Lecture Notes

Quantum Cryptography Week 4:

From imperfect information to (near) perfect security

Contents

This week we discuss *privacy amplification*. This task is an essential component of many cryptographic protocols; in particular it forms the final step in the quantum key distribution protocols we'll see in the coming weeks. Moreover, we'll see that privacy amplification can be achieved using a beautiful family of objects from theoretical computer science called *randomness extractors* — themselves well worth studying in their own right!

4.1 Privacy amplification

Let's start by introducing the task of privacy amplification. Imagine (as usual!) that Alice and Bob want to use cryptography to exchange messages securely. For this they have access to a classic public communication channel: they can send each other any messages they like, *but* the channel is public: the malicious eavesdropper Eve may be listening in on the whole communication. Our only cryptographic assumption on the channel is that it is *authenticated*, meaning that when Alice (or Bob) receives a message she has the guarantee that it came directly from Bob (or Alice). (We will return to the topic of authentication in Week 6; for the time being think of it as a convenient assumption that will usually be met in practice. We will also assume the channel is noiseless, which in practice is easily ensured by a proper use of error-correcting codes.)

Alice and Bob would like to use symmetric-key cryptography: they know (as you do!) that the one-time pad is unconditionally secure, so the only thing they need is to come up with a shared secret key. Moreover, Alice and Bob being old-time friends, they already have a lot of shared secrets, such as the flavor of the first ice-cream cone they shared. By putting all these secrets together and translating them in a string of bits, they're pretty confident they can come up with some value, call it $x \in \{0,1\}^n$, that's fairly secret...but only "fairly" so. Unfortunately they're not fully confident about which parts of x can be considered a secret, and which may have leaked. Alice might have told her best friend Mary about the ice-cream. She definitely wouldn't have told Mary about her (embarrassing) all-time favorite cheeky cartoon, but then her little brother John might now about this...Is there a way for Alice and Bob to somehow "boil down" the secrecy that x contains, throwing away some of the bits but without knowing a priori which are secure and which may potentially have been leaked?

Answer: yes! This is precisely what privacy amplification will do for them. To describe the task more precisely, consider the following scenario. Two mutually trusting parties, Alice and Bob, each holds a copy of the same string of bits x, which we'll call a "weak secret". This secret is taken from a certain distribution p_x, which we can represent through a random variable X; later on we'll call X the "source". The distribution of X itself is not known, but the sample x is available to both parties. An eavesdropper has side information E that may be correlated to X; for example E could be the first bit of X, the parity of X, or an arbitrary quantum state ρ_x^E. Given this setup, the goal for Alice and Bob is to each produce the same string z, which could be shorter than x but must be such that the distribution of z (represented via a random variable Z) is (close to) uniform, even from the point of view of the eavesdropper.

To summarize using symbols, privacy amplification is the transformation:

$$\rho_{XE} \qquad \xmapsto{\mathrm{PA}_X \otimes \mathbb{I}_E} \qquad \rho_{ZE} \approx_\varepsilon \frac{\mathbb{I}_Z}{|Z|} \otimes \rho_E. \tag{4.1}$$

Of course this will only be possible under some assumption on X: for example if $X = E$ always there is not much we can do. Given what we've already learned, it's natural to measure the "potential for privacy amplification" of a source X through the min-entropy (equivalently, the guessing probability) $\mathrm{H}_{\min}(X|E)$, as this is a measure of "uncertainty" the eavesdropper has about X. But we're getting ahead of ourselves. First let's see how to perform a simpler but closely related task, *randomness extraction*. Then we'll see how to use this to achieve privacy amplification.

Before diving in, consider the following warm-up exercise:

Exercise 4.1.1 Suppose that $X \in \{0,1\}^3$ is uniformly distributed, and $E = X_1 \oplus X_2 \in \{0,1\}$. Give a protocol for privacy amplification that outputs two secure bits (without any communication). What if $E = (X_1 \oplus X_2, X_2 \oplus X_3) \in \{0,1\}^2$, can you still do it? If not, give a protocol extracting just one bit.

Suppose the eavesdropper is allowed to keep any two of the bits of X as side information. Give a protocol for Alice and Bob to produce a Z which contains a single bit that is always uniformly random, irrespective of which two bits of X are stored by the eavesdropper. How about an R that contains two bits — can they do it? ∎

4.2 Randomness extractors

In the task of randomness extraction there is a single party, Alice, who has access to an n-bit string x with distribution X. We call X the *source*. X is unknown, and it may be correlated to an additional system E over which Alice has no control. For example, E could contain some information about the way in which the source was generated, or some information that an adversary has gathered during the course of an earlier protocol involving the use of X. The only promise that is given to Alice is a lower bound on the min-entropy, $\mathrm{H}_{\min}(X|E) \geq k$. Alice's goal is to produce a new string Z that is close to uniform and uncorrelated with E. (As you can see, this problem is very similar to privacy amplification, but without the added complication of Alice having to coordinate with Bob!)

Now, of course Alice could dump X and create her own uniformly random Z, say by measuring a $|0\rangle$ qubit in the Hadamard basis. To make the problem interesting we won't allow any quantum resources to Alice. She also doesn't have that much freely accessible randomness — maybe she can get some, but it will be slow and costly. So Alice's goal is to leverage what she has to the best she can: she wants to *extract* randomness from X, not import it from some magical elsewhere!

4.2.1 Randomness sources

Let's see some concrete examples of sources X, and how it is possible to extract uniform bits from them.

