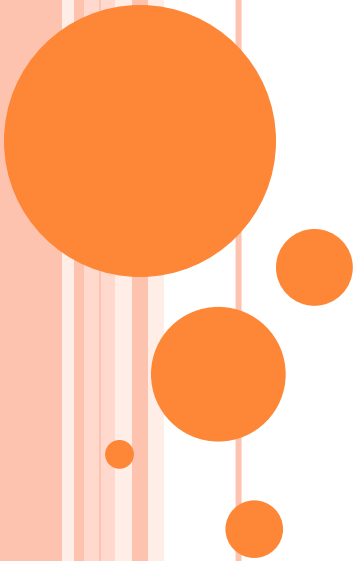
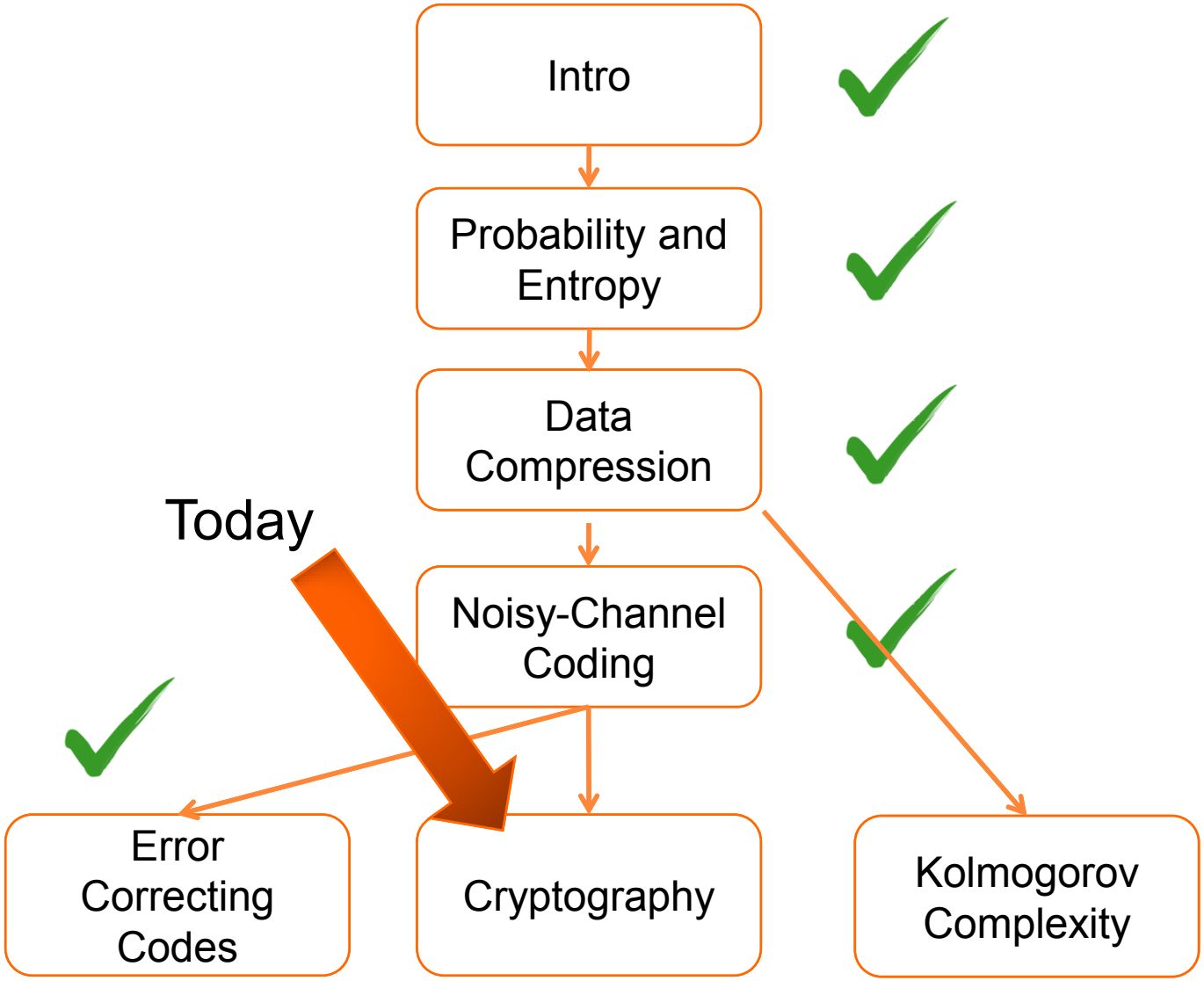


