CS 7880 Graduate Cryptography September 22, 2015

Lecture 3: Lower bound on statistically secure encryption, extractors

Lecturer: Daniel Wichs *Scribe: Giorgos Zirdelis*

1 Topics Covered

- Statistical Secrecy
- Randomness extractors
- Universal hash functions
- Leftover Hash Lemma

2 Statistical Secrecy

DEFINITION 1 Let X, Y be two random variables supported over some set $\mathcal{V}$. The statistical distance between X and Y is defined as follows in three equivalent ways:

$$\mathsf{SD}(X,Y) = \max_{f:\mathcal{V}\to\{0,1\}} |\Pr[f(X)=1] - \Pr[f(Y)=1]| \tag{1}$$

$$\mathsf{SD}(X,Y) = \max_{\mathcal{W}\subseteq\mathcal{V}} |\Pr[X\in\mathcal{W}] - \Pr[Y\in\mathcal{W}]| \tag{2}$$

$$\mathsf{SD}(X,Y) = \frac{1}{2}\sum_{v\in\mathcal{V}} |\Pr[X=v] - \Pr[Y=v]| \tag{3}$$

$\diamondsuit$

In the first definition we can think of f as a distinguisher between X and Y.

Claim 1 *Two properties of the statistical distance are the following:*

- *For all $g : \mathcal{V}\to\mathcal{T}$: $\mathsf{SD}(g(X),g(Y)) \le \mathsf{SD}(X,Y)$, i.e. by manipulating the variables we will not get a better probability.*
- *For all X, Y, Z: $\mathsf{SD}(X,Z) \le \mathsf{SD}(X,Y) + \mathsf{SD}(Y,Z)$ (triangle inequality).*

Proof:

- For an arbitrary deterministic function $g : \mathcal{V}\to\mathcal{T}$ we have the following:

$$\begin{aligned}\mathsf{SD}(g(X),g(Y)) &= \max_{f:\mathcal{T}\to\{0,1\}} |\Pr[f(g(X)=1] - \Pr[f(g(Y))=1]| \\ &\le \max_{f':\mathcal{V}\to\{0,1\}} \big|\Pr[f'(X)=1] - \Pr[f'(Y))=1]\big|\end{aligned}$$

Where the second line follows by thinking of $f' = f\circ g$.

- The triangle inequality.

$$\begin{aligned}
\mathsf{SD}(X,Z) &= \frac{1}{2}\sum_{v\in\mathcal{V}} |\Pr[X=v] - \Pr[Z=v]| \\
&= \frac{1}{2}\sum_{v\in\mathcal{V}} |(\Pr[X=v] - \Pr[Y=v]) + (\Pr[Y=v] - \Pr[Z=v])| \\
&\leq \frac{1}{2}\sum_{v\in\mathcal{V}} |(\Pr[X=v] - \Pr[Y=v])| + \frac{1}{2}\sum_{v\in\mathcal{V}} |(\Pr[Y=v] - \Pr[Z=v])| \\
&= \mathsf{SD}(X,Y) + \mathsf{SD}(Y,Z)
\end{aligned}$$

□

We use SD to relax the requirements we had for encryption secrecy, which is more of a parameterized notion of security.

DEFINITION 2 An encryption scheme has ε-statistical secrecy if for any $m_0, m_1 \in \mathcal{M}$ and $k \in \mathcal{K}$: $\mathsf{SD}(\mathsf{Enc}(k,m_0), \mathsf{Enc}(k,m_1)) \leq \varepsilon$. ♢

Theorem 1 *If an encryption scheme has ε-statistical secrecy then $\varepsilon \geq 1 - \frac{|\mathcal{K}|}{|\mathcal{M}|}$, where $|\mathcal{K}|$ and $|\mathcal{M}|$ are the key and message space, respectively.*

The above theorem implies that you cannot get much secrecy if the key space is e.g. half the message space.

Proof: The intuition is similar to Shannon's theorem proof. We define the following set $\mathcal{D}(c) \subseteq \mathcal{M}$ viaL

$$\mathcal{D}(c) = \{m \in \mathcal{M} : \exists k \in \mathcal{K} \text{ such that } \mathsf{Dec}(k,c) = m\}$$

It holds that $|\mathcal{D}(c)| \leq |\mathcal{K}|$.