I.I.D. sources

The simplest case of a randomness source is the i.i.d. source, where the term i.i.d. stands for *independent and identically distributed*. A (classical) i.i.d. source $X \in \{0,1\}^n$ has a distribution $\{p_x\}$ which has a product form: there is a distribution $\{p_0, p_1\}$ on a single bit such that for all $(x_1, \ldots, x_n) \in \{0,1\}^n$,

$$\Pr[X = (x_1, \ldots, x_n)] = \Pr[X_1 = x_1] \cdots \Pr[X_n = x_n] = p_{x_1} \cdots p_{x_n}.$$

Such sources are sometimes called *von Neumann* sources, since they were already considered by von Neumann. If you are curious about the history of randomness extraction, go look up the von Neumann extractor online!

Can we extract uniformly random bits from an i.i.d. source? As a warmup, let's consider how we could obtain a nearly uniform bit from a source such that each bit X_i is 0 with probability $p_0 = 1/4$ and 1 with probability $p_1 = 3/4$. Suppose we let $Z = X_1 \oplus X_2 \oplus \ldots \oplus X_n \in \{0,1\}$ be the parity of all n bits of X. Our goal is to show that $\Pr[Z = 0] \approx 1/2 \pm \varepsilon$ for reasonably small ε.

- Let's first consider $n = 2$. How well does our strategy work? We can compute

$$\begin{aligned}\Pr[Z = 0] &= \Pr[X_1 = 0 \wedge X_2 = 0] + \Pr[X_1 = 1 \wedge X_2 = 1] \\ &= p_0^2 + p_1^2 = 1/16 + 9/16 = 0.625,\end{aligned}$$

  and using a similar calculation we find $\Pr[Z = 1] = 0.375$. Not quite uniform, but closer than what we started with!

- By doing the calculation for increasingly large values of n you will see that the trace distance ε of Z from a uniformly distributed random variable gets smaller and smaller. At what rate? Give a bound on ε as a function of n. Do you find our procedure efficient?

Independent bit sources

A slightly broader class of sources are *independent bit* sources. As their name suggests such sources are characterized by the condition that each bit is chosen independently; however the distribution could be different for different bits. Clearly, any i.i.d. source is also an independent bit source, but the converse does not hold.

Exercise 4.2.1 Show that there exists an independent 2-bit source X such that $\Pr[X = (0,0)] = \Pr[X = (1,1)] = 3/16$, but there is no i.i.d. source satisfying the same condition. ■

It turns out that taking the parity of all the bits in the string generated by an independent bit source still results in a bit that is increasingly close to uniform as $n \to \infty$, provided each bit from the source is not fully biased to start with: $0 < \Pr[X_j = 0] < 1$ for all j.

Exercise 4.2.2 Let X be an independent n-bit source such that $\delta < \Pr[X_j = 0] < 1 - \delta$ for some $\delta > 0$ and all $j \in \{1, \ldots, n\}$. Give an upper bound on the distance from uniform of the parity of the bits of x, as a function of the number of bits n of X and δ. ■

Bit-fixing sources

Bit-fixing sources are a special case of independent sources where each bit of X can be of one of two kinds only: either the bit is completely fixed, or it is uniformly random. For example, the three-bit source X such that $\Pr[X = (1,0,0)] = \Pr[X = (1,1,0)] = 1/2$, with all other probabilities being 0, is a bit-fixing source: the first bit is fixed to 1, the second is uniformly random, and the third is fixed to 0.

You can verify for yourselves that, just as for the previous two types of sources we considered, taking the parity of all bits from a bit-fixing source gives a uniformly random bit. This time, we do even better: as long as at least one of the bits from the source is not fixed, the parity is (exactly) uniformly random.

General sources

The randomness sources we just discussed all have something in common: they produce a string in which each bit is chosen *independently*. What if we relax this condition?

Consider a tricky example, called an *adversarial* bit-fixing source: this is the same as a bit-fixing source, except the value taken by the fixed bits can depend on the previous bits. For example, the three-bit source X such that $\Pr[X = (1,0,0)] = \Pr[X = (1,1,1)] = 1/2$, with all other probabilities being 0, is an adversarial bit-fixing source: the first bit is fixed to 1, the second is uniformly random, and the third is fixed to, either 0 if the second was a 0, or 1 if the second was a 1. To see that this kind of source can be much more tricky, first check that our earlier choice of Z as the parity of all the bits of X no longer works on the example. However, the parity of the first two, or the first and last, bits does work on that example. Nevertheless, show that for any fixed choice of a subset of bits, there exists an adversarial bit-fixing source such that only one bit is fixed, but nevertheless the parity of the bits in the chosen subset is a constant — arbitrarily far from uniform!

As you can imagine there is a whole jungle of possible kinds of sources. How do we classify them? For the purposes of extracting randomness, we aim to measure the inherent uncertainty of the source, or in other words its *entropy*. It turns out that the min-entropy provides just the "right" measure of extractable randomness, in a precise way that we'll soon see.

Definition 4.2.1 A random variable X is a k-source if $\mathrm{H}_{\min}(X) \geq k$.

Before we move on, we should realize there is something crucial missing from this definition. Remember we're going to apply the idea of randomness extraction to a cryptographic task, privacy amplification. But we forgot to account for the eavesdropper! The process of randomness extraction is not going to happen in a void: we ought to take into account the possibility for an additional system E that may be correlated with X. Call E the *side information*. X is a classical string of bits, but E may be quantum. How do we model this? The proper way to do it is to introduce a cq state ρ_{XE}, which in general takes the form

$$\rho_{XE} = \sum_x |x\rangle\langle x|_X \otimes \rho_x^E,$$

where each ρ_x^E is positive semidefinite and $\mathrm{Tr}(\rho_{XE}) = \sum_x \mathrm{Tr}(\rho_x^E) = 1$. Using side information gives us a convenient way to model any source X as the result of an initially uniform string about which the adversary has gained partial information. For instance, you can think of a bit-fixing source as a uniform source correlated with a system E which contains some of the bits of x.