CS3236 Introduction to Information theory

Lecture 11: Introduction to Cryptography

Course given by Pierre Senellart
Material by Stephanie Wehner, with additions by P. Senellart

Where do we go from here?



What we'll do today

- Use of randomness in cryptography
- Entropy in cryptography
- Randomness extraction
  - Idea
  - Measure of success
  - Essential concepts

Example: sending messages securely

- Goal: The eavesdropper (Eve) should not gain any information about the message by listening to the communication.

Example: One time pad encryption

$$
\begin{array}{cccccc}
m_1 & m_2 & m_3 & \dots & m_n \\
\oplus & \oplus & \oplus & \dots & \oplus \\
k_1 & k_2 & k_3 & \dots & k_n \\
= & = & = & \dots & = \\
t_1 & t_2 & t_3 & \dots & t_n
\end{array}
$$

$$m_j, k_j, t_j \in \{0, 1\} \qquad m_j \oplus k_j = m_j + k_j \mod 2$$

What does it mean that Eve doesn't know the key?

- Key should be "random"

$$P_K(k = k_1, \ldots, k_n) = \frac{1}{2^n}$$

What does it mean that Eve doesn't know the key?

- Key should be "random"

$$P_K(k = k_1, \ldots, k_n) = \frac{1}{2^n}$$

- Eve is uncorrelated. For Eve's side information E

$$P_{KE}(k, e) = P_K(k) \times P_E(e)$$

What does it mean that Eve doesn't know the key?

- Key should be "random"

$$P_K(k = k_1, \ldots, k_n) = \frac{1}{2^n}$$

- Eve is uncorrelated. For Eve's side information E

$$P_{KE}(k, e) = P_K(k) \times P_E(e)$$

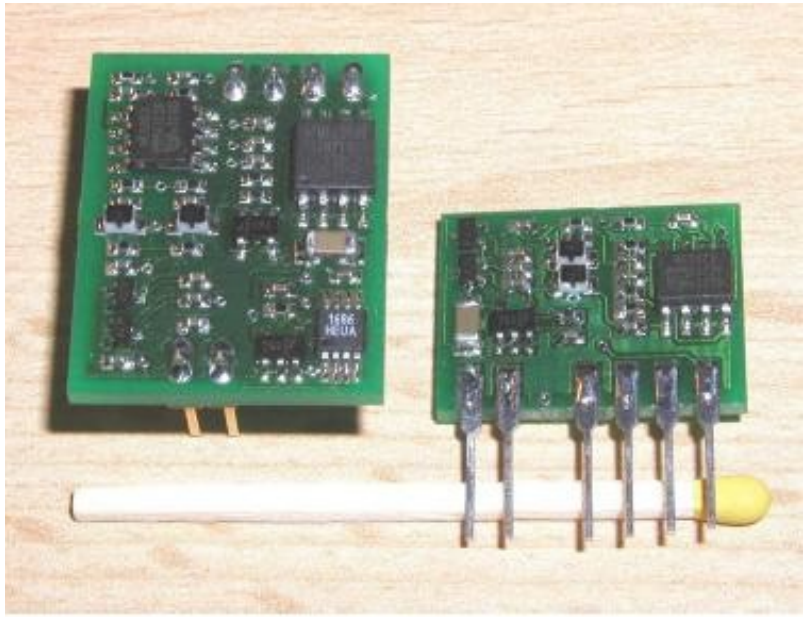
- Together we want that

$$P_{KE}(k, e) = \frac{1}{2^n} \times P_E(e)$$

How do we obtain such randomness?

- Any key exchange protocol will
  - Need some source of randomness
  - Protect this randomness from Eve

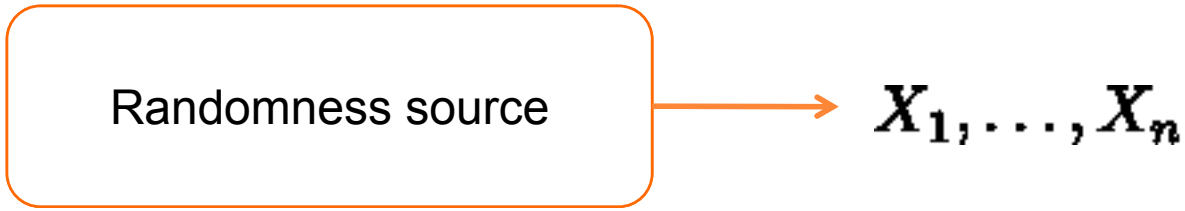
“Guaranteed” randomness is hard to come by…



PRG210 PRG220

How do we obtain such randomness?

Imagine a key generation machine



How can we quantify how much Eve knows about the output?

IDEA: USING ENTROPY

- We know that for the uniform distribution $H(X) = n$
- If Eve is uncorrelated $H(X) = H(X|E)$

- How about a somewhat imperfect source?

$$H(X) \geq \frac{n}{2}$$

Shannon entropy as a cryptographic measure

$$P_X(x) = \begin{cases} \frac{1}{2} & x = 111\ldots1 \\ \frac{1}{2} \cdot \frac{1}{2^n - 1} & \text{otherwise} \end{cases}$$

$$\downarrow$$

$$X_1, \ldots, X_n$$

- How useful is this box for generating keys?

Min-Entropy

- Entropic measure used in cryptography

$$H_{\min}(X) = -\log \max_x \Pr_X(x) = \min_x (-\log \Pr_X(x))$$

- Measures how well Eve can guess x

$$H_{\min}(X \mid E) = -\log \sum_e \Pr_E(e) \max_x \Pr_{X \mid E}(x \mid E = e)$$
$$= -\log \Pr_{\text{guess}}(X \mid E)$$

Compare entropies for our example

$$P_X(x) = \begin{cases} \frac{1}{2} & x = 111\ldots1 \\ \frac{1}{2} \cdot \frac{1}{2^n - 1} & \text{otherwise} \end{cases}$$

$$X_1, \ldots, X_n$$

$$H(X) \approx \frac{n}{2}$$

$$H_{\min}(X) = 1$$

Compare entropies for our example

$$P_X(x) = \begin{cases} \frac{1}{2} & x = 111\ldots1 \\ \frac{1}{2} \cdot \frac{1}{2^n - 1} & \text{otherwise} \end{cases}$$

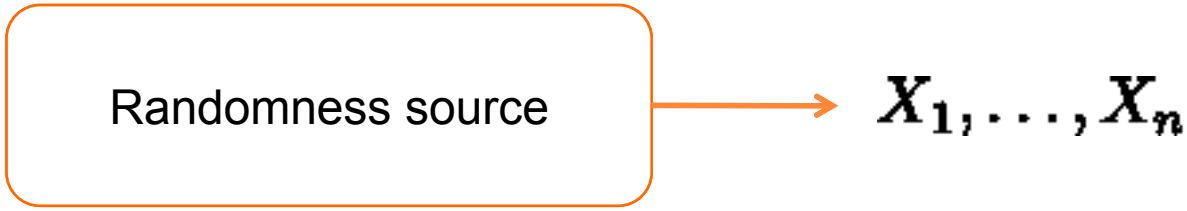
$$X_1, \ldots, X_n$$

$$H(X) \approx \frac{n}{2} \qquad H_{\min}(X) = 1$$

The min entropy is used to quantify how useful randomness is for the use in a cryptographic protocol

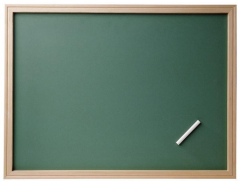
Generating randomness from imperfect sources

- Weak source of randomness: does not generate bits according to the uniform distribution



- Example: i.i.d. source $P_{X_j}(0) = \frac{1}{3}\ P_{X_j}(1) = \frac{2}{3}$

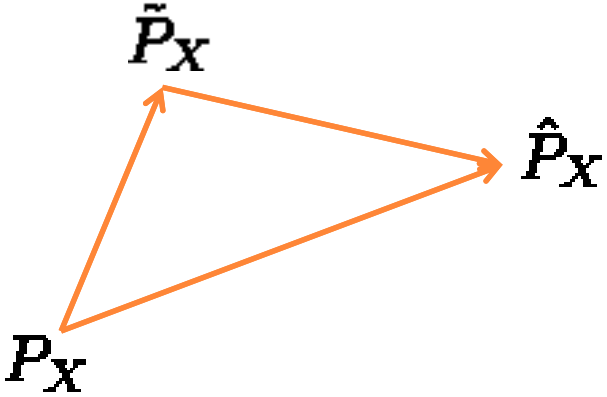
How can we produce one bit that is **more** random? (i.e., closer to 50:50?)



Measure of success

- How do we measure “closer to uniform”?
- The statistical distance (or *total variation distance*) compares two distributions

$$\Delta(P_X, \hat{P}_X) = \frac{1}{2} \sum_x \left| P_X(x) - \hat{P}_X(x) \right|$$



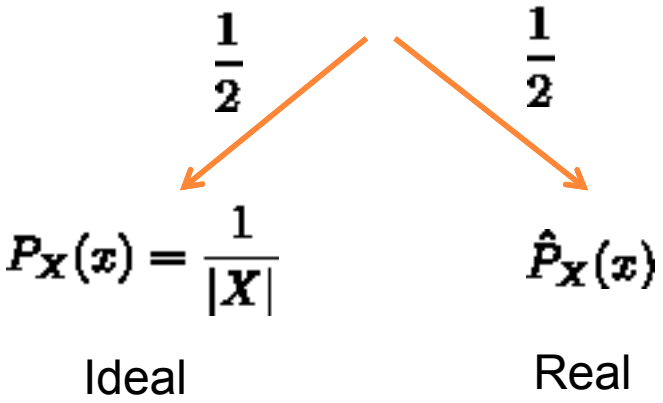
Properties of distance measure:

1. $\Delta(P_X, \hat{P}_X) \geq 0$
2. $\Delta(P_X, \hat{P}_X) = 0 \rightarrow P_X = \hat{P}_X$
3. $\Delta(P_X, \hat{P}_X) = \Delta(\hat{P}_X, P_X)$
4. $\Delta(P_X, \hat{P}_X) \leq \Delta(P_X, \bar{P}_X) + \Delta(\bar{P}_X, \hat{P}_X)$

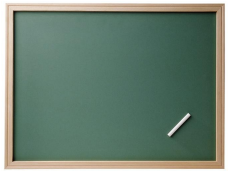
Property of the statistical distance

- Definition $$\Delta(P_X, \hat{P}_X) = \frac{1}{2} \sum_x \left| P_X(x) - \hat{P}_X(x) \right|$$

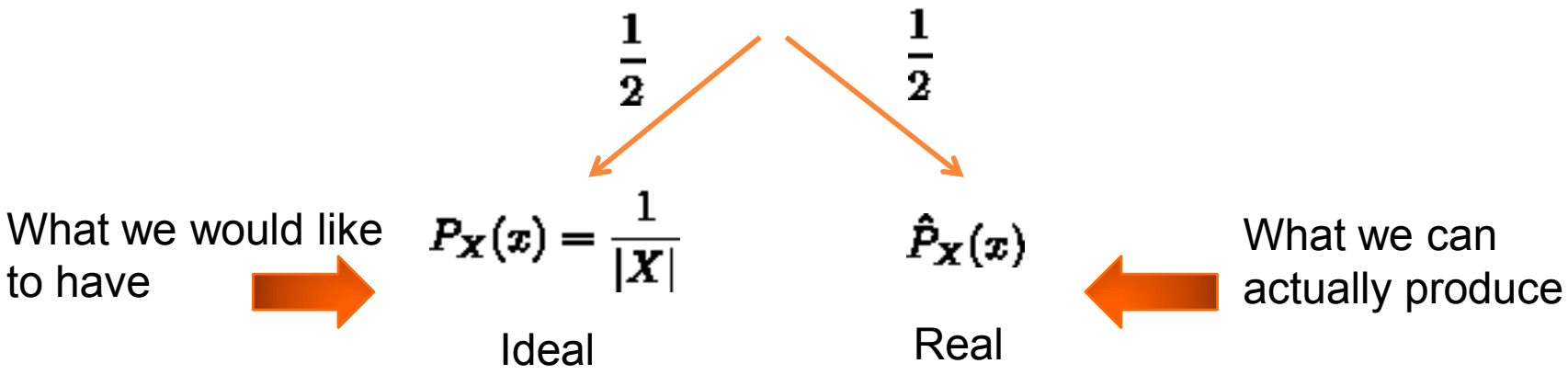
- Operational meaning: determines how well we can distinguish two distributions



Let's show that $$P_{\text{distinguish}} \leq \frac{1}{2} + \frac{\Delta(P_X, \hat{P}_X)}{2}$$



Significance for cryptography



- When using randomness generated by the “real” process, we will assume in further analysis that we replaced the “real” by the “ideal” process.
- If the statistical distance is small, then the error thus incurred is small - otherwise the subsequent protocol would allow us to tell real from ideal but we know:

$$P_{\text{distinguish}} \leq \frac{1}{2} + \frac{\Delta(P_X, \hat{P}_X)}{2}$$

Goal of a randomness extractor

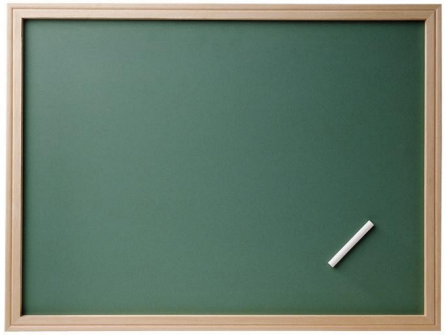
- Given a weak source of randomness $X_1, \ldots, X_n$, produce a (possibly shorter) string $K_1, \ldots, K_\ell$

$$\Delta(P_K, \text{unif}(|K|)) \leq \epsilon$$

uniform distribution

acceptable error

- For our example: $\ell = 1, |K| = 2$



Types of random sources

- I.I.D. Source (von Neumann source)

$$P_X(x) = P_{X_1}(x_1) \dots P_{X_1}(x_n)$$

- Independent bit source

$$P_X(x) = P_{X_1}(x_1) \dots P_{X_n}(x_n)$$

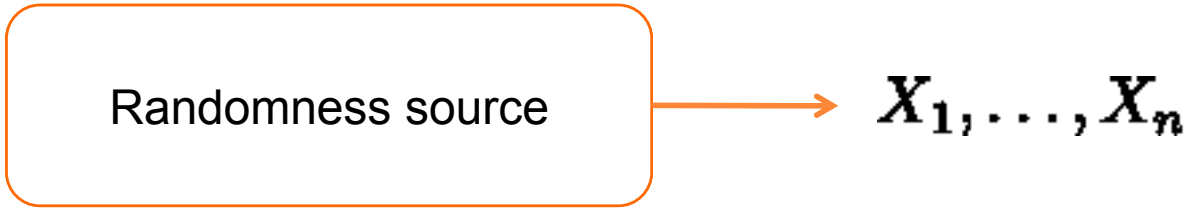
- k-source

$$H_{\min}(X) \geq k$$

$$H_{\min}(X|E) \geq k$$

Deterministic randomness extractor

- In our example we applied a fixed function (parity)



- A deterministic randomness extractor is a function

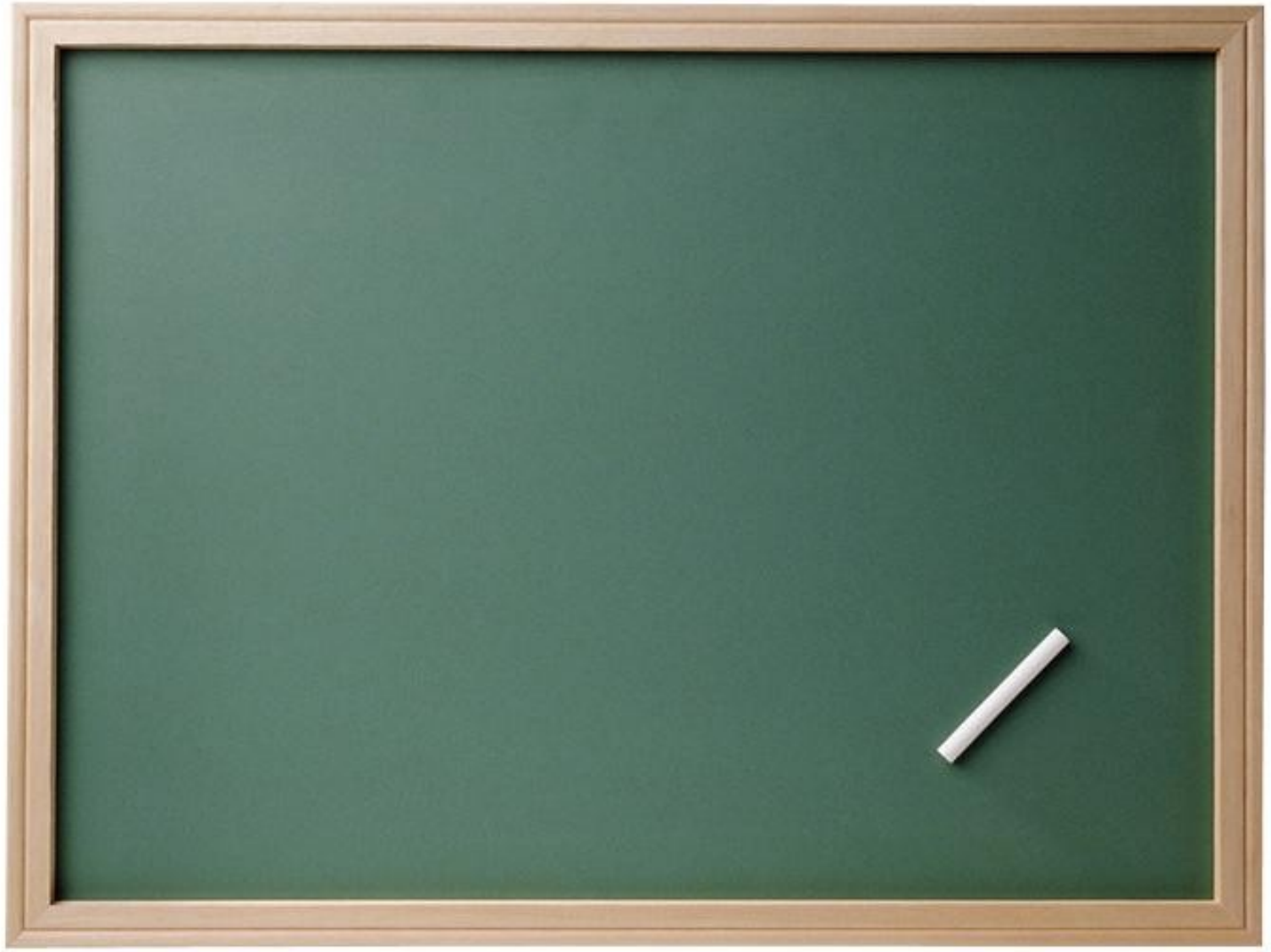
$$\mathrm{Ext} : \{0,1\}^n \to \{0,1\}^\ell$$

- Is deterministic randomness extraction always possible?

Impossibility

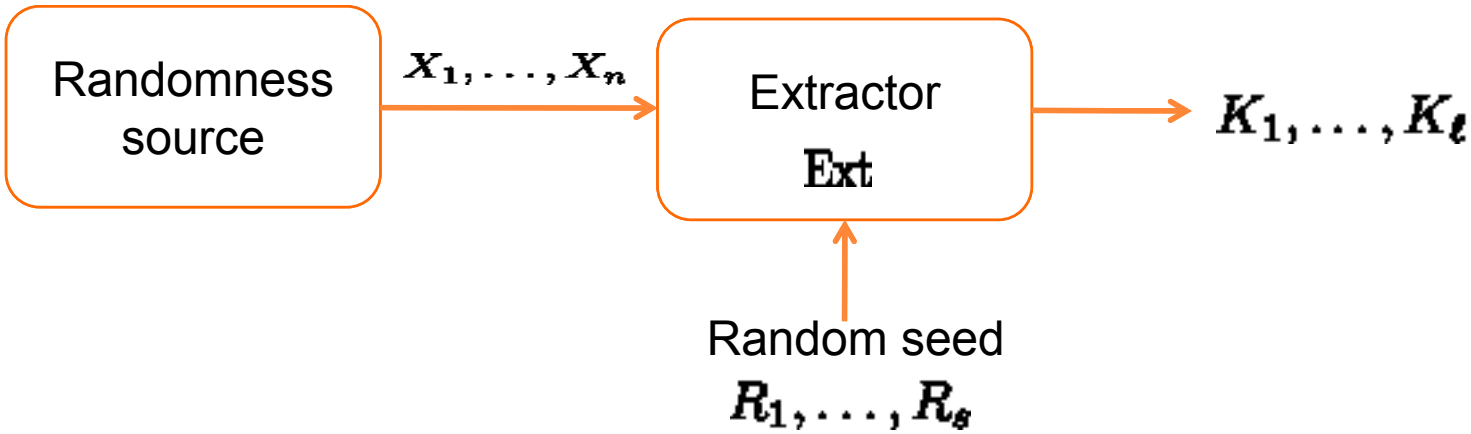
- There is no deterministic randomness extractor for an arbitrary k-source!
- Even if $H_{\min}(X) = n - 1$
- We show that for any function $\mathbf{Ext}$ there exists a source (probability distribution) such that we can predict the outcome perfectly!

Proof of impossibility of a perfect deterministic extractor



NOT ALL IS LOST…..

- Investing a random seed



- Can extract randomness this way!

- …. hardly surprising?

Weak vs. strong extractors

- Weak (k, ϵ)–extractor: If the source has min-entropy $H_{\min}(X) = k$ the output bits obey

$$\Delta(P_K, \mathrm{unif}(|K|)) \leq \epsilon$$

$$\Delta(P_{KE}, \mathrm{unif}(|K|) \times P_E) \leq \epsilon$$

- Strong (k, ϵ)–extractor: If the source has min-entropy $H_{\min}(X) = k$ the output bits obey

$$\Delta(P_{KER}, \mathrm{unif}(|K|) \times P_{RE}) \leq \epsilon$$

Uniform (ie hard to predict) even given the random seed R!

How much randomness can we generate?

- Depends on
  - Seed size that we invest
  - Min-entropy of the source
- There exists extractors such that
  - Seed size $\log(n) - 2\log(1/\epsilon) + O(1)$
  - Output size $k - 2\log(1/\epsilon) + O(1)$
- The output size can never exceed the initial min-entropy

$$\ell \leq k = H_{\min}(X|E)$$

Constructing strong extractors

- Many constructions exist!
- Randomness extractors are equivalent to
  - List-decodable codes
  - Certain expander graphs
- A simple example: 2-universal hashing

$$F = \{f : \{0,1\}^n \to \{0,1\}^\ell\}$$
$$\forall x \neq x', \ \Pr_{f \in F}[f(x) = f(x')] \leq \frac{1}{2^\ell}$$

- The set of all possible functions is 2-universal ☺

Leftover hash lemma

- Imagliazzo, Levin and Luby
  - (Informal) Choosing f at random from a set of 2-universal hash functions yields a (k, ϵ)–strong randomness extractor with output length

$$\ell = k - 2\log(1/\epsilon)$$

- Significant, since it tells us that we can actually generate roughly $k = H_{\min}(X|E)$ bits of randomness!

What we did today

- Cryptographic measure of entropy: min-entropy

$$H_{\min}(X) = -\log \max_x P_X(x)$$

- Randomness extractