Hyper-Encryption against Space-Bounded Adversaries from On-Line Strong Extractors

Chi-Jen Lu[1]

Institute of Information Science, Academia Sinica, Taipei, Taiwan
cjlu@iis.sinica.edu.tw

Abstract. We study the problem of information-theoretically secure encryption in the bounded-storage model introduced by Maurer [10]. The sole assumption of this model is a limited storage bound on an eavesdropper Eve, who is even allowed to be computationally unbounded. Suppose a sender Alice and a receiver Bob agreed on a short private key beforehand, and there is a long public random string accessible by all parties, say broadcast from a satellite or sent by Alice. Eve can only store some partial information of this long random string due to her limited storage. Alice and Bob read the public random string using the shared private key, and produce a one-time pad for encryption or decryption. In this setting, Aumann, Ding, and Rabin [2] proposed protocols with a nice property called everlasting security, which says that the security holds even if Eve later manages to obtain that private key. Ding and Rabin [5] gave a better analysis showing that the same private key can be securely reused for an exponential number of times, against some adaptive attacks.

We study this problem from the approach of constructing randomness extractors ([13, 11, 16, 15] and more), which seems to provide a more intuitive understanding together with some powerful tools. A strong extractor is a function which purifies randomness from a slightly random source using a short random seed as a catalyst, so that its output and its seed together look almost random. We show that any strong extractor immediately yields an encryption scheme with the nice security properties of [2, 5]. To have an efficient encryption scheme, we need strong extractors which can be evaluated in an on-line and efficient way. We give one such construction. This yields an encryption scheme, which has the same nice security properties as before but now can encrypt longer messages using a shorter private key. In addition, our scheme works even when the long public random string is not perfectly random, as long as it contains enough amount of randomness.

1 Introduction

Almost all cryptographic protocols in use today are based on some intractability assumptions. That is, adversaries are assumed to be computationally bounded, and some problems are assumed to be computationally hard. However, no such complexity lower bound has been proved, and in fact it seems to remain far beyond the reach of current techniques in complexity theory. So it is possible that future advances in cryptanalysis or computer technology may jeopardize the security of today's cryptographic systems. One extreme then is to look for protocols with provable information-theoretical security, which is against an computationally-unbounded adversary. However, a well-known pessimistic result of Shannon says that no interesting protocol can be expected. Taking a step back, it would still be nice if one could prove information-theoretical security basing only on some minimal and reasonable assumption. Maurer proposed one such model, called the bounded-storage model [10], where the only assumption is that an adversary has a bounded amount of storage. To exploit such weakness of an adversary, a very long public random string, accessible by all parties, is usually employed. With respect to this model, several interesting cryptographic protocols have been proposed, with provable information-theoretical security.

One important task in cryptography is secure transmission against eavesdropping, where a sender Alice wants to send a message to a receiver Bob in a way to keep an eavesdropper Eve from learning the content. Any public-key encryption scheme unavoidably must be based on some computational lower bound, as an adversary with unbounded computation power certainly can invert the publicly-known encryption procedure. To achieve information-theoretical security, we have to use private-key encryption. In this paper, we study private-key encryption using one-time pads in the bounded-storage model. Assume Alice and Bob share a private key beforehand, say sent from Alice to Bob via today's public-key encryption. Then a long public random string X of length n is generated, say broadcast from a satellite or sent by Alice, which is accessible by all parties. Eve only has a storage of νn bits, for some constant $\nu < 1$, so she can only store some partial information about X. To be reasonable, the same storage bound is also imposed on Alice and Bob. Alice and Bob read the string X on the fly, and compute a one-time pad Z. Then Alice encrypts her message as $C = M \oplus Z$ and sends C to Bob. When X is sent, Eve computes and stores some νn bits of information about X, hoping later to recover M after eavesdropping the cipher-text C. In this setting, Aumann, Ding, and Rabin [2] gave protocols, improving those of Maurer [10] and Cachin and Maurer [4], which enjoy a nice provable property called *everlasting security.* This is an information-theoretical security that will last even if Eve later after the transmission manages to obtain that private key, say by breaking that public-key encryption. Such a feature seems quite attractive, as the security is guaranteed by the limitation of current storage technology, and will not be affected by future advances of any kind. It is possible as some crucial information has been lost forever. Shortly after, Ding and Rabin [5] gave a better analysis showing that the same private key in [2] can be securely reused for an exponential number of times in a way that each encryption remains secure even after revealing all previous plain-texts.

Observe that as Eve can only store νn bits of information Q from the n-bit random string X, very likely a substantial amount of randomness still remains in X relative to Q. The remaining randomness may be crude and not directly applicable, so one would like to have it purified. This is exactly the issue addressed in the research on constructing the so-called *extractors*, first explicitly defined by Nisan and Zuckerman [13]. Extractors turn out to have many important applications (see [11] for a survey) and have become a subject of intense study, with a break-through by Trevisan [16]. An extractor is a function that given any source with enough randomness uses a short perfect random seed to extract many bits which are close to random. We can use the output from an extractor as a one-time pad for encryption as it looks random to Eve. A strong extractor is an extractor with a stronger guarantee that its output and its seed together are close to random, which gives the following nice properties. On one hand, with high probability, the output of the extractor still looks random even given the value of the seed, which guarantees the property of everlasting security of [2] discussed above. On the other hand, with high probability, the seed still looks random even given the output value of the extractor, which implies that the same private key can be reused again for the next encryption even if Eve knows the pads of previous encryptions. Formally, we show that any strong extractor immediately yields an encryption scheme with everlasting security, against an exponential number of adaptive attacks, the same property enjoyed by protocols of Aumann, Ding, and Rabin [2, 5].