Exercise 4.2.3 Let X be an independent source, where the i-th bit X_i has distribution $\{p_i, 1-p_i\}$. Show that there exists a pair of correlated random variables (Y,Z) on $\{0,1\}^n \times \{0,1\}^n$ such that Y is uniformly distributed in $\{0,1\}^n$ but for any $z \in \{0,1\}^n$ the random variable $V = Y_{|Z=z}$ is such that V_i has the same distribution as X_i if $z_i = 0$, and as $1 - X_i$ if $z_i = 1$. ■

Let's update our definition:

Definition 4.2.2 A cq state ρ_{XE} is called a k-source if $\mathrm{H}_{\min}(X|E) \geq k$.

Can we construct extraction procedures that produce uniformly random bits from any k-source, without knowing anything else about the source?

4.2.2 Strong seeded extractors

In all examples we've seen so far we applied a fixed function, call it Ext, to the source X; for example we considered $\mathrm{Ext}(X) = X_1 \oplus \cdots \oplus X_n$. Such a function is known as a deterministic extractor, meaning that is it just one fixed function $Z = \mathrm{Ext}(X)$ that does not introduce any randomness beside what is already present in X.

Ideally we'd like to show that it is possible to extract randomness from any k-source using such a deterministic function. Unfortunately this is not possible: there is no fixed deterministic procedure that can be used to extract even a *single* bit of randomness from every possible k-source, even when k is almost maximal, $k = n-1$! This is a bit disappointing, but let's understand why.

Lemma 1 For any function $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}$ there exists an $(n-1)$-source X such that $\mathrm{Ext}(X)$ is constant.

Proof. Let $b \in \{0,1\}$ be such that $|S_b| \geq 2^n/2 = 2^{n-1}$ with $S_b = \{x \mid \mathrm{Ext}(x) = b\}$. Note that there must exist such a b. Choose a subset $S' \subseteq S_b$ such that $|S'| = 2^{n-1}$. Define X by the following distribution:

$$p_x = \begin{cases} 1/2^{n-1} & \text{if } x \in S' \,, \\ 0 & \text{otherwise} \,. \end{cases} \tag{4.2}$$

Clearly, $\mathrm{H}_{\min}(X) = n-1$, but $\mathrm{Ext}(X) = b$ is a constant! ■

Have we reached the end of the road — are we stuck to designing special-purpose functions which only work for this or that special kind of source, as we did with independent sources? Luckily

there is a way out, but we're going to need an additional resource: a little extra randomness. This extra randomness will be called the *seed* of the extractor; think of it as a second input $Y \in \{0,1\}^d$ to which Alice has access and is promised to be uniformly random and independent from X and E. This gives us the notion of a *seeded extractor*:

Definition 4.2.3 A (k,ε)-*weak seeded randomness extractor* is a function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for any k-source ρ_{XE},

$$D\Big(\rho_{\mathrm{Ext}(X,Y)E}, \frac{\mathbb{I}}{2^m} \otimes \rho_E\Big) \leq \varepsilon \,, \tag{4.3}$$

where $Y \sim U_d$ is uniformly distributed and independent from X and E, and

$$\rho_{\mathrm{Ext}(X,Y)E} = \sum_z |z\rangle\langle z|_Z \otimes \rho_z^E \qquad \text{with} \qquad \rho_z^E = 2^{-d} \sum_y \sum_{x:\mathrm{Ext}(x,y)=z} \rho_x^E .$$

If the seed is perfectly uniform, why don't we just return it as our output: define $\mathrm{Ext}(X,Y) = Y$? Well, this satisfies the definition. So maybe there is something wrong with the definition? Remember that our goal is to extract randomness from X, and that additional uniform randomness should not be considered free. So we want to keep Y as small as possible, even though X, and k, could be very large, in which case we'd like to maintain a long output (large m) with only a small help from the seed (small d).

A better answer considers our ultimate goal of privacy amplification. Remember that in that setting Alice and Bob share a weak secret X, and they want to produce a uniformly random secret R. Our solution of an extractor outputting its seed would be similar to asking Alice and Bob to throw away their initial secret X and share a fresh random string Y. But they only have access to a public communication channel, how would they agree on the same Y without the eavesdropper learning it as well?

This motivates a stronger definition of extractor, which is the one we'll use from now on:

Definition 4.2.4 A (k,ε)-*strong seeded randomness extractor* is a function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for any k-source ρ_{XE},

$$D\big(\rho_{\mathrm{Ext}(X,Y)YE}, \frac{\mathbb{I}}{2^m} \otimes \rho_{YE}\big) \leq \varepsilon \,, \tag{4.4}$$

where $Y \sim U_d$ is uniformly distributed and independent from X and E.

Before we start trying to construct strong extractors, let's consider the notion of k-source a little more closely. Why do we think that the min-entropy provides the right way to quantify the amount of randomness that can be extracted from a given source?

Let's first argue informally that the min-entropy is an upper bound on the amount of extractable randomness: there is no strong extractor that has output length more than $\mathrm{H}_{\min}(X|E)$. To see why this is the case, recall that $\mathrm{H}_{\min}(X|E) = -\log P_{\mathrm{guess}}(X|E)$. Suppose now that we apply some function f to X. How hard is it to guess $f(X)$ given E, i.e., what's $P_{\mathrm{guess}}(f(X)|E)$? Clearly, since one way to guess $f(X)$ is to guess X, and then apply f to our guess, we have $P_{\mathrm{guess}}(f(X)|E) \geq P_{\mathrm{guess}}(X|E)$. However, this is equivalent to

$$\mathrm{H}_{\min}(f(X)|E) \leq \mathrm{H}_{\min}(X|E) \,. \tag{4.5}$$

This means that also the output of the extractor, which for fixed seed y is obtained as a function $f(X) = \mathrm{Ext}(X,y)$, must have min-entropy at most $\mathrm{H}_{\min}(X|E)$, which implies that the output $\mathrm{Ext}(X,Y)$, conditioned on $Y = y$, can be uniform on at most $\mathrm{H}_{\min}(X|E)$ bits!

Now, how about a converse: does there exist a strong extractor that can extract approximately

$\mathrm{H}_{\min}(X|E)$ bits from *any* k-source ρ_{XE}? The answer turns out to be yes, and we're going to see how this can be done in the next section.

4.3 An extractor based on hashing

Much research has gone into constructing randomness extractors, and they have found many applications throughout computer science and mathematics. The quality of an extractor is measured by the parameters it achieves, and different applications require different trade-offs. The main targets consist in extracting as much randomness as possible (large m) using the smallest possible seed (small d) and with the best possible error (small ε), all from arbitrary sources with min-entropy (at least) k.

By using a probabilistic argument (select a function Ext at random from all possible functions, and fix it to be the extractor), for any given input length k and min-entropy k the best trade-offs that can be achieved are seed length $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$ and output length $m = k + d - 2\log(1/\varepsilon) + O(1)$ [RS00]. Moreover, there are efficient constructions known that achieve essentially both parameters simultaneously! Rather than aiming for optimal, but often intricate, constructions, here we will focus on a simple construction which nevertheless achieves very good parameters for the application we have in mind (privacy amplification!).

Going back to the intuition we developed on the examples, we saw that taking the parity of a random subset of the bits of the source often (but not always) provides a good way to extract a bit of randomness. In this case we can think of the seed of the extractor as specifying the subset of bits whose parity is taken. We could repeat this procedure to extract even more bits, each chosen as the parity of a different random subset. It is a good exercise to show that this procedure works, but it has one major drawback: it is excessively costly in terms of seed length, requiring an investment of approximately n bits of randomness (to specify the subset of bits whose parity is taken) for each new bit produced!

Let's see how we can do better. For this we'll have to make a little detour and learn about certain families of hash functions.

4.3.1 Two-universal families of hash functions

Informally, a hash function is a function that maps long strings to shorter strings, with the property that the output of the hash functions tends to be "well-distributed". What this means depends on the application we have in mind for the hash function — indeed, the term "hash function" can be interpreted in many different ways, with the only standard requirement, as its name indicates, being that a hash function should not increase the length of its input! An additional reasonable requirement, which formalizes the "well-distributed" aspect of the output of a hash function, is the following:

Definition 4.3.1 — 1-universal family. A family of hash functions $\mathscr{F} = \{f : \{0,1\}^n \to \{0,1\}^m\}$, where $m \leq n$, is called 1-*universal* if for every string $x \in \{0,1\}^n$ and $z \in \{0,1\}^m$ we have

$$\Pr_{f \in \mathscr{F}}[f(x) = z] = \frac{1}{2^m} . \tag{4.6}$$

It is worth reading this definition carefully: in (4.6) both x and z are fixed, and the probability is taken over a uniformly random function from the family. The condition is equivalent to saying that for any fixed x, the random variable $F(x)$, where F is uniformly distributed over all f in $\mathscr{F}$, in uniformly distributed in $\{0,1\}^m$. Let's see an example of a 1-universal family of hash functions.

Exercise 4.3.1 For any $y \in \{0,1\}^n$ let $f_y : \{0,1\}^n \to \{0,1\}^n$ be defined by $f_y(x) = x \oplus y$, where the parity is taken bitwise. Show that the family of functions $\mathscr{F} = \{f_y, y \in \{0,1\}^n\}$ is 1-universal. ■

You may want to convince yourself that a family of 1-universal hash functions is already sufficient to construct a *weak* seeded extractor: use the seed to select a random function from the family, and output the value of the function evaluated at the source. The property of 1-universality ensures that the output will be uniformly distributed, even if the input is fixed. However, recall our earlier criticism: in this case it is apparent that we are "cheating", and that all the randomness is coming from the seed. Indeed, it turns out that the property of 1-universality is not sufficient to obtain a *strong* seeded extractor. We'll need the following stronger property, first introduced by Carter and Wegman:

Definition 4.3.2 — 2-universal family. A family of hash functions $\mathscr{F} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ is called 2-*universal* if for every two strings $x, x' \in \{0,1\}^n$ with $x \neq x'$, and any two $z, z' \in \{0,1\}^m$, we have

$$\Pr_{f \in \mathscr{F}}[f(x) = z \wedge f(x') = z'] = \frac{1}{2^{2m}} . \tag{4.7}$$

Condition (4.7) in the definition would be satisfied if $f(x)$ and $f(x')$ were *jointly* chosen uniformly and independently at random in $\{0,1\}^m$. This is a stronger condition than (4.6): we now require that the pair of random variables $(F(x), F(x'))$, for F uniformly distributed over $f \in \mathscr{F}$, are jointly uniform (as an exercise, verify that the family of hash functions from Exercise 4.3.1 is *not* 2-universal).

You can check that for any $m \leq n$ the set of all possible functions $f : \{0,1\}^n \to \{0,1\}^m$ is 2-universal. But it is too big a set: it has size $|\mathscr{F}| = 2^{m2^n}$, so that selecting a function at random from the set would require a seed length $d = m2^n$! Let's see a much more efficient construction.

