Introduction to Cryptography

CSCI-GA.3210-001 Instructor: Nir Bitansky

Lecture 3: Computational Indistinguishability

1 Previously on Introduction to Crypto

In Lecture 2, we saw that computational hardness is necessary for encryption with short keys. We then gave a definition for computational security of encryption, taking an asymptotic approach. We then gave a contruction of encryption for messages of any polynomial length $\ell'(n)$ from an encryption scheme (E, D) for messages of length $n+1$ with keys of length n. For a message $x = x_1 \cdots x_{\ell'}$, the new scheme samples independent uniform keys $sk_1, \ldots, sk_{\ell'} \leftarrow \{0,1\}^n$, sets $sk_0 = sk$, and outputs

$$E'_{sk}(x) = \left(E_{sk_0}(sk_1, x_1),\ E_{sk_1}(sk_2, x_2),\ \ldots,\ E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'})\right) \ .$$

Our informal "fake proof of security" replaced the encrypted messages by 0^{n+1} one ciphertext at a time, pretending that the underlying scheme was perfectly secret. This introduced the idea of a hybrid argument, but requires proper adaptation to the computational setting. Today we will develop the language needed to make this formal.

2 Computational Indistinguishability

Intuitively, we wish to generalize the notion of distributions being identical to that of their being close in the eyes of computationally bounded adversaries. Before we go all the way and do that, it is useful to first generalize equality to *statistical closeness*: the case where distributions are close not only in the eyes of bounded observers but also in the eyes of unbounded ones.

Definition 2.1 (Statistical Indistinguishability)**.** *Two distributions X, Y over the same finite support S are ε-statistically indistinguishable if, for any (unbounded) distinguisher $A : S \to \{0,1\}$:*

$$\Delta_A(X,Y) := \left|\Pr\left[A(X)=1\right] - \Pr\left[A(Y)=1\right]\right| \leq \varepsilon \ .$$

The quantity $\Delta(X,Y) := \max_A \Delta_A(X,Y)$ is the statistical distance between the distributions.

In what sense is this the statistical distance?

Claim 2.2.

$$\Delta(X,Y) = \frac{1}{2}\|X - Y\|_1 := \frac{1}{2}\sum_x \left|\Pr\left[X=x\right] - \Pr\left[Y=x\right]\right| \ .$$

We can connect this definition to the intuitions developed from previous definitions:

Claim 2.3. *For any X_0, X_1, A:*

$$\left|\Pr\left[A(x)=b \;\middle|\; \begin{array}{c} b \leftarrow \{0,1\} \\ x \leftarrow X_b \end{array}\right] - \frac{1}{2}\right| = \frac{\Delta_A(X_0,X_1)}{2} \ .$$

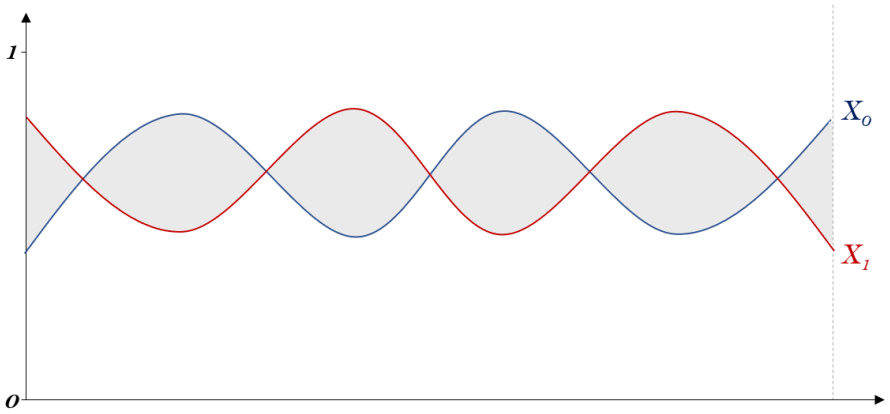

Figure 1: Illustration - The statistical distance between the two (continuous) distributions X_0, X_1 is the area between the graphs of the two density functions.