However, for practical consideration, not every strong extractor is suitable for an encryption scheme in the setting we consider here. As the public random string is broadcast at a very high rate and each party does not have enough memory to store the whole string, we need a strong extractor that can be evaluated in an on-line and very efficient way. Existing extractor constructions fail to meet this criterion. One way of constructing extractors, introduced by Trevisan [16], is to encode the input string using some list-decodable code and then project that codeword onto some dimensions determined by the random seed as output. A list-decodable code, roughly speaking, is a code guaranteeing some upper bound on the number of codewords in any Hamming ball of certain radius. The work of Aumann, Ding, and Rabin [2, 5] can also be understood within this framework. Suppose the security parameter is k, so that Eve can only distinguish different messages with an advantage 2^{-k}. One of the main technical work in [2, 5] can be seen as constructing a list-decodable code mapping from an n-bit string to an $n^{O(k)}$-bit codeword, where each bit of a codeword is the parity of $O(k)$ input bits. For an extractor with one output bit, they encode the input, pick a random dimension of that codeword, and output the bit there, needing a seed of length $O(k \log n)$ for sampling. To output m bits, they independently choose m random dimensions, and project that codeword onto those m dimensions, needing a seed of length $O(mk \log n)$. So in their encryption scheme, a private key of length $O(mk \log n)$ is required to encode a message of length m. That is, in order to achieve the everlasting security, they need a private key much longer than the message, which may limit the applicability of their protocol. Our improvement comes from two directions. First, we construct a list-decodable code mapping an n-bit string onto a codeword of length $n2^{O(k)}$, where each bit of a codeword is again the parity of $O(k)$ input bits. To have an m-bit output, if we simply project the codeword onto m random dimensions, needing a seed of length $O(m(k+\log n))$, we already have improvement over that of [2, 5]. For large m, we follow the approach of [16, 15] by picking some pseudo-random collection of m dimensions, instead of m random dimensions, for projection. As a result, for any $m \le n^\gamma$ with $\gamma \in (0, 1)$, we only need a seed of length $O((\log n + k)^2 / \log n)$. Note that n is typically much larger than k. If we assume or choose $n = 2^{\Omega(k)}$, we get an encryption scheme using a private key of length $O(\log n)$, which is a dramatic improvement over [2, 5].

To encrypt an m-bit message, Alice and Bob need to prepare m families of $O(k)$ indices and remember them. We use random walk on expander graphs and some set system to determine those m families, which seems to cost some extra computation compared to [2, 5]. However, this small extra effort is only required during the preprocessing phase, when computation time may not be a concern, so hopefully would not be an issue from a practical point of view. During the broadcast of the long public random string X, what we do is exactly the same as that of [2, 5]. That is, Alice and Bob compute the m-bit one-time pad, where each of the m bits is a parity of $O(k)$ bits from X, which can be done in an extremely fast way.

Our protocol enjoys the following key expansion property, which is addressed by a new result of Dziembowski and Maurer [6]. In [2], to generate a one-time pad of length m, Alice and Bob need to first agree on a private key of length $O(mk \log n)$, much longer than the pad they want to generate. In [5], a key doubling technique is used to generate a one-time pad of length m using a private key of length $O(k \log n)$, but it requires a public random string of length $\Theta(nk \log m \log n)$, much longer than Eve's storage bound νn. So an open question was whether one could generate a pad longer than the private key while using a public random string only slightly longer than Eve's storage bound. The main contribution of Dziembowski and Maurer [6] is to settle this question, and they need a private key of length $O(k \log n)$. Independently[1], we also settles this question but via a very different method, needing a private key of length $O((k + \log n)^2 / \log n)$, which is smaller than theirs for $k = O(\log^2 n)$. Both [6] and our result allow the ratio between the length of public random string and Eve's storage bound to be arbitrarily close to one. On the other hand, we do not require the long public random string X to be perfectly random, while this is not clear in [6]. In reality, perfect random sources may not be available and one may have to rely on sources of lower quality. Our scheme works as long as X contains enough amount of randomness, which only needs to be slightly larger than Eve's storage bound.

In addition to obtaining a better construction, we try to place this line of research in an appropriate framework, where ideas could be understood in a more intuitive and perhaps deeper way and various powerful tools have been developed. Most of the techniques we use are standard in the research area on pseudo-randomness. The idea of using extractors in such a setting actually appeared before in the work of Cachin and Maurer [4]. Since then, the theory

[1] We were not aware of their result when we obtained ours.

of pseudo-randomness has made some advancement, which enables us to have a better understanding and derive stronger results.

Notation, definitions, and some simple facts are given in Section 2. In Section 3, we prove that any strong extractor gives an encryption scheme with everlasting security. In Section 4, we construct an efficient on-line strong extractor which yields an efficient encryption scheme.

2 Preliminaries

For a positive integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. For an n-dimensional vector m, let $m(i)$, for $i \in [n]$, denote the component in the i-th dimension of m, and let $m(I)$, for $I \subseteq [n]$, denote the vector consisting of those $m(i)$'s for $i \in I$. For our convenience, we use $\{-1, 1\}$, instead of the usual $\{0, 1\}$, for the binary values unless noted otherwise. For a positive integer n, let U^n denote the uniform distribution over $\{-1, 1\}^n$, and we omit the superscript n when no confusion is caused. All the logarithms in this paper have base 2.