Let $q = 2^n$ and $\mathbb{F}_q$ the finite field with 2^n elements. (If you have never seen this field before, the details of its construction will not be matter to us, but you may still want to check it out online! The multiplication rule is *not* the same as multiplication over the integers, $\mod 2^n$.) For any $(a,b) \in \mathbb{F}_q^2$ let

$$f_{a,b} : \mathbb{F}_q \to \mathbb{F}_q, \qquad f_{a,b}(x) = ax + b,$$

where addition and multiplication are done in $\mathbb{F}_q$. Then $\mathscr{F} = \{f_{a,b}, (a,b) \in \mathbb{F}_q^2\}$ is a 2-universal family of only $q^2 = 2^{2n}$ hash functions. To show this we need to verify that equation (4.7) from the definition holds. So let's fix distinct $x \neq x' \in \mathbb{F}_q$ and two $z, z' \in \mathbb{F}_q$. What is the probability, over a uniformly random choice of (a,b), that $f_{a,b}(x) = z$ and $f_{a,b}(x') = z'$? The two conditions are equivalent to $ax + b = z$ and (taking the difference) $a(x' - x) = z' - z$, thus $a = (z' - z)/(x' - x)$, where the condition $x \neq x'$ and the fact that $\mathbb{F}_q$ is a field allows us to perform the division. This equation determines a unique possible value for a. Moreover, once a is fixed there is a unique possible value for b: $b = z - ax$ (this shouldn't be a surprise, since we started with two linear equations and two unknowns). Out of 2^{2m} possibilities, we end up with a single one: $\Pr_{a,b}[f_{a,b}(x) = z \wedge f_{a,b}(x') = z'] = 2^{-2m}$, as desired.

One last technicality: recall that our goal was to construct a 2-universal family of functions $f : \{0,1\}^n \to \{0,1\}^m$, for arbitrary n and $m \leq n$, whereas what we managed to construct so far are functions from $\mathbb{F}_q \to \mathbb{F}_q$. Since $|\mathbb{F}_q| = q = 2^n$ the domain of f can be identified with $\{0,1\}^n$ in an arbitrary way. The range of f may be bigger than $\{0,1\}^m$, but there is a simple solution: throw away the last $(n-m)$ bits of $f(x)$! I'll let you verify that this works, i.e. it preserves the property of 2-universality.

4.3.2 The 2-universal extractor

Equipped with an arbitrary family of 2-universal hash functions, we define an extractor as follows.

Definition 4.3.3 — 2-universal extractor. Let $\mathscr{F} = \{f_y : \{0,1\}^n \to \{0,1\}^m, y \in \{0,1\}^d\}$ be a 2-universal family of hash functions such that $|\mathscr{F}| = 2^d$. The associated 2-universal extractor is

$$\mathrm{Ext}_{\mathscr{F}} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m, \qquad \mathrm{Ext}_{\mathscr{F}}(x,y) = f_y(x).$$

One way to think of $\mathrm{Ext}_{\mathscr{F}}$ is as using its seed y to select a function from the family $\mathscr{F}$ uniformly at random, and then returning the output of the function when evaluated on the source X.

How good is this extractor? The key result required to analyze it is known as the *leftover hash lemma*. It was first proven by Impagliazzo, Levin, and Luby for the case when there is no side information E, and later extended to the case of quantum E by Renner. Here is a statement of the lemma when there is no side information.

Theorem 4.3.1 — Leftover hash lemma. Let n and $k \leq n$ be arbitrary integers, $\varepsilon > 0$, $m = k - 2\log(1/\varepsilon)$, and $\mathscr{F} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ a 2-universal family of hash functions. Then the 2-universal extractor $\mathrm{Ext}_{\mathscr{F}}$ is a (k,ε)-strong seeded randomness extractor.

In the previous section we saw how to construct a 2-universal family with 2^{2n} functions, meaning that the seed length of the two-universal extractor is $2n$. This is much longer than the optimal length $d \approx O(\log(n/\varepsilon))$, and it can be a drawback in some applications for which the randomness required to produce the seed is particularly costly. However, for our application to privacy amplification, and especially later to quantum key distribution, it is not a significant limitation. Much more important for us is the dependence of the output length on the initial min-entropy, which will ultimately govern the length of key that we are able to produce. In this respect the two-universal construction is essentially optimal, a good reason to use it!

4.3.3 Analysis with no side information

We first prove the leftover hash lemma in the case when there is no side information, stated in Theorem 4.3.1. This will be a good warm-up for the general case, which will follow the same structure.

The proof proceeds in two steps. In the first step we reduce our ultimate goal, bounding the error of the extractor, i.e. the trace distance between the extractor's output and the uniform distribution, to bounding a different quantity called the *collision probability*. In the second step we show that the collision probability is sufficiently small to imply the desired bound on the error of the extractor.

(i) From trace distance to collision probability.

Our goal is to bound $D(\rho_{\mathrm{Ext}(X,Y)Y}, 2^{-(m+d)}\mathbb{I})$, where X has min-entropy at least k and Y is uniformly distributed over d-bit strings. The joint distribution of $(Z = \mathrm{Ext}(X,Y), Y)$ is given by

$$p_{zy} = \Pr[(\mathrm{Ext}(X,Y),Y) = (z,y)] = 2^{-d} \sum_{x : f_y(x) = z} p_x. \tag{4.8}$$

Using the definition of the trace distance, we get

$$\begin{aligned} D(\rho_{\mathrm{Ext}(X,Y)Y}, 2^{-(d+m)}\mathbb{I}) &= \frac{1}{2}\sum_{z,y} \left| 2^{-d} \sum_{x : f_y(x) = z} p_x - 2^{-d-m} \right| \\ &\leq 2^{\frac{m}{2}-1} \left(2^{-d} \sum_{z,y} \left| \sum_{x : f_y(x) = z} p_x - 2^{-m} \right|^2 \right)^{1/2} \\ &= 2^{\frac{m}{2}-1} \left(2^{d} \sum_{z,y} p_{zy}^2 - 2^{-m} \right)^{1/2}, \end{aligned}$$

where for the second line we applied the Cauchy-Schwarz inequality. This completes our first step. The quantity $CP(ZY) = \sum_{z,y} p_{zy}^2$ is called the collision probability of (Z,Y), and we turn to bounding it next.

(ii) A bound on the collision probability.
Using the definition (4.8) and expanding the square,

$$\begin{aligned}\sum_{z,y} p_{zy}^2 &= 2^{-2d} \sum_{y,z} \sum_{\substack{x,x':\\ f_y(x)=f_y(x')=z}} p_x p_{x'} \\ &= 2^{-2d} \sum_{y,z} \Big(\sum_{\substack{x\neq x':\\ f_y(x)=f_y(x')=z}} p_x p_{x'} + \sum_{x:\, f_y(x)=z} p_x^2 \Big) \\ &= 2^{-(d+m)} \sum_{x\neq x'} p_x p_{x'} + 2^{-d} \sum_x p_x^2 \\ &\leq 2^{-(d+m)} + 2^{-(d+k)}.\end{aligned}$$

Here the crucial step is in bounding the summation over $x \neq x'$ when going from the second to the third line: we are using the property of 2-universality to argue that for any $x \neq x'$ there is a fraction exactly 2^{-m} of all f_y that map both x and x' to the same value. To bound the second term in going from the second-last to last lines we used $\sum_x p_x^2 \leq \max_x p_x = 2^{-\mathrm{H}_{\min}(X)}$ and the assumption $\mathrm{H}_{\min}(X) \geq k$.

Plugging this back into the bound on the trace distance from (i) we obtain

$$D(\rho_{\mathrm{Ext}(X,Y)Y}, 2^{-(d+m)}\mathbb{I}) \leq 2^{\frac{m-k}{2}-1},$$

proving the lemma.

4.3.4 The pretty good measurement and quantum side information

We would like to extend the proof in the previous section to the case where the source X is correlated with some quantum side information E, that is, $\rho_{XE} = \sum_x |x\rangle\langle x| \otimes \rho_x^E$ is an arbitrary cq state such that $\mathrm{H}_{\min}(X|E) \geq k$. Before diving into this, let's make a small detour by considering the related problem of optimally distinguishing between a set of quantum states.

The pretty-good measurement
Let $\rho_{XE} = \sum_x |x\rangle\langle x| \otimes \rho_x^E$ be a cq state. What is the optimal probability with which Eve, holding the quantum system E, can successfully guess x? We've seen this problem already: the answer is captured by the guessing probability,

$$P_{\mathrm{guess}}(X|E)_\rho = \max_{\{M_x\}} \sum_x \mathrm{Tr}(M_x \rho_x^E), \tag{4.9}$$

where the maximum is taken over all POVM $\{M_x\}$ on E. But what is the best POVM? If $x \in \{0,1\}$ takes only two values you've already seen the answer: in this case we can write

$$\begin{aligned}\mathrm{Tr}(M_0\rho_0^E) + \mathrm{Tr}(M_1\rho_1^E) &= \mathrm{Tr}\left(\frac{M_0+M_1}{2} \cdot \left(\rho_0^E + \rho_1^E\right)\right) + \mathrm{Tr}\left(\frac{M_0-M_1}{2} \cdot \left(\rho_0^E - \rho_1^E\right)\right) \\ &\leq \frac{1}{2} + \frac{1}{2} D(\rho_0^E, \rho_1^E),\end{aligned}$$

and moreover the last inequality is an equality if M_0 and M_1 are the projectors on the positive and negative eigenspaces of the Hermitian matrix $\rho_0^E - \rho_1^E$ respectively.

When $|X| > 2$ unfortunately the situation is a bit more murky. The problem of finding the optimal measurement can be solved efficiently with a computer by expressing the optimization problem (4.9) as a *semidefinite program*, a generalization of linear programs for which there are efficient algorithms. But what we'd really like is a nice, clean mathematical expression for what the optimal measurement is, so that we can work with it in our proofs! No such simple closed form is known. However, what we can do is find a simple measurement that always achieves *close* to the optimum: the *pretty-good measurement*.

So what is this "pretty-good" measurement? To get some intuition first consider the case where the states ρ_x^E are perfectly distinguishable; for example $\rho_x^E = p_x|x\rangle\langle x|$ is simply a classical copy of X. Then it is clear what we should do: measure in the computational basis, and recover x! Observe that in this case the POVM elements M_x are directly proportional to ρ_x: we can think of the states as "pointing" in some direction correlated with x, and it is natural to make a measurement along that direction.

Can we generalize this idea? Let's try defining $M_x = \rho_x^E$. This is positive semidefinite, so it satisfies the first condition for a POVM. However, $\sum_x M_x = \sum_x \rho_x^E = \rho^E$ is not necessarily the identity, as required by the second condition. The solution? Normalize!

Definition 4.3.4 Given a collection of positive semidefinite matrices $\{\rho_x\}$, the *pretty-good measurement* (PGM) associated to the collection is the POVM with elements

$$M_x = \rho^{-1/2}\rho_x\rho^{-1/2},$$

where $\rho = \sum_x \rho_x$ and the inverse is the Moore-Penrose pseudo-inverse, i.e. we use the convention $0^{-1} = 0$.

Note how we dealt with division by zero in the definition. Defining division by zero may seem odd, but this convention makes sense in the context of linear operators. If ρ is orthogonal to some subspace, i.e. it is an eigenspace of eigenvalue 0, then the pseudo-inverse ρ^{-1} should also be orthogonal to that subspace. A useful property of this convention is that it makes it so that if P is an orthogonal projection and $P\rho P$ is invertible, then $(P\rho P)^{-1} = P\rho^{-1}P$.

How well does the pretty-good measurement compare to the optimal guessing measurement? Let $\{N_x\}$ be an optimal guessing POVM for Eve. Then by definition

$$\begin{aligned}
P_{\text{guess}}(X|E) &= \sum_x \operatorname{Tr}\left(N_x\rho_x^E\right) \\
&= \sum_x \operatorname{Tr}\left((\rho^{1/4}N_x\rho^{1/4})(\rho^{-1/4}\rho_x^E\rho^{-1/4})\right) \\
&\leq \Big(\sum_x \operatorname{Tr}\left(\rho^{1/2}N_x\rho^{1/2}N_x\right)\Big)^{1/2}\Big(\sum_x \operatorname{Tr}\left(\rho^{-1/2}\rho_x^E\rho^{-1/2}\rho_x^E\right)\Big)^{1/2} \\
&\leq \left(\text{PGM}(X|E)\right)^{1/2},
\end{aligned}$$

where

$$\text{PGM}(X|E) = \sum_x \operatorname{Tr}(M_x\rho_x^E) = \sum_x \operatorname{Tr}\left(\rho^{-1/2}\rho_x\rho^{-1/2}\rho_x\right) \tag{4.10}$$

is the success probability of the PGM in the guessing task. The second and third lines are the most important here. To go from the first to the second line we inserted factors $\rho^{1/4}$ and $\rho^{-1/4}$ that cancel each other out (using cyclicity of the trace), but are important for normalization. To go from the second to the third line we used the Cauchy-Schwarz inequality twice: first, for each x we apply a matrix version of the inequality,

$$\left|\operatorname{Tr}(AB)\right| \leq \left(\operatorname{Tr}(AA^\dagger)\right)^{1/2}\left(\operatorname{Tr}(BB^\dagger)\right)^{1/2}, \tag{4.11}$$

with $A = \rho^{1/4} N_x \rho^{1/4}$ and $B = \rho^{-1/4} \rho_x^E \rho^{-1/4}$; and second, we apply the usual version

$$\Big|\sum_x a_x b_x\Big| \leq \Big(\sum_x a_x^2\Big)^{1/2} \Big(\sum_x b_x^2\Big)^{1/2},$$

valid for any real a_x and b_x (here $a_x = \mathrm{Tr}(\rho^{1/2} N_x \rho^{1/2} N_x)$ and $b_x = \mathrm{Tr}(\rho^{-1/2} \rho_x^E \rho^{-1/2} \rho_x^E)$). Finally to get to the last line we used $\sum_x N_x = \mathbb{I}$ to bound the first term, and the definition of the pretty-good measurement for the second.

Proof of the leftover hash lemma with quantum side information

The proof follows the same structure as the proof we saw for the case with no side information, but it is slightly more involved technically. We will use the following inequality:for any positive Hermitian σ and positive semidefinite τ such that $\mathrm{Tr}(\tau) = 1$ and the support of τ contains the support of σ,

$$\mathrm{Tr}\big(|\sigma|\big) \leq \mathrm{Tr}\big((\tau^{-1/4} \sigma \tau^{-1/4})^2\big)^{1/2}. \tag{4.12}$$

To prove the inequality, observe that

$$\begin{aligned}
\mathrm{Tr}\big(|\sigma|\big) &= \mathrm{Tr}\big(\tau^{1/4} \tau^{-1/4} |\sigma| \tau^{-1/4} \tau^{1/4}\big) \\
&= \mathrm{Tr}\big(\tau^{1/4} |\tau^{-1/4} \sigma \tau^{-1/4}| \tau^{1/4}\big) \\
&= \mathrm{Tr}\big(|\tau^{-1/4} \sigma \tau^{-1/4}| \tau^{1/2}\big).
\end{aligned}$$

Here the second line is obtained by computing the trace in the eigenbasis of $\tau^{-1/4} \sigma \tau^{-1/4}$; see [Ren08, Lemma 5.1.2] for details of the calculation. To conclude the proof of (4.12), apply (4.11) with the choise $A = \tau^{-1/4} \sigma \tau^{-1/4}$ and $B = \tau^{1/2}$.

(i) From trace distance to collision probability.

Our goal is to bound $D(\rho_{\mathrm{Ext}(X,Y)YE}, 2^{-(m+d)} \mathbb{I} \otimes \rho_E)$, where Y is uniformly distributed and X is such that $\mathrm{H}_{\min}(X|E) \geq k$. We can write

$$\rho_{\mathrm{Ext}(X,Y)YE} = \sum_{z,y} |z\rangle\langle z| \otimes |y\rangle\langle y| \otimes \rho_{zy}, \qquad \text{with} \qquad \rho_{zy} = 2^{-d} \sum_{x:\, f_y(x) = z} \rho_x.$$

Note that our normalization makes is so that

$$\sum_{z,y} \mathrm{Tr}(\rho_{zy}) = 2^{-d} \sum_{x,y} \mathrm{Tr}(\rho_x) = \mathrm{Tr}(\rho) = 1.$$

Since the state $\rho_{\mathrm{Ext}(X,Y)YE}$ is a ccq state, using the definition of the trace distance we can expand

$$\begin{aligned}
D(\rho_{\mathrm{Ext}(X,Y)YE}, 2^{-(d+m)} \mathbb{I} \otimes \rho_E) &= \frac{1}{2} \sum_{z,y} \big\| \rho_{zy} - 2^{-(d+m)} \rho \big\|_1 \\
&\leq 2^{\frac{m+d}{2} - 1} \Big(2^{-(m+d)} \sum_{z,y} \mathrm{Tr}\big((\rho^{-1/4} (\rho_{zy} - 2^{-m} \rho) \rho^{-1/4})^2\big)\Big)^{1/2} \\
&= 2^{\frac{m}{2} - 1} \Big(2^d \sum_{z,y} \mathrm{Tr}\big(\rho_{zy} \rho^{-1/2} \rho_{zy} \rho^{-1/2}\big) - 2^{-m}\Big)^{1/2},
\end{aligned}$$

where for the second line we first applied (4.12) for each (y,z) with $\sigma = \rho_{zy} - 2^{-(d+m)} \rho$ and $\tau = \rho$, and then the usual Cauchy-Schwarz inequality. Do you recognize the expression in the last line? Using the notation from (4.10), we have

$$\mathrm{PGM}(Z|YE) = 2^d \sum_z \mathrm{Tr}\big(\rho_{zy} \rho^{-1/2} \rho_{zy} \rho^{-1/2}\big),$$

so the sequence of equations above show that

$$D(\rho_{\mathrm{Ext}(X,Y)YE}, 2^{-(d+m)}\mathbb{I}\otimes\rho_E) \leq 2^{\frac{m}{2}-1}\big(\mathrm{PGM}(Z|YE)-2^{-m}\big)^2.$$

We have thus managed to relate the distance from uniform to the advantage of the pretty good measurement over random guessing (that would succeed with probability 2^{-m}). We can understand this step of the proof as a reduction from arbitrary attacks of an adversary to the extractor, whose optimal success probability is expressed in the first line, to attacks of a very specific form, where the adversary, given a sample (z,y), measures its side information using the pretty-good measurement associated with the family of states $\{\rho_{zy}\}$. The square root factor on the third line expresses the fact that the pretty-good measurement is quadratically far from optimal. What is the point of losing this square root? The pretty-good measurement has a crucial advantage, that we are going to use in the second step of the proof: it has a form of "linearity", in the sense that the PGM operators associated with the family of states $\{\rho_{zy}\}$ can be obtained by summing up PGM operators associated with the states $\{\rho_x\}$. Let's see how this works in our favor.

(ii) A bound on the collision probability.
Proceeding exactly as in the case with no side information, we can calculate

$$\begin{aligned}
\mathrm{PGM}(Z|YE)-2^{-m} &= 2^{-d}\sum_{y,z}\sum_{\substack{x,x':\\ f_y(x)=f_y(x')=z}}\mathrm{Tr}(\rho_x\rho^{-1/2}\rho_{x'}\rho^{-1/2})-2^{-m}\\
&= 2^{-d}\sum_{y,z}\Bigg(\sum_{\substack{x\neq x':\\ f_y(x)=f_y(x')=z}}\mathrm{Tr}(\rho_x\rho^{-1/2}\rho_{x'}\rho^{-1/2})\\
&\qquad\qquad +\sum_{x:f_y(x)=z}\mathrm{Tr}(\rho_x\rho^{-1/2}\rho_x\rho^{-1/2})\Bigg)-2^{-m}\\
&= 2^{-m}\sum_{x\neq x'}\mathrm{Tr}(\rho_x\rho^{-1/2}\rho_{x'}\rho^{-1/2})+\sum_x\mathrm{Tr}(\rho_x\rho^{-1/2}\rho_x\rho^{-1/2})-2^{-m}\\
&\leq \mathrm{PGM}(X|E).
\end{aligned}$$

Using the 2-universal hashing property, we have managed to relate the advantage over random of the pretty good measurement in guessing Z, to the success probability of the pretty good measurement to guess X directly. But the last expression is, by assumption, at most $2^{-\mathrm{H}_{\min}(X|E)}$, since the guessing probability achieved from using the PGM cannot be the optimal one. Together with the bound proven in step (i) we finally obtain

$$D(\rho_{\mathrm{Ext}(X,Y)Y}, 2^{-(d+m)}\mathbb{I}) \leq 2^{\frac{m-k}{2}-1},$$

precisely the same bound as when there was no side information at all.

4.4 Solving privacy amplification using extractors

Back to cryptography...how do we use extractors to solve privacy amplification? By now you must have a good idea how this can be done. Let Ext be a (k,ε) strong seeded randomness extractor. Here is a simple protocol:

1. Alice and Bob share a weak secret X, which may be correlated with an eavesdropper holding quantum side information E.
2. Alice choses a random seed Y for the extractor, and computes $R_A = \mathrm{Ext}(X,Y)$. She sends Y to Bob over a public communication channel.
3. Upon receiving Y, Bob sets $R_B = \mathrm{Ext}(X,Y)$.

First note that this protocol is always correct: Alice and Bob output the same string, $R_A = R_B$. Is it secure? Remember the criterion (4.1) we introduced to define security of privacy amplification. Note also that here, at the end of the protocol, Eve has access to her original side information E, but also to any communication exchanged over the public channel: precisely the seed Y. So the condition becomes

$$X : \mathrm{H}_{\min}(X|E)_\rho \geq k \quad \overset{\text{PA}}{\longmapsto} \quad R = \mathrm{Ext}(X,Y) : \rho_{RYE} \approx_\varepsilon \frac{\mathbb{I}_R}{|R|} \otimes \rho_{YE},$$

which is precisely the requirement of a (k, ε) strong extractor! All the pieces have come into place: by instantiating the extractor with the 2-universal extractor based on the 2-universal family of hash functions from Section 4.3.1 you now have a complete construction of a secure one-way protocol for privacy amplification. This will be crucially used in our quantum key distribution protocols.

Acknowledgments

 The lecture notes are written by Nelly Ng, Thomas Vidick and Stephanie Wehner. We thank David Elkouss, Kenneth Goodenough, Jonas Helsen, Jérémy Ribeiro, and Jalex Stark for proofreading. We thank Joe Renes for spotting a typo in a previous version of the notes, and suggesting a streamlined analysis of the leftover hash lemma with quantum side information.

Bibliography

[Ren08] Renato Renner. “Security of quantum key distribution”. In: *International Journal of Quantum Information* 6.01 (2008), pages 1–127 (cited on page 13).

[RS00] Jaikumar Radhakrishnan and Amnon Ta-Shma. “Bounds for dispersers, extractors, and depth-two superconcentrators”. In: *SIAM Journal on Discrete Mathematics* 13.1 (2000), pages 2–24 (cited on page 8).