The proof is divided into two steps. We start with the following claim.

Claim 2 *There exist m_0, m_1 such that*

$$\Pr[m_1 \in \mathcal{D}(\mathsf{Enc}(K, m_0))] \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}$$

where K is random on $\mathcal{K}$.

Proof:(Claim 2) We want to find those two messages m_0, m_1 without knowing anything about the scheme. Let $m_0 \in \mathcal{M}$ be arbitrary and let M be a uniform random variable in $\mathcal{M}$. For all $k \in \mathcal{K}$ we have that

$$\Pr[M \in \mathcal{D}(\mathsf{Enc}(k, m_0))] \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}$$

because $|\mathcal{D}(\mathsf{Enc}(k,m_0))| \leq |\mathcal{K}|$. □

Since the above holds for all keys k, it also holds for a random key K:

$$\Pr[M \in \mathcal{D}(\mathsf{Enc}(K, m_0))] = \sum_{k \in \mathcal{K}} \Pr[M \in \mathcal{D}(\mathsf{Enc}(k, m_0))]/|\mathcal{K}| \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}$$

Since the above holds for a random M, there must be some $m_1 \in \mathcal{M}$ that minimizes $\Pr[m_1 \in \mathcal{D}(\mathsf{Enc}(K, m_0))]$ and achieves:

$$\begin{aligned}\Pr[m_1 \in \mathcal{D}(\mathsf{Enc}(K, m_0))] &\leq \sum_{m \in \mathcal{M}} \Pr[m \in \mathcal{D}(\mathsf{Enc}(K, m_0))]/|\mathcal{M}| \\ &= \Pr[M \in \mathcal{D}(\mathsf{Enc}(K, m_0))] \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}\end{aligned}$$

This proves the claim. Note that we showed that m_0, m_1 as above exist, but not how to find them efficiently. □

Next we need to show that the statistical distance between $\mathsf{Enc}(K, m_0)$ and $\mathsf{Enc}(K, m_1)$ is high. To do that we will to define a distinguisher function for those two messages. Define $f : \mathcal{C} \to \{0, 1\}$ as:

$$f(c) = 1 \text{ iff } m_1 \in \mathcal{D}(c)$$

We have (by correctness of f) that $\Pr[f(\mathsf{Enc}(K, m_1)) = 1] = 1$ and by Claim 2 that $\Pr[f(\mathsf{Enc}(K, m_0)) = 1] \leq \frac{|\mathcal{K}|}{|\mathcal{M}|}$. Using the 1st definition of statistical distance we get that an if an encryption scheme has ε-statistical security then

$$\varepsilon \geq |\Pr[f(\mathsf{Enc}(K, m_1)) = 1] - \Pr[f(\mathsf{Enc}(K, m_0)) = 1]| \geq 1 - \frac{|\mathcal{K}|}{|\mathcal{M}|}$$

3 Randomness extraction

In all schemes that we saw there was involvement of randomness. So where does this randomness comes from? Computers don't toss coins but rely on "random" events such as mouse clicking/movement and interrupts. These events have entropy but aren't uniformly random. Can we convert a source of entropy into uniform randomness? The short answer is, yes, using a tool called a randomness extractor.

Consider the following example where we want to "create" uniform randomness out of biassed coin tosses.

Example 1 *Assume we have a biassed coin and as many independent throws as we want,* $\Pr[B = 1] = p$ *(Heads) and* $\Pr[B = 0] = 1 - p$ *(Tails). The following method is due to von Neumann.*

Sample: $b_1 \leftarrow B$, $b_2 \leftarrow B$;
if $b_1 = b_2$ **then**
  sample again
else
  if $b_1 = 0$ **and** $b_2 = 1$ **then output** *1;*
  if $b_1 = 1$ **and** $b_2 = 0$ **then output** *0;*
end

Conditioned on any step of the algorithm outputting a bit, it's easy to see that the bit is uniformly random. This is because $\Pr[b_1 = 0, b_2 = 1] = \Pr[b_1 = 1, b_2 = 0]$. *The probability of any step outputting a bit is* $2p(1-p) = 2(p-p^2)$. *Therefore the probability that this process doesn't terminate after* n *steps is* $(1-2(p-p^2))^n$.

Looking at the problem more generally, assume we have a random variable X that we want to take one sample from it, e.g. converting a password to a uniform encryption key. The goal is to design an extractor Ext such that for every random variable X (with some arbitrary distribution), $\mathsf{Ext}(X)$ is uniformly random.

It's clear that we need to put some restrictions on X; for example if X is always 0 then $\mathsf{Ext}(X)$ is some fixed constant and not uniformly random. Intuitively, we need to require that X has some sufficient level of *entropy*. It turns out that Shannon entropy does not suit our needs because it measures how many bits of randomness are contained in X on average. For example, think of a random variable X that half the times outputs zero and half the time outputs 1,000 random bits. The Shannon entropy of this variable is high (500 bits) but this randomness is not good for cryptographic purposes since half the time it's a fixed value.

We will instead use a measure of entropy called *min-entropy*. DEFINITION 3 The min-entropy of a random variable X is

$$\mathbf{H}_\infty(X) = -\log\left(\max_x \Pr[X = x]\right)$$

$\diamondsuit$ For example if the X is uniformly random over some set of size 2^k then $\mathbf{H}_\infty(()X) = k$.

Ideally, we would want an extractor that works for all random variables X, as long as the min-entropy of X is above some threshold k. For example, we would like to have an extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ such that for every random variable X over $\{0,1\}^n$ with $\mathbf{H}_\infty(X) \geq k$, $\mathsf{Ext}(X)$ is close to uniform.

It turns out that this is still impossible, even if $k = n-1$ (i.e., even if X has almost full entropy). For any candidate extractor Ext as above let $b \in \{0,1\}$ be the value that maximizes $|\mathsf{Ext}^{-1}(b)|$ in which case $|\mathsf{Ext}^{-1}(b)| \geq 2^{n-1}$. Let X be uniformly random over $\mathsf{Ext}^{-1}(b)$. Then $\mathbf{H}_\infty(X) \geq n-1$ but $\mathsf{Ext}(X)$ is always equal to the constant b and therefore is not random.

The previous example shows that if we choose the distribution after we pick the extractor Ext, an adversary can take advantage of which distribution to pick to make the extractor fail. Therefore, we will put some randomness into selecting the Ext. In particular, we consider the notion of a *seeded* extractor $\mathsf{Ext}(X, s)$ which is parametrized by a seed s; each s defines a different extractor. We will essentially require that the extractor is good if s is chosen randomly, but made public. In other words, we need to invest some uniform randomness into selecting the seed s, but then we get a good return on the investment by deriving additional randomness from the source X.

DEFINITION 4 A function $\mathsf{Ext} : \mathcal{U} \times \mathcal{S} \to \mathcal{V}$ is a (k, ε)-extractor if for all r.v. X over $\mathcal{U}$ such that $\mathbf{H}_\infty(X) \geq k$, we have that

$$\mathsf{SD}\left((S, \mathsf{Ext}(X, S)), (S, V)\right) \leq \varepsilon$$

where S is uniform over $\mathcal{S}$ and V is uniform over $\mathcal{V}$. The choice of S is called the seed. $\diamondsuit$

Usually, we will want $\mathcal{V} = \{0,1\}^\ell$ so that the extractor outputs uniformly random bits.

4 Constructing Extractors: Leftover Hash Lemma

4.1 Universal hash functions

DEFINITION 5 A function $H : \mathcal{U} \times \mathcal{S} \to \mathcal{V}$ is a universal hash function if for all $x, x' \in \mathcal{U}$ with $x \neq x'$ it holds that

$$\Pr[H(x, S) = H(x', S)] = \frac{1}{|\mathcal{V}|}$$

where S is a random seed over $\mathcal{S}$. ♢

The only randomness of the above definition, comes from the seed S. An example of a universal hash function is the following.

Example 2 *Consider a field $\mathbb{F}$, and set $\mathcal{U} = \mathbb{F}^n$, $\mathcal{V} = \mathbb{F}$ and $\mathcal{S} = \mathbb{F}^n$. The function $H(\vec{x}, \vec{s}) = \langle \vec{x}, \vec{s} \rangle$ is a universal hash function where $\langle \cdot, \cdot \rangle$ is the inner product, $\langle \vec{x}, \vec{s} \rangle = \sum_{i=1}^{n} x_i \cdot s_i$.*

To prove that, say that two arbitrary $\vec{x} \neq \vec{x}' \in \mathcal{U}$ differ in the i-th position, i.e. $x_i \neq x'_i$. We have that,

$$\begin{aligned} \Pr[\langle \vec{x}, \vec{S} \rangle = \langle \vec{x}', \vec{S} \rangle] &= \Pr[S_i = (x_i - x'_i)^{-1} \sum_{j \neq i} (x'_j - x_j) S_j] \\ &= \frac{1}{|\mathbb{F}|} \end{aligned}$$

Where the second line holds over a random choice of S_i over $\mathbb{F}$ even for any fixed choice of all other values S_j : $j \neq i$.