The distance between two vectors x, y is defined as $d_H(x, y) \equiv |\{i : x(i) \neq y(i)\}|$, which is also known as their *Hamming* distance.

Definition 1. *A mapping $C : \{-1, 1\}^n \to \{-1, 1\}^{\bar{n}}$ defines a code, with $\{C(x) : x \in \{-1, 1\}^n\}$ as the set of codewords. It is called a list-decodable code with parameter $(\frac{1}{2} - \varepsilon, L)$, or a $(\frac{1}{2} - \varepsilon, L)$-list code, if for any $z \in \{-1, 1\}^{\bar{n}}$, there are at most L codewords within distance $(\frac{1}{2} - \varepsilon)\bar{n}$ from z.*

We also need a way to measure how close two distributions are.

Definition 2. *The distance between two distributions A, B is defined as $\|A - B\| \equiv \frac{1}{2} \sum_x |\Pr[A = x] - \Pr[B = x]|$.*

This is usually called the variational distance, and it is easy to verify that $\|A - B\| \leq 1$. We say that a distribution is ε-random if its distance to the uniform one is at most ε. Here is a simple but useful lemma, which is proved in Appendix A.

Lemma 1. *Suppose the distribution A is independent of distributions B and B'. Then for any function g, $\|\langle g(A, B), B\rangle - \langle g(A, B'), B'\rangle\| = \|B - B'\|$.*

We need a measure to quantify the randomness contained in a slightly random source. Shannon's entropy function captures the average randomness of a source, while we need some worst-case measure of randomness instead.

Definition 3. *For a distribution X, its* min-entropy *is defined as $H_\infty(X) \equiv \min_x \log \frac{1}{\Pr[X=x]}$.*

So $H_\infty(X) \geq r$ iff $\Pr[X = x] \leq 2^{-r}$ for any x. For a distribution with enough randomness, guaranteed by its min-entropy, we look for a procedure that can purify the randomness by using a short random seed as a catalyst.

Definition 4. *A function* $\text{EXT} : \{-1,1\}^n \times \{-1,1\}^s \to \{-1,1\}^m$ *is called a strong* (r,ε)*-extractor if for any distribution* X *over* $\{-1,1\}^n$ *with* $H_\infty(X) \geq r$ *and for* $Y = U^s$*, the distribution of* $\langle \text{EXT}(X,Y), Y\rangle$ *is* ε*-random.*

The usual definition of extractors only requires the distribution of $\text{EXT}(X,Y)$ being ε-random. The stronger requirement of a strong extractor EXT gives the following nice property. With high probability, the distribution of $\text{EXT}(X,Y)$ still looks random even given the value of Y, and also with high probability, the distribution of Y still looks random even given the value of $\text{EXT}(X,Y)$. This is guaranteed by the following lemma, which is proved in Appendix B.

Lemma 2. *Suppose the distribution* $\langle A, B\rangle$ *is* ε*-random. Then the probability over* $b \in B$ *that the distribution* $\langle A \mid B = b\rangle$ *is not* $\sqrt{\varepsilon}$*-random is at most* $2\sqrt{\varepsilon}$*.*

We consider private-key encryption schemes using one-time pads. In such a scheme, a sender Alice and a receiver Bob first agreed on a private key Y and then later generate a one-time pad Z_Y so that the message M is encrypted as $C = M \oplus Z_Y$, where $\oplus$ denote the bit-wise exclusive or operation. Note that an eavesdropper Eve with C knows M iff she knows Z_Y. So for the security of such a scheme, it suffices to show that Z_Y looks random to Eve. An even stronger notion of security is the following, introduced by Aumann, Ding, and Rabin [2].

Definition 5. *A private-key encryption scheme is said to have everlasting security of degree* k*, if with probability at least* $1 - 2^{\Omega(k)}$*, the distribution of* Z_Y *remains* 2^{-k}*-random conditioned on Eve's information and* Y*.*

This guarantees that even if Eve is given that private key Y afterwards, she is still unlikely to know anything about the message M.

3 Everlasting Security from Strong Extractors

In this section, we show that using a strong extractor for encryption guarantees the everlasting security of [2, 5]. Recall the following scenario. Before the transmission of an m-bit message M, a sender Alice and a receiver Bob agree on a random private key Y. Then they read the n-bit public random string X, compute the one-time pad Z, and the encrypted message $C = M \oplus Z$ is sent from Alice to Bob. An adversary Eve uses a function f to store νn bits of information $Q = f(X)$ from the public string X, and then eavesdrops the encrypted message C. If later Eve is given that private key Y, can she obtain any information about M?

For the security parameter k, which is usually much smaller than n, choose $\varepsilon = 2^{-2k}$. Pick $\mu < 1-\nu$, say $\mu = 0.99-\nu$, so that $2^{-(1-\mu-\nu)n} \leq \frac{\varepsilon}{2}$, and let $r = \mu n$. We will use a strong $(r, \frac{\varepsilon}{2})$-extractor $\text{EXT} : \{-1,1\}^n \times \{-1,1\}^s \to \{-1,1\}^m$. The private key Y is selected randomly according to the distribution U^s and the one-time pad is generated as $Z = \text{EXT}(X,Y)$. For $M = C \oplus Z$, Eve knows M iff she knows Z, so for the security, it suffices to show that Z looks random to Eve even given Y. The idea is that X conditioned on Q is likely to contain enough randomness, so for a strong extractor $\textsc{Ext}$, very likely the distribution of $Z = \textsc{Ext}(X, Y)$ would still look random even given the value of Y. Here is the key lemma.

Lemma 3. $\|\langle f(X), Y, \textsc{Ext}(X, Y)\rangle - \langle f(X), U^s, U^m\rangle\| \leq \varepsilon$.

Proof. First, note that $\|\langle f(X), Y, \textsc{Ext}(X, Y)\rangle - \langle f(X), U^m, U^s\rangle\|$ equals

$$\sum_q \Pr[f(X) = q] \cdot \|\langle Y, \textsc{Ext}(X, Y) \mid f(X) = q\rangle - \langle U^m, U^s \mid f(X) = q\rangle\|.$$

We call a value $q \in \{-1, 1\}^{\nu n}$ *bad* if $\Pr[f(X) = q] \leq 2^{-(1-\mu)n}$, and let B denote this set of bad q's. For $q \notin B$, $\Pr[X = x \mid f(X) = q] \leq 2^{-n}/2^{-(1-\mu)n} = 2^{-\mu n}$ for any x, so $H_\infty(X \mid f(X) = q) \geq \mu n$. Then for $q \notin B$,

$$\begin{aligned}
&\|\langle Y, \textsc{Ext}(X, Y) \mid f(X) = q\rangle - \langle U^s, U^m \mid f(X) = q\rangle\| \\
&= \|\langle Y, \textsc{Ext}(\langle X \mid f(X) = q\rangle, Y)\rangle - \langle U^s, U^m\rangle\| \\
&\leq \frac{\varepsilon}{2}.
\end{aligned}$$

For $q \in B$, we only have $\|\langle \textsc{Ext}(X, Y), Y \mid f(X) = q\rangle - \langle U^m, U^s \mid f(X) = q\rangle\| \leq 1$, but fortunately it is unlikely to have a bad q as

$$\begin{aligned}
\sum_{q \in B} \Pr[f(X) = q] &\leq \sum_{q \in B} 2^{-(1-\mu)n} \\
&\leq 2^{(\nu - 1 + \mu)n} \\
&\leq \frac{\varepsilon}{2}.
\end{aligned}$$

Thus,

$$\begin{aligned}
&\|\langle f(X), Y, \textsc{Ext}(X, Y)\rangle - \langle f(X), U^s, U^m\rangle\| \\
&= \sum_q \Pr[f(X) = q] \cdot \|\langle Y, \textsc{Ext}(X, Y) \mid f(X) = q\rangle - \langle U^s, U^m \mid f(X) = q\rangle\| \\
&\leq \sum_{q \in B} \Pr[f(X) = q] \cdot 1 + \sum_{q \notin B} \Pr[f(X) = q] \cdot \frac{\varepsilon}{2} \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\
&= \varepsilon.
\end{aligned}$$

□

Now according to Lemma 2, even if Eve saved the information $f(X)$ and later obtains the private key Y, the distribution of $\textsc{Ext}(X, Y)$ is still 2^{-k}-random with probability at least $1 - 2^{-k+1}$. That is, it gives an encryption scheme with everlasting security of degree k.

Just like previous work, we assume above that the public string X is perfectly random. However, such a perfect random source may not be available in reality, and one may need to rely on sources of lower quality. Note that our proof above actually works for any source X of length $n' > n$ with $H_\infty(X) \geq n$. That is, our result holds as long as Eve's storage bound is at most a fraction ν of the source's min-entropy.

3.1 Reusing the Same Private Key

Next, we show that the same key can be reused an exponential number of times, even given all previous one-time pads. Now the idea is that for a strong extractor Ext, very likely the distribution of Y would still look random even given the value of $\textsc{Ext}(X, Y)$, so the same Y could be used again to extract randomness in the next round.

Consider the following scenario, with K messages $M_1, M_2, \ldots, M_K$ to be transmitted. For $i \in [K]$, Alice reads the i-th block of n-bit public random string X_i, computes the i-th pad $Z_i = \textsc{Ext}(X_i, Y)$, and sends the encrypted message as $C_i = M_i \oplus Z_i$ to Bob. For $i \in [K]$, suppose Eve is also given $Z_{[i-1]} = (Z_1, \ldots, Z_{i-1})$ and uses a function f_i to store νn bits of information $Q_i = f_i(X_i, Q_{i-1}, Z_{[i-1]})$ from the public string X_i. Finally, Eve eavesdrops the encrypted message C_K and is given Y, can she learn anything about M_K? For the security, again it suffices to show that Z_K looks random even given $Z_{[K-1]}$, Q_K, and Y. We choose $\varepsilon = 2^{-3k}$, $r = \mu n$, and use a strong $(r, \frac{\varepsilon}{2})$-extractor Ext. Here is the key lemma.

Lemma 4. *For any $i \in \mathbb{N}$, $\|\langle Z_{[i-1]}, Q_i, Y, Z_i\rangle - \langle Z_{[i-1]}, Q_i, U^s, U^m\rangle\| \leq i\varepsilon$.*

