^n$.*

1. *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. *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 $\mathcal{F} \overset{1-\epsilon}{\mapsto} O(t \times T)$ 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 over the T decodings of the T copies. By Chernoff, The probability of error is at most ϵ.

Fix $\epsilon = \frac{c}{n^2}$ for some constant c that will be fixed 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 applying sequentially the measurements $\mathcal{E}^1, \ldots, \mathcal{E}^n$ results in a distribution Y that outputs $(f_1, \ldots, f_n)$ with probability at least $1 - 4n\sqrt{\epsilon} = 1 - 4\sqrt{c}$. Taking c small enough, we recover y with probability $\frac{1}{2}$. By Holevo's theorem, $Tt \geq I(U_{\mathcal{F}} : Y) \geq \frac{1}{2}\log(|\mathcal{F}|)$.

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

[5] Implicit in the proof of Lemma 4.2.

Oded Regev showed us an example where the bound in Theorem 3.2 is tight. Partition the n bits to $\sqrt{n}$ blocks each of size $\sqrt{n}$. Take the set $\mathcal{F}$ to be all bit strings containing exactly one 1 in each block. $\mathcal{F}$ has $\Theta(\sqrt{n} \cdot \log n)$ entropy. Yet, consider the following RAC 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 RAC 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 0. By the pairwise independent property, we output the correct answer with probability $2/3$ for each question.

We proved Theorem 3.2 with the 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, here there is a crucial difference between worst-case and average-case complexity over i.

4 Local list-decoding

A code is a function $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. We identify a binary 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 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 few possible decodings. 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 $ag(z, y) \stackrel{\text{def}}{=} |\{i \in [\bar{n}] | z_i = y_i\}| \geq p\bar{n}$.

As always one can study the combinatorial properties of a code, or ask for an explicit decoding algorithm. If the decoding algorithm makes only few queries to the corrupted word, we say it is *local*. Formally,

Definition 4.1. *(local list-decoding) Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. We say $\mathcal{C}$ has a (p, L, q, β) local list-decoding if:*

- *$\mathcal{C}$ is (p, L) list-decodable.*
- *There exists a probabilistic, polynomial time oracle machine A that on input $k \in [L]$ and $i \in [n]$ outputs a value $A^*(k, i) \in \{0, 1\}$. A can make at most q queries and each query is in the range $[\bar{n}]$.*
- *For every deterministic function $y : \Sigma^{\bar{n}} \to \Sigma$ 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 $i \in [n]$, $\Pr_A[A^y(k, i) = x(i)] \geq \beta$.*

Sudan, Trevisan and Vadhan proved:

Theorem 4.1. *[23] For every $\delta = \delta(n) > 0$, there exists an explicit $[\bar{n}, n]_2$ binary code with output length $\bar{n} = poly(n, \frac{1}{\delta})$ and $poly(\bar{n})$ encoding time, that is $(p = \frac{1}{2} + \delta, L = poly(\bar{n}), q = poly(\log n, \frac{1}{\delta}), \beta = 1 - \delta)$ local list-decodable.*[6]

In our case we do not have access to a deterministic function $y : [\bar{n}] \to \Sigma$, but rather to a probabilistic procedure that has high on average success probability. We are given 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,

Definition 4.2. *(probabilistic oracle, local list-decoding) Let $\mathcal{C} : \Sigma^n \to \Sigma^{\bar{n}}$. We say $\mathcal{C}$ has a (p, L, q, β)* probabilistic oracle*, local list-decoding if:*

[6] The code in [23] is Reed Muller concatenated with Hadamard. The list-decoding algorithm first list-decodes the Hadamard code, and then uses the result to list-decode the Reed Muller code. As the Hadamard list decoding returns a list, it is better to use there list recovery. Working out the parameters we get field size $|F|$ that is $|F| = O(\frac{\log^2 n}{\delta^5})$. With $|F|^3$ queries the algorithm solves the local list-decoding problem, worst-case over i. We remark that using a better inner code the query complexity can be reduced.

- *$\mathcal{C}$ is (p, L) list-decodable.*

- *There exists a probabilistic, polynomial time oracle machine A that on input $k \in [L]$ and $i \in [n]$ outputs a value $A^*(k, i) \in \{0,1\}$. A can make at most q queries and each query is in the range $[\bar{n}]$.*

- *For every probabilistic oracle $O : \Sigma^{\bar{n}} \to \mathcal{D}$ 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 $i \in [n]$, $\Pr_A[A^O(k, i) = x(i)] \geq \beta$.*

If we are just interested 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 Chernoff, 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 [23] one can check that essentially the same analysis shows that:[7]

Theorem 4.2. *(based on [23]) For every $\delta = \delta(n) > 0$, there exists an explicit $[\bar{n}, n]_2$ binary code with output length $\bar{n} = poly(n, \frac{1}{\delta})$ and $poly(\bar{n})$ encoding time, that is $(p = \frac{1}{2} + \delta, L = poly(\bar{n}), q = poly(\log n, \frac{1}{\delta}), \beta = 1 - \delta)$ probabilistic oracle, local list-decodable.*

5 Black-box PRGs

Trevisan showed that good classical black-box PRGs give rise to good classical extractors. In this section we show that good classical black-box PRGs with *few queries* give rise to good classical extractors against *quantum storage*.

We begin with a purely classical definition:

Definition 5.1. *(black-box PRG) Let $G^{f:[n]\to\{0,1\}} : \{0,1\}^t \to \{0,1\}^m$ be a classical oracle machine with oracle calls to a function $f : [n] \to \{0,1\}$. (G^f, R) is a black-box (ϵ, p)-PRG with a advice bits and q queries, if:*

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

- *For every Boolean function $f : [n] \to \{0,1\}$, and every probabilistic oracle T that ϵ–distinguishes $U_t \circ G^f(U_t)$ from uniform, there exists an advice $adv = adv(T, f) \in \{0,1\}^a$ such that for all $i \in [n]$, $\Pr_{R,T}[R^T(adv, i) = f(i)] \geq p$.*

We call R the reconstruction algorithm. *Sometimes we omit R and say G^f is a black-box (ϵ, p)-PRG, meaning that there exists some reconstruction algorithm such that (G^f, R) is a black-box (ϵ, p)-PRG.*

Trevisan [24] showed that black-box pseudorandom generators give rise to extractors. We show they actually give rise to extractors against quantum storage, alas their quality depends on the number of oracle calls in the reconstruction algorithm.

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

Proposition 5.1. *(generalizing [24]) Let G^f, R be as above. Suppose (G^f, R) is a black-box $(\epsilon, p = 1-\delta)$-PRG with a advice bits and q queries. Then $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ defined by $E(f, y) = G^f(y)$ is a $(k, b, 2\epsilon)$ strong extractor against quantum storage, for $k = \Omega(\frac{\log n}{\log(1/4\delta)}(a+qb)) + \log \epsilon^{-1}$.*

Proof. Let T be a quantum test using b qubits of side information ρ. We currently think of T as a probabilistic oracle. Let $\mathcal{F}$ be the set of all functions $f \in \{0,1\}^n$ for which T ϵ-distinguishes $U_t \times E(f, U_t) \times \rho(f)$ from $U_t \times U_m \times \rho(f)$. We will show $|\mathcal{F}| = 2^{O((a+qb)\cdot \log(n)/log(1/4\delta))}$. It will then follow that for any $X \subseteq \{0,1\}^n$, $|\Pr[T(U_t \times E(X, U_t) \times \rho(X)) = 1] - \Pr[T(U_t \times U_m \times \rho(X)) = 1]| \le \mathbb{E}_{x \in X} |\Pr[T(U_t \times E(x, U_t) \times \rho(x)) = 1] - \Pr[T(U_t \times U_m \times \rho(x)) = 1]| \le \epsilon + \Pr_{x \in X}[x \in \mathcal{F}]$. Thus, E is a $(\log \frac{|\mathcal{F}|}{\epsilon}, b, 2\epsilon)$ strong extractor against quantum storage.

We now show $\mathcal{F}$ is indeed small. For any $f \in \mathcal{F}$, given the right advice $adv = adv(T, f) \in \{0,1\}^a$ the 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 $\rho(f)$. Thus, 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 $p = 1 - \delta$. By Theorem 3.2, item (2), $a + qb = \Omega(\frac{\log(1/4\delta)}{\log n} log|\mathcal{F}|)$ as desired. □

Thus, we reduced the problem of finding extractors against quantum storage to the classical question of finding good black-box PRG with *few queries*. In the next section we will prove:

Theorem 5.1. *Let $\epsilon > 0$, $m \le n$. There exists an explicit black-box $(\epsilon, 1 - \frac{\epsilon}{2m})$ 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, seed length $t = O(\frac{\log^2 \frac{n}{\epsilon}}{\log m})$ and $q = \text{poly}(\log n, \frac{m}{\epsilon})$ queries.*

Plugging Thm 5.1 into Proposition 5.1 we get Theorem 1.1.

5.1 A black-box PRG with few queries

Trevisan's PRG [24] is based on the Nisan-Wigderson PRG [20], which has a *good on average* reconstruction algorithm. Formally,

Definition 5.2. *Let G^f, R be as above. (G^f, R) is a black-box (ϵ, p)-PRG with* average-case *reconstruction with a advice bits and q queries, if for every Boolean function $f : [n] \to \{0,1\}$, and every probabilistic oracle T that ϵ–distinguishes $U_t \circ G^f(U_t)$ from uniform, there exists an advice $adv = adv(T, f) \in \{0,1\}^a$ such that $R^T(adv, x)$ makes at most q queries to T and $\Pr[R^T(adv, i) = f(i)] \ge p$, where the probability is over a uniform $i \in [n]$ and the internal coins of R and T.*

The NW PRG is a black-box PRG with average-case reconstruction. Specifically, for every $\epsilon > 0$, $\text{NW}^{f:[n]\to\{0,1\}} : \{0,1\}^t \to \{0,1\}^m$ has $(\epsilon, p = \frac{1}{2} + \frac{\epsilon}{2m})$ average-case reconstruction with $a = O(m^2)$ advice bits and $t = O(\frac{\log^2 n}{\log m})$. The NW reconstruction algorithm uses exactly one oracle call to the distinguishing algorithm. Trevisan used that to prove the following:

Lemma 5.1. *(Trevisan's worst-case to average-case reduction for black-box PRG) Assume (G^f, R) is a black-box $(\epsilon, \frac{1}{2} + \delta)$-PRG with average-case reconstruction using a advice bits. Further assume the reconstruction algorithm R is deterministic. Let $\mathcal{C}[\bar{n}, n]_2$ be a $(\frac{1}{2} + \delta, L)$ list-decodable code. Define $\text{TR}^f(y) = \text{NW}^{\mathcal{C}(f)}(y)$. Then TR^f is a black-box (ϵ, p)-PRG with $a + \log L$ advice bits.*

Proof. Suppose T ϵ–breaks the PRG $\text{TR}^f = \text{NW}^{\mathcal{C}(f)}$. W.l.o.g. we can assume T is deterministic. Let $\bar{f} = \mathcal{C}(f) \in \{0,1\}^{\bar{n}}$. Given the right advice $adv = adv(f, T)$ to R, $R^T(adv, \cdot)$ is a deterministic function computing $\bar{f}_i$ with average success probability p over $i \in [\bar{n}]$, and using only one query to T. The advice to the new reconstruction algorithm R' includes the string adv. R' uses the reconstruction algorithm

$R^T(adv,\cdot)$ on each $j \in [\bar{n}]$. The resulting string $\widehat{y} \in \{0,1\}^{\bar{n}}$ has $(\frac{1}{2}+\delta)\bar{n}$ agreement with $\bar{f}$. We now use the list decoding algorithm to get a list of up to L codewords in $\mathcal{C}$ that are $\frac{1}{2}+\delta$ close to $\widehat{y}$. We know f is the list. By adding $\log(L)$ bits to the advice, we can let the advice tell us which of the codewords in the list is f. We have recovered f using $a+\log L$ advice bits and $\bar{n}$ queries. □

Trevisan could tolerate $\bar{n}$ queries. We, however, in light of Proposition 5.1, need to reduce the number of queries. We still want, however, a *worst-case* reconstruction. The idea is to take $\mathcal{C}$ to be a locally list-decodable code. As our oracle is a probabilistic function what we actually need is a probabilistic oracle, locally list-decodable code. This leads to:

Lemma 5.2. *(worst-case to average-case reduction for black-box PRG using only few queries) Assume (G^f,R) is a black-box $(\epsilon,\frac{1}{2}+\delta)$-PRG with average-case reconstruction using a advice bits. Let $\mathcal{C}$ be a $(p=\frac{1}{2}+\delta,L,q,\beta)$ probabilistic oracle, local list-decodable binary code. Define $\mathrm{TR}^f(y)=\mathrm{NW}^{\mathcal{C}(f)}(y)$. Then TR^f is a black-box (ϵ,β)-PRG with $a+\log L$ advice bits and q queries.*

Proof. Suppose T ϵ–breaks the PRG $\mathrm{TR}^f=\mathrm{NW}^{\mathcal{C}(f)}$. Let $\bar{f}=\mathcal{C}(f)$. 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 $i\in[\bar{n}]$ and a single query to T. The advice to the new reconstruction algorithm R' includes the string adv.

Now assume we ask R' for the value of f_i, $i\in[n]$, i.e., we wish to compute $R'^T(adv,i)$. We do that as follows. We apply the probabilistic oracle, local list-decoding algorithm of $\mathcal{C}$, and get q queries $i_1,\ldots,i_q\in[\bar{n}]$ to $\bar{f}=\mathcal{C}(f)$. We answer the j'th query with the probabilistic oracle $R^T(adv,i_j)$ and we output the decoding result. By the probabilistic oracle, local list-decoding property, for every $i\in[n]$ the reconstruction oracle R'^T, with additionally the right $k\in[L]$, outputs the right answer with probability at least β. □

Putting it together, we prove Theorem 5.1

Proof. Let $\epsilon>0, m\le n$. 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 $t'=O(\frac{\log^2 n}{\log m})$. Nisan and Wigderson showed that 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, local list-decodable binary code of Theorem 4.2. Define $\mathrm{TR}^{f:[n]\to\{0,1\}}:\{0,1\}^{t'}\to\{0,1\}^m$ by $\mathrm{TR}^f(y)=\mathrm{NW}^{\mathcal{C}(f)}(y)$ with $t'=O(\frac{\log^2\frac{n}{\delta}}{\log m})$. By Lemma 5.2 TR^f is a black-box $(\epsilon,1-\delta)$ PRG with $a=O(m^2+\log\frac{n}{\epsilon})$ advice bits and q queries. □

6 Open problems

The ideal solution to the problem of classical extractors against quantum storage, is to find a natural, generic transformation from a strong extractor to a strong extractor against quantum storage with about the same parameters. Gavinsky et. al. [9] showed this is impossible. Is there a natural class of constructions that does hold against quantum storage? Even if not, a natural objective is to prove that many of the current explicit extractors (and in particular [18, 11, 7]) are good even against quantum storage.

The parameters given in Theorem 1.1 can probably be improved. It would be interesting to construct an extractor against quantum storage with logarithmic seed length and arbitrarily small polynomial error, as this may serve as a building block in other constructions.

Acknowledgements

I would like to thank Avi 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 3.2 is tight.

References

[1] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani. Dense quantum coding and quantum finite automata. In *STOC*, pages 376–383, 1999.

[2] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM*, 49(4):496–511, 2002.

[3] C.H. Bennett, G. Brassard, C. Crepeau, and U. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6 Part 2):1915–1923, 1995.

[4] C.H. Bennett, G. Brassard, and J.M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[5] M. Christandl, R. Renner, and A. Ekert. A Generic Security Proof for Quantum Key Distribution. Technical report, Arxiv preprint quant-ph/0402131, 2004.

[6] Y. Dodis and A. Smith. Correcting errors without leaking partial information. *STOC*, pages 654–663, 2005.

[7] Z. Dvir and A. Wigderson. Kakeya sets, new mergers and old extractors. In *FOCS*, page ??, 2008.

[8] S. Fehr and C. Schaffner. Randomness Extraction via Delta-Biased Masking in the Presence of a Quantum Attacker. *Arxiv preprint arXiv:0706.2606*, 2007.

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

[10] O. Goldreich and A. Wigderson. Tiny families of functions with random properties: a quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[11] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes. In *Computational Complexity*, pages 96–108, 2007.

[12] A.S. Holevo. Some estimates of the information transmitted by quantum communication channels. *Problems of Information Transmission*, 9:177–183, 1973.

[13] R. Impagliazzo, L. Levin, and M. Luby. Pseudo-random generation from one-way functions. *STOC*, pages 12–24, 1989.

[14] R. Konig, U. Maurer, and R. Renner. On the Power of Quantum Memory. *Arxiv preprint quant-ph/0305154*, 2003.

[15] R. Konig, U. Maurer, and R. Renner. On the power of quantum memory. *IEEE Transactions on Information Theory*, 51(7):2391–2401, 2005.

[16] R. Konig and R. Renner. Sampling of min-entropy relative to quantum knowledge. *Arxiv preprint arXiv:0712.4291*, 2007.

[17] R. Konig and B. Terhal. The Bounded-Storage Model in the Presence of a Quantum Adversary. *IEEE Transactions on Information Theory*, 54(2):749–762, 2008.

[18] C. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *STOC*, pages 602–611, 2003.

[19] A. Nayak. Optimal lower bounds for quantum automata and random access codes. In *FOCS*, pages 369–376, 1999.

[20] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[21] R. Renner. *Security of Quantum Key Distribution*. PhD thesis, Swiss Federal Institute of Technology (ETH) Zurich, September 2005. available at http://arxiv.org/abs/quant-ph/0512258.

[22] A. Srinivasan and D. Zuckerman. Computing with Very Weak Random Sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[23] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the xor lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[24] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.