4.2 Leftover hash lemma

The next theorem shows that universal hash functions are good extractors.

Theorem 2 ([ILL89]) *A universal hash function is a (k, ε)-extractor for*

$$k \geq \ell + 2 \log \left(\frac{1}{\varepsilon} \right) - 2$$

where $\ell = \log_2(|\mathcal{V}|)$.

In other words, this means that to extract ℓ uniformly random bits, we need the source X to have entropy $k \geq \ell + 2\log(1/\varepsilon) - 2$. We can think of $2\log(1/\varepsilon) - 2$ as the entropy loss - we need this extra randomness to extract, but we can't output it. It's known that all extractors must have this much entropy loss, and so the above is tight.

To prove the theorem, we first prove the following claim.

Claim 3 *Let Z be a r.v. over $\mathcal{W}$ such that*

$$\mathsf{Col}(Z) := \sum_{z \in \mathcal{W}} \Pr[Z = z]^2 \leq \frac{1}{|\mathcal{W}|}(1 + 4\varepsilon^2).$$

Then, $\mathsf{SD}(Z, W) \leq \varepsilon$ where W is uniform on $\mathcal{W}$.

Note: we call $\mathsf{Col}(Z)$ the collision probability. If Z, Z' are identically and independently distributed then $\mathsf{Col}(Z) = \Pr[Z = Z']$. In other words, this measures the probability of two independent samples from Z colliding.

Proof:(Of Claim 3) We have that

$$\mathsf{SD}(Z, W) = \frac{1}{2} \sum_z \left| \Pr[Z = z] - \frac{1}{|\mathcal{W}|} \right|$$

Set $q_z = \Pr[Z = z] - \frac{1}{|\mathcal{W}|}$. We define the next sign function in order to remove the absolute value.

$$s_z = \begin{cases} 1 & \text{if } q_z \geq 0 \\ -1 & else. \end{cases}$$

Therefore,

$$\begin{aligned} \mathsf{SD}(Z, W) &= \frac{1}{2} \sum_z q_z s_z \\ &= \frac{1}{2} \langle \vec{q}, \vec{s} \rangle \\ &\leq \frac{1}{2} \sqrt{\langle \vec{q}, \vec{q} \rangle \langle \vec{s}, \vec{s} \rangle} \quad \text{(Cauchy-Schwarz inequality)} \\ &= \frac{1}{2} \sqrt{\sum_z q_z^2 \, |\mathcal{W}|} \end{aligned}$$

Analyzing the term $\sum_z q_z^2$ we have that

$$\begin{aligned} \sum_z q_z^2 &= \sum_z \left(\Pr[Z = z]^2 - \frac{2\Pr[Z = z]}{|\mathcal{W}|} + \frac{1}{|\mathcal{W}|^2} \right) \\ &\leq \frac{1}{|\mathcal{W}|}(1 + 4\varepsilon^2) - \frac{2 \cdot 1}{|\mathcal{W}|} + \frac{|\mathcal{W}|}{|\mathcal{W}|^2} \\ &= \frac{4\varepsilon^2}{|\mathcal{W}|} \end{aligned}$$

Using the above inequality, we obtain the following to complete this proof

$$\begin{aligned} \mathsf{SD}(Z, W) &\leq \frac{1}{2} \sqrt{\sum_z q_z^2 \, |\mathcal{W}|} \\ &\leq \frac{1}{2} \sqrt{\sum_z \frac{4\varepsilon^2}{|\mathcal{W}|} \, |\mathcal{W}|} \\ &= \frac{1}{2} \sqrt{4\varepsilon^2} \\ &= \varepsilon \end{aligned}$$

□

Bibliography

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random Generation from One-way Functions. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 12–24, New York, NY, USA, 1989. ACM.