Proof. We use induction on i. Lemma 3 handles precisely the case $i = 1$. So assume it holds for some $i \geq 1$ and let's consider the case $i + 1$. By triangle inequality, we have

$$\begin{aligned}
&\|\langle Z_{[i]}, Q_{i+1}, Y, Z_{i+1}\rangle - \langle Z_{[i]}, Q_{i+1}, U^s, U^m\rangle\| \\
\leq\ &\|\langle Z_{[i]}, Q_{i+1}, Y, \textsc{Ext}(X_{i+1}, Y)\rangle - \langle Z_{[i]}, Q_{i+1}, Y', \textsc{Ext}(X_{i+1}, Y')\rangle\| + \\
&\|\langle Z_{[i]}, Q_{i+1}, Y', \textsc{Ext}(X_{i+1}, Y')\rangle - \langle Z_{[i]}, Q_{i+1}, U^s, U^m\rangle\|,
\end{aligned}$$

where Y' is the distribution U^s which is independent of $Z_{[i]}$. The first term is about how random Y remains after i iterations, while the second term is about how good the $(i+1)$-th extraction is using a new random key Y'. In the first term, $Q_{i+1} = f_{i+1}(X_{i+1}, Q_i, Z_{[i]})$, and X_{i+1} is independent of Q_i, $Z_{[i]}$, Y, and Y'. Then according to Lemma 1, the first term equals

$$\begin{aligned}
\|\langle Z_{[i]}, Q_i, Y\rangle - \langle Z_{[i]}, Q_i, Y'\rangle\| &= \|\langle Z_{[i]}, Q_i, Y\rangle - \langle Z_{[i]}, Q_i, U^s\rangle\| \\
&= \|\langle Z_{[i-1]}, Q_i, Y, Z_i\rangle - \langle Z_{[i-1]}, Q_i, U^s, Z_i\rangle\| \\
&\leq i\varepsilon,
\end{aligned}$$

where the inequality is from the inductive hypothesis. In the second term, X_{i+1} is independent of $Z_{[i]}$, so only Q_{i+1} affects the distribution of X_{i+1}. Then using an argument similar to the proof of Lemma 3, the second term can be bounded above by ε. Therefore, we have

$$\|\langle Z_{[i]}, Q_{i+1}, Y, Z_{i+1}\rangle - \langle Z_{[i]}, Q_{i+1}, U^s, U^m\rangle\| \leq (i+1)\varepsilon,$$

which proves the inductive step and thus the lemma. □

For $K \leq 2^k$, the distance is at most 2^{-2k}. Then according to Lemma 2, even given the previous pads $Z_1, \ldots, Z_{K-1}$ and the private key Y, the K-th pad Z_K is still 2^{-k}-random with probability at least $1 - 2^{-k+1}$. So we have the following theorem.

Theorem 1. *Any strong $(\mu n, 2^{-3k})$-extractor yields an encryption scheme with everlasting security of degree k, whose private key can be securely reused for 2^k times, against the adaptive attacks discussed above.*

4 An On-Line Strong Extractor

Although using any strong extractor for encryption does provide a good security guarantee, not any arbitrary strong extractor is suitable in the setting we consider here. Recall that the public random string X is very long so it needs to be broadcast at a very high rate. Alice and Bob do not have enough memory to store the whole string, so they would like to be able to apply the extractor in an on-line and efficient way. One way of constructing extractors, introduced by Trevisan [16], is based on list-decodable codes. In order to have an efficient on-line strong extractor, we need a list-decodable code with an efficient on-line encoding procedure. The main work of [2, 5] can be seen as constructing one such code, where each output bit of a codeword is a parity of a small number of input bits. Here we give another code with a better parameter.

4.1 An On-Line List-Decoding Code

We will use some type of graphs called expander graphs which have their second largest eigenvalue bounded away from their largest one.[2]

Lemma 5. *[7, 8] There exists an explicit family of expander graphs $(G_n)_{n\in\mathbb{N}}$ with the following property. There is a constant d and a constant $\lambda < d$ such that for every $n \in \mathbb{N}$, G_n is a d-regular graph of n vertices with the second largest eigenvalue λ.*

Expander graphs enjoys some pseudo-random properties, and we will use the following one.

Lemma 6. *[1] Suppose G is a d-regular graph with the second largest eigenvalue λ, and B is a set containing at least α fraction of G's vertices. Then a t-step random walk on G misses B with probability at most $\beta_\alpha \equiv (1 - \alpha + (\frac{\lambda}{d})^2)^{t/2}$.*

[2] The eigenvalues of a graph's adjacency matrix are called the eigenvalues of that graph.

In fact, this probability is not far off away from the probability $(1-\alpha)^t$ achieved by sampling t vertices randomly and independently, but now we only need $\log n + t \log d = \log n + O(t)$ random bits for sampling, instead of $t \log n$ random bits. Observe that smaller $\frac{\lambda}{d}$ gives smaller β_α. This ratio could be reduced to $(\frac{\lambda}{d})^i$ by considering the graph G^i, but then each step of a walk on G^i would need $i \log d$ bits, instead of $\log d$ bits, to sample.

Let G denote one such d-regular expander graph on n vertices. For a vertex v and for $w \in [d]^t$, let $W_G(v, w)$ denote the sequence of t vertices visited by a t-step walk on G starting from v and then following the directions $w(1), \ldots, w(t)$. Consider the encoding $\textsc{Ecc}_t : \{-1,1\}^n \to \{-1,1\}^{\bar{n}}$, for $\bar{n} = nd^t 2^t$, defined in Figure 1.

- **Input:** $x \in \{-1,1\}^n$.
- **Output index:** $(v, w, b) \in [n] \times [d]^t \times \{0,1\}^t$.
- **Algorithm:**
  - Do a t-step walk on G to get $W_G(v, w) = (v_1, \ldots, v_t)$.
  - Output the bit indexed by (v, w, b) as

$$\textsc{Ecc}_t(x)(v, w, b) \equiv \prod_{i \in [t]} x(v_i)^{b(i)}.$$

Fig. 1. The code $\textsc{Ecc}_t$

Note that the bit $\textsc{Ecc}_t(x)(v, w, b) = \prod_{i \in [t]} x(v_i)^{b(i)}$ is just the parity of at most t input bits, those bits $x(j)$'s with $j = v_i$ and $b(i) = 1$. Suppose one had (v, w, b) and derived those indices j beforehand. Then when x is given, the bit $\textsc{Ecc}_t(x)(v, r, b)$ can be computed in an on-line and extremely fast way.