Claim 2.4 (Triangle Inequality)**.** $\Delta_A(X,Y) \leq \Delta_A(X,Z) + \Delta_A(Z,Y)$.

We can now define our notion of computational closeness, commonly called *computational indistinguishability*, which is a natural analog of statistical indistinguishability. Sticking with the asymptotic approach, it is not meaningful to talk about computational indistinguishability (against all efficient attackers) for two finite distributions. Accordingly, we consider an asymptotic notion of *distribution ensembles*.

A distribution ensemble $X = \{X_i\}_{i \in I_n, n \in \mathbb{N}}$ is an infinite collection of distributions associated with some infinite collection $I = \uplus I_n$ of disjoint index sets. Intuitively, n represents the security parameter (the size of the secret key), and I_n represents some additional parameters that define the distribution.

Example: The ensemble of encryptions of length-$\ell(n)$ messages

$$\{E_{sk}(m) \mid sk \leftarrow \{0,1\}^n\}_{m \in \{0,1\}^{\ell(n)}, n \in \mathbb{N}} \ .$$

Here, for every message m, we have a distribution over encryptions of that message.

Definition 2.5 (Computational Indistinguishability)**.** *Two ensembles of distributions $X = \{X_i\}_{i \in I_n, n \in \mathbb{N}}$ and $Y = \{Y_i\}_{i \in I_n, n \in \mathbb{N}}$ over the same set of indices $I = \uplus I_n$ are computationally indistinguishable if, for any non-uniform PPT $A = \{A_n\}_{n \in \mathbb{N}}$, there exists a negligible function $\mu(n)$ such that for all $n \in \mathbb{N}$ and every $i \in I_n$:*

$$\Delta_{A_n}(X_i, Y_i) := |\Pr[A_n(X_i) = 1] - \Pr[A_n(Y_i) = 1]| \leq \mu(n) \ .$$

We denote this by $X \approx_c Y$.

Remark (Abusing Notation). One convenient abuse of notation is to write:

$$\text{For any } n \in \mathbb{N} \text{ and any } i \in I_n,\ X_i \approx_c Y_i.$$

Taken literally, this is not formally meaningful, since X_i and Y_i are distributions, whereas $\approx_c$ is an asymptotic relation between distribution ensembles. What we actually mean here is that the ensembles X and Y are computationally indistinguishable. Once you feel comfortable with the notion of computational indistinguishability, you can use whichever notation you prefer.

Claim 2.6 (Transitivity)**.** *If $X \approx_c Z$ and $Z \approx_c Y$ then $X \approx_c Y$.*

We can also think asymptotically about statistical distance. We will say that two ensembles X, Y are statistically indistinguishable if the preceding definition also holds for unbounded A, and we denote this by $X \approx_s Y$.

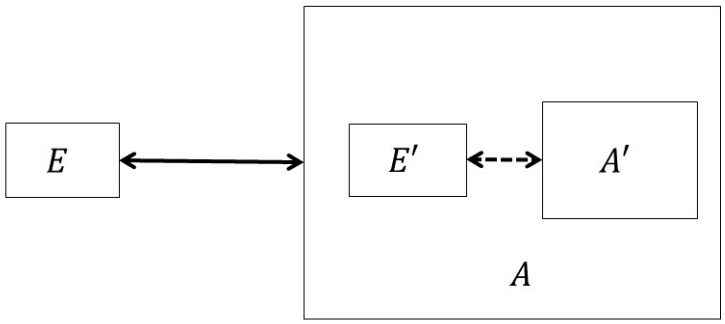

Figure 2: An efficient A' against E' can be translated into an efficient A against E; A translates an "E-type" ciphertext into an "E'-type" ciphertext.

Proving the Security of Our Encryption. We are now ready to give a real (rather than fake) proof that our scheme E' is computationally secure if the original scheme E is computationally secure. To this end, we would like to prove that the encryptions of any two messages are computationally indistinguishable. Formally, we want to show that:

$$\{E'_{sk}(x) \mid sk \leftarrow \{0,1\}^n\}_{\substack{n\in\mathbb{N} \\ x,y\in\{0,1\}^{\ell'(n)}}} \approx_c \{E'_{sk}(y) \mid sk \leftarrow \{0,1\}^n\}_{\substack{n\in\mathbb{N} \\ x,y\in\{0,1\}^{\ell'(n)}}}$$

You're probably still not used to this notation, so let's make sure we can make sense of it. It says that for any n.u. PPT A, there exists a negligible function μ such that for any $n \in \mathbb{N}$ and any $x, y \in \{0,1\}^{\ell'(n)}$:

$$\Delta_{A_n}(E'_{sk}(x), E'_{sk}(y)) \leq \mu(n) \ .$$

To prove the above, we will actually show that there exists a single distribution ensemble $S = \{S_n\}_n$ such that encryptions of different messages are all computationally indistinguishable from this distribution:

$$\{E'_{sk}(x) \mid sk \leftarrow \{0,1\}^n\}_{\substack{n\in\mathbb{N} \\ x\in\{0,1\}^{\ell'(n)}}} \approx_c \{S_n\}_{\substack{n\in\mathbb{N} \\ x\in\{0,1\}^{\ell'(n)}}} \ .$$

This is actually an equivalent definition for secure encryption, similar to the one we had for perfect security, and we can prove it using the fact that computational indistinguishability is transitive:

$$\{E'_{sk}(x) \mid sk \leftarrow \{0,1\}^n\}_{\substack{n\in\mathbb{N} \\ x,y\in\{0,1\}^{\ell'(n)}}} \approx_c \{S_n\}_{\substack{n\in\mathbb{N} \\ x,y\in\{0,1\}^{\ell'(n)}}} \approx_c \{E'_{sk}(y) \mid sk \leftarrow \{0,1\}^n\}_{\substack{n\in\mathbb{N} \\ x,y\in\{0,1\}^{\ell'(n)}}} \ .$$

We define $S = \{S_n\}$ as follows:

$$S_n \quad := \quad E_{sk=sk_0}(0^{n+1}) \quad E_{sk_1}(0^{n+1}) \quad \ldots \quad E_{sk_{\ell'-1}}(0^{n+1})$$

The Security Reduction. To prove that $S \approx_c \{E'_{sk}(x)\}_{n,x}$, we will show that any efficient adversary A' against E', namely one that distinuishes the above two ensembles, can be translated into an efficient adversary A against E. This is what we call *a security reduction*; it is the bread and butter of provably secure crypto, and we'll do it all the time in this course (see Figure 2).

Assume, toward a contradiction, that there is a n.u. PPT attacker $A' = \{A_n\}_n$ and a polynomial $p(\cdot)$ such that, for infinitely many n, there exists $x \in \{0,1\}^{\ell'(n)}$ such that:

$$\Delta_{A'_n}(E'_{sk}(x), S_n) = |\Pr[A'_n(E'_{sk}(x)) = 1] - \Pr[A'(S_n) = 1]| \geq 1/p(n) \ .$$

We will turn it into a non-uniform PPT attacker A that (for the same infinitely many n) will break (E, D).

Fix such n and x. For every $0 \leq i \leq \ell'$, we define the following hybrid distribution:

$$H_i := E_{sk_0}(0^{n+1}), \ldots, E_{sk_{i-1}}(0^{n+1}), E_{sk_i}(sk_{i+1}, x_{i+1}), \ldots, E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'}) \ .$$

According to this definition, we have:

$$\begin{array}{cccccccccc}
 & H_0 & = & E_{sk_0}(sk_1, x_1) & E_{sk_1}(sk_2, x_2) & \ldots & & & & E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'}) \\
 & H_1 & = & E_{sk_0}(0^{n+1}) & E_{sk_1}(sk_2, x_2) & \ldots & & & & E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'}) \\
 & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
\hline
i-1 & H_{i-1} & = & E_{sk_0}(0^{n+1}) & E_{sk_1}(0^{n+1}) & \ldots & \boldsymbol{E_{sk_{i-1}}(sk_i, x_i)} & E_{sk_i}(sk_{i+1}, x_{i+1}) & \ldots & E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'}) \\
i & H_i & = & E_{sk_0}(0^{n+1}) & E_{sk_1}(0^{n+1}) & \ldots & \boldsymbol{E_{sk_{i-1}}(0^{n+1})} & E_{sk_i}(sk_{i+1}, x_{i+1}) & \ldots & E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'}) \\
\hline
 & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
 & H_{\ell'} & = & E_{sk_0}(0^{n+1}) & E_{sk_1}(0^{n+1}) & \ldots & & & & E_{sk_{\ell'-1}}(0^{n+1})
\end{array}$$

Note that $H_0 = E'_{sk=sk_0}(x)$ and $H_{\ell'} = S_n$.

Claim 2.7. *There exists $i \in [\ell']$ such that $\Delta_{A'}(H_{i-1}, H_i) \geq \frac{1}{p(n) \cdot \ell'(n)}$.*

Proof.

$$\frac{1}{p(n)} \leq \Delta_{A'}(E'_{sk}(x), S_n) = \Delta_{A'}(H_0, H_{\ell'}) \leq \sum_{i \in [\ell']} \Delta_{A'}(H_{i-1}, H_i) \leq \max_{i \in [\ell']} \Delta_{A'}(H_{i-1}, H_i) \cdot \ell'(n) \ ,$$

where the first inequality follows from the definition of A' and the second inequality from the triangle inequality. □

Notice that H_{i-1} and H_i differ only in the ith encryption:

- In H_{i-1}, we have $E_{sk_{i-1}}(sk_i, x_i)$,
- In H_i, we have $E_{sk_{i-1}}(0^{n+1})$.

We now need to construct an adversary A that breaks E. More precisely, we will show that for a random $sk_i \leftarrow \{0,1\}^n$

$$\Delta_A((E_{sk}(sk_i, x_i), sk_i), (E_{sk}(0^{n+1}), sk_i)) \geq \frac{1}{\ell'(n) \cdot p(n)} \ .$$

(Make sure you understand why this is enough.)

Adversary A: Given a ciphertext ct and sk_i, A will generate all the ciphertexts except the ith one itself and insert ct as the ith ciphertext, as follows:

$$\underbrace{E_{sk_0}(0^{n+1})}_{ct_1} \quad \underbrace{E_{sk_1}(0^{n+1})}_{ct_2} \quad \ldots \quad \underbrace{E_{sk_{i-2}}(0^{n+1})}_{ct_{i-1}} \quad \underbrace{ct}_{ct_i} \quad \underbrace{E_{sk_i}(sk_{i+1}, x_{i+1})}_{ct_{i+1}} \quad \ldots \quad \underbrace{E_{sk_{\ell'-1}}(sk_{\ell'}, x_{\ell'})}_{ct_{\ell'}}$$

It will then run A' on this sample and output the same bit.

Note that if $ct \leftarrow E_{sk}(sk_i, x_i)$, then the sample is distributed exactly like H_{i-1}, and if $ct \leftarrow E_{sk}(0^{n+1})$, then it is distributed exactly like H_i. Accordingly, we have:

$$\Delta_A((E_{sk}(sk_i, x_i), sk_i), (E_{sk}(0^{n+1}), sk_i)) = \Delta_{A'}(H_{i-1}, H_i) \geq \frac{1}{\ell'(n) \cdot p(n)} \ .$$

Crucially, in both hybrids, $ct_1, \ldots, ct_{i-1}, ct_{i+1}, \ldots, ct_{\ell'}$ are completely independent of sk_{i-1} and can be generated efficiently. (Notice that to sample ct_{i+1}, we use the input sk_i.) □

Note on Using Non-Uniformity in the Reduction. In the above reduction, we've implicitly relied on non-uniformity. That is, even if A' were uniform, the A that we constructed would be non-uniform — where does A get the value i for which A' has an advantage? Non-uniformity! This use of non-uniformity is not really inherent. You can show, for instance, that if A chooses i at random, the reduction still works. Alternatively, you can show that A can efficiently (and uniformly) find i by sampling and estimating the advantage for each i.