To show that this is a good list-decodable code, we would like to bound the number of codewords within any Hamming ball of some radius. The Johnson bound in coding theory (e.g. Chapter 17 of [9]) provides one such bound for codes with some minimum distance guarantee between any two codewords. It cannot be applied directly to our code $\textsc{Ecc}_t$ as some codewords are in fact close to each other. However, we do have some distance guarantee between some codewords, as shown by the following lemma.

Lemma 7. *For any $x, y \in \{-1,1\}^n$ with $d_H(x, y) \geq \alpha n$, $d_H(\textsc{Ecc}_t(x), \textsc{Ecc}_t(y)) \geq (\frac{1}{2} - \beta_\alpha)\bar{n}$.*

Proof. Let $B = \{v \in [n] : x(v) \neq y(v)\}$, which has at least αn elements. Consider any walk $(v_1, \ldots, v_t) \in [n]^t$ that hits B, say at v_{i_0} with $x(v_{i_0}) \neq y(v_{i_0})$. For any assignment to b, flipping the bit $b(i_0)$ flips exactly one value of $x(v_{i_0})^{b(i_0)}$ and $y(v_{i_0})^{b(i_0)}$, and thus exactly one value of $\prod_{i \in [t]} x(v_i)^{b(i)}$ and $\prod_{i \in [t]} y(v_i)^{b(i)}$. Then for such a walk that hits B,

$$\Pr_{b\in\{0,1\}^t}\left[\prod_{i\in[t]} x(v_i)^{b(i)} = \prod_{i\in[t]} y(v_i)^{b(i)}\right] = \frac{1}{2}.$$

According to Lemma 6, a t-step random walk on G misses B with probability at most β_α. So,

$$\begin{aligned}
&\Pr_{v,w,b}[\textsc{Ecc}_t(x)(v,r,b) = \textsc{Ecc}_t(y)(v,r,b)]\\
&\le \Pr_{v,w,b}[W_G(v,w) \text{ misses } B] +\\
&\quad \Pr_{v,w,b}\left[\prod_{i\in[t]} x(v_i)^{b(i)} = \prod_{i\in[t]} y(v_i)^{b(i)} \mid W_G(v,w) \text{ hits } B\right]\\
&\le \beta_\alpha + \frac{1}{2},
\end{aligned}$$

where $W_G(v,w) = (v_1,\ldots,v_t)$ in the first inequality. Then,

$$\begin{aligned}
d_H(\textsc{Ecc}_t(x), \textsc{Ecc}_t(y)) &= \left(1 - \Pr_{v,w,b}[\textsc{Ecc}_t(x)(v,r,b) = \textsc{Ecc}_t(y)(v,r,b)]\right)\bar{n}\\
&\ge \left(\frac{1}{2} - \beta_\alpha\right)\bar{n}.
\end{aligned}$$

□

Now for each codeword $\textsc{Ecc}_t(x)$, the only possible codewords with distance less than $(\frac{1}{2}-\beta_\alpha)\bar{n}$ from $\textsc{Ecc}_t(x)$ are those $\textsc{Ecc}_t(x')$'s with $d_H(x,x') \le \alpha n$, and there are at most $2^{h(\alpha)n}$ such codewords, where $h(\alpha) = \alpha\log\frac{1}{\alpha} + (1-\alpha)\log\frac{1}{1-\alpha}$ is the binary entropy function. Choose $\alpha = 1 + (\frac{\lambda}{d})^2 - \delta^{4/t}$, so $\beta_\alpha \le \delta^2$. Then the following lemma implies that our code $\textsc{Ecc}_t$ is a $(\frac{1}{2}-\delta, \frac{2^{h(\alpha)n}}{2\delta^2})$-list code. It can be seen as a generalization to the Johnson bound, and our proof generalizes the version given in Appendix A of [3].

Lemma 8. *Suppose* $\textsc{Ecc} \subseteq \{-1,1\}^{\bar{n}}$ *is a code such that for any codeword* $c \in \textsc{Ecc}$*, there are at most* M *codewords in* $\textsc{Ecc}$ *within distance* $(\frac{1}{2}-\beta)\bar{n}$ *from* c*. Then for any* $z \in \{-1,1\}^{\bar{n}}$ *and any* $\delta > \sqrt{\beta/2}$*, the number of codewords within distance* $(\frac{1}{2}-\delta)\bar{n}$ *from* z *is at most* $M/(4\delta^2 - 2\beta)$*.*

Proof. For any two vectors $u,v \in \{-1,1\}^{\bar{n}}$, let $u \odot v$ denote their inner product, and note that

$$u \odot v \equiv \sum_{j\in[\bar{n}]} u(j)v(j) = |\{j : u(j) = v(j)\}| - |\{j : u(j) \neq v(j)\}|.$$

Suppose $c_1 = \textsc{Ecc}_t(x_1),\ldots,c_L = \textsc{Ecc}_t(x_L) \in \{-1,1\}^{\bar{n}}$ are those codewords within distance $(1/2-\delta)\bar{n}$ from z. For $i \in [L]$, let $u_i \in \{-1,1\}^{\bar{n}}$ represent the discrepancy between c_i and z, with $u_i(j) = c_i(j)z(j)$ for $j \in [\bar{n}]$.[3] Define

$$T \equiv \left(\sum_{i\in[L]} u_i\right) \odot \left(\sum_{i\in[L]} u_i\right).$$

On one hand,

$$\begin{aligned} T &= \sum_{j\in[\bar{n}]} \left(\sum_{i\in[L]} u_i(j)\right)^2 \\ &\geq \frac{1}{\bar{n}} \left(\sum_{j\in[\bar{n}]} \sum_{i\in[L]} u_i(j)\right)^2, \end{aligned}$$

from Cauchy-Schwartz inequality. As $\sum_{j\in[\bar{n}]} u_i(j) = c_i \odot z \geq 2\delta\bar{n}$ for any $i \in [L]$, we have

$$\begin{aligned} T &\geq \frac{1}{\bar{n}} \left(\sum_{i\in[L]} \sum_{j\in[\bar{n}]} u_i(j)\right)^2 \\ &\geq \frac{1}{\bar{n}} (2\delta\bar{n}L)^2 \\ &= 4\delta^2\bar{n}L^2. \end{aligned}$$

On the other hand, we can write

$$T = \sum_{i\in[L]} \sum_{i'\in[L]} u_i \odot u_{i'}.$$

For $i \in [L]$, let $N(i) = \{i' \in [L] : d_H(x_i, x_{i'}) \leq \alpha n\}$. For $i' \notin N(i)$, we have $u_i \odot u_{i'} \leq 2\beta\bar{n}$. For $i' \in N(i)$, we only have $u_i \odot u_{i'} \leq \bar{n}$, but $|N(i)| \leq M$. So

$$\begin{aligned} T &= \sum_{i\in[L]} \sum_{i'\in N(i)} u_i \odot u_{i'} + \sum_{i\in[L]} \sum_{i'\notin N(i)} u_i \odot u_{i'} \\ &\leq \bar{n}ML + 2\beta\bar{n}L^2, \end{aligned}$$

Combining the two inequalities, we have $4\delta^2\bar{n}L^2 \leq T \leq \bar{n}ML + 2\beta\bar{n}L^2$, which implies $L \leq M/(4\delta^2 - 2\beta)$. □

4.2 Expanding the Output

Given any list-decodable code $\text{ECC}' : \{-1,1\}^n \to \{-1,1\}^{\bar{n}}$, it is known according to [16, 15] that one immediately gets a strong extractor $\text{EXT}' : \{-1,1\}^n \times [\bar{n}] \to \{-1,1\}$ defined as

$$\text{EXT}'(x, y) = \text{ECC}'(x)(y).$$

[3] In [3], $u_i(j)$ is defined as 1 if $c_i(j) = z(j)$ and 0 otherwise. Our definition makes the proof slightly cleaner.

That is, the extractor encodes the input string x as $\text{Ecc}'(x)$ and projects it onto a random dimension y. To obtain m output bits, the approach of [2, 5] is to project $\text{Ecc}'(x)$ onto m independent and random dimensions, needing $m \log \bar{n}$ random bits for sampling. That is, they use the extractor $\text{Ext}'' : \{-1,1\}^n \times [\bar{n}]^m \to \{-1,1\}^m$ defined as

$$\text{Ext}''(x, (y_1, \ldots, y_m)) = \left(\text{Ecc}'(x)(y_1), \ldots, \text{Ecc}'(x)(y_m)\right).$$

They use a code with $\bar{n} = n^{O(k)}$, so their extractor needs a seed of length $O(mk \log n)$. Using our code Ecc_t for some $t = O(k)$, we immediately have a extractor of the same quality but needing a seed of length only $O(m(k + \log n))$.

For large m, we can get a dramatic improvement by picking some pseudo-random collection of m dimensions instead of m random ones for projection. This is the idea behind Trevisan's extractor construction [16] and the improvement by Raz, Reingold, and Vadhan [15]. The pseudo-random projection is determined by some set system defined next.

Definition 6. *[15] A family of sets $S_1, \ldots, S_m \subseteq [s]$ is called a weak (ℓ, ρ)-design if*

- *$\forall i$, $|S_i| = \ell$, and*
- *$\forall i$, $\sum_{j<i} 2^{|S_i \cap S_j|} \leq \rho(m-1)$.*

Lemma 9. *[15] For every ℓ, m and $\rho > 1$, there exists a weak (ℓ, ρ)-design $S_1, \ldots, S_m \subseteq [s]$ with $s = \left\lceil \frac{\ell}{\ln \rho} \right\rceil \ell$. Such a family can be found in time* $\text{poly}(m, s)$.

For a random $y \in \{-1,1\}^s$, such a weak $(\log \bar{n}, \rho)$-design gives a pseudo-random collection of m dimensions $y(S_1), \ldots, y(S_m)$, where each $y(S_i)$ is the integer in $[\bar{n}]$ represented by the $\log \bar{n}$ bits of y indexed by S_i.

Lemma 10. *[16, 15] Suppose $S_1, \ldots, S_m \subseteq [s]$ is a weak $(\log \bar{n}, \rho)$-design, and $\text{Ecc}' : \{-1,1\}^n \to \{-1,1\}^{\bar{n}}$ is a $(\frac{1}{2} - \frac{\varepsilon}{2m}, L)$-list code. Then the function $\text{Ext} : \{-1,1\}^n \times \{-1,1\}^s \to \{-1,1\}^m$ defined as*

$$\text{Ext}(x, y) = \left(\text{Ecc}'(x)(y(S_1)), \ldots, \text{Ecc}'(x)(y(S_m))\right)$$

is a strong (r, ε)-extractor, for any $r \geq \rho(m-1) + \log \frac{2L}{\varepsilon}$.

To have a strong $(\mu n, 2^{-3k})$-extractor for our encryption scheme, we use our $(\frac{1}{2} - \delta, L)$-list code Ecc_t, with

- $\delta = \frac{1}{2m2^{3k}}$,
- $t = O(k)$ large enough and $\frac{\lambda}{d} = O(1)$ small enough so that $h(\alpha) < \mu$ for $\alpha = 1 + (\frac{\lambda}{d})^2 - \delta^{4/t}$, and
- $L = \frac{2^{h(\alpha)n}}{2\delta^2} = 2^{h(\alpha)n + 6k + 1} m^2$.

Note that $\rho(m-1) + \log \frac{2L}{\varepsilon} = \rho(m-1) + h(\alpha)n + 9k + 2 + 2\log m$. This can be made smaller than μn for any $m \leq n^{\gamma}$ with constant $\gamma \in (0,1)$, by choosing a proper $\rho = n^{O(1)}$. Using such a weak $(\log \bar{n}, \rho)$-design and our $(\frac{1}{2} - \delta, L)$ list-code Ecc_t, we have the following theorem.

Theorem 2. *For any constant $\gamma \in (0,1)$, there is a strong $(\mu n, 2^{-3k})$-extractor with a seed of length $O((k+\log n)^2/\log n)$ and a output of length n^γ, where each output bit is the parity of $O(k)$ input bits.*

This, together with Theorem 1, gives the encryption scheme we claim. Recall that n is typically much larger than k. A larger n provides a higher security but only costs a negligible slow-down during encryption time. If we assume or choose $n = 2^{\Omega(k)}$, then we have an encryption scheme using a private key of length only $O(\log n)$.

References

1. N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley Interscience, New York, 1992.
2. Y. Aumann, Y. Z. Ding, and M. O. Rabin. Everlasting security in the bounded storage model. To appear in *IEEE Transactions on Information Theory*, 2002.
3. M. Bellare, O. Goldreich and M. Sudan. Free bits, PCPs and non-approximability — towards tight results. *SIAM Journal on Computing*, 27(3), pages 804–915, 1998.
4. C. Cachin and U. Maurer. Unconditional security against memory-bounded adversaries. In *Advances in Cryptology – CRYPTO'97*, Lecture Notes in Computer Science, Springer-Verlag, vol. 1294, pages 292–306, 1997.
5. Y. Z. Ding and M. O. Rabin. Hyper-encryption and everlasting security. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science*, pages 1–26, 2002.
6. S. Dziembowski and U. Maurer. Tight security proofs for the bounded-storage model. To appear in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002.
7. O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3), pages 407–420, 1981.
8. A. Lubotzky, R. Philips, and P. Sarnak. Explicit expanders and the Ramanujan conjecture. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 240–246, 1986.
9. F. J. MacWilliams and N. J. A. Sloan. *The Theory of Error-Correcting Codes*. Noth-Holland, 1981.
10. U. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1), pages 53–66, 1992.
11. N. Nisan. Extracting randomness: how and why — a survey. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 44–58, 1996.
12. N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2), pages 149–167, 1994.
13. N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1), pages 43–52, 1996.
14. R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 1–10, 2000.
15. R. Raz, O. Reingold, and S. P. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 149–158, 1999.

16. L. Trevisan. Extractors and pseudorandom generators. *Journal of ACM*, 48(4), pages 860–879, 2001.
17. A. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on the Foundations of Computer Science*, pages 80–91, 1982.

A Proof of Lemma 1

$$\begin{aligned}
&\|\langle g(A,B),B\rangle - \langle g(A,B'),B'\rangle\| \\
&= \frac{1}{2}\sum_b\sum_c |\Pr[g(A,B)=c \wedge B=b] - \Pr[g(A,B')=c \wedge B'=b]| \\
&= \frac{1}{2}\sum_b\sum_c |\Pr[g(A,b)=c]\Pr[B=b] - \Pr[g(A,b)=c]\Pr[B'=b]| \\
&= \frac{1}{2}\sum_b\sum_c \Pr[g(A,b)=c]\,|\Pr[B=b] - \Pr[B'=b]| \\
&= \frac{1}{2}\sum_b |\Pr[B=b] - \Pr[B'=b]| \\
&= \|B-B'\|,
\end{aligned}$$

where the second equality holds because A is independent of both B and B'.

B Proof of Lemma 2

The expectation of $\|\langle A \mid B=b\rangle - U\|$, with b sampled from the distribution B, is

$$\begin{aligned}
&\sum_b \Pr[B=b]\frac{1}{2}\sum_a \left|\Pr[A=a \mid B=b] - \frac{1}{|A|}\right| \\
&= \frac{1}{2}\sum_{a,b}\left|\Pr[(A,B)=(a,b)] - \Pr[B=b]\frac{1}{|A|}\right| \\
&\le \frac{1}{2}\sum_{a,b}\left|\Pr[(A,B)=(a,b)] - \frac{1}{|A||B|}\right| + \frac{1}{2}\sum_{a,b}\left|\Pr[B=b]\frac{1}{|A|} - \frac{1}{|A||B|}\right| \\
&= \|\langle A,B\rangle - U\| + \|B-U\| \\
&\le 2\varepsilon.
\end{aligned}$$

Then the lemma follows from the Markov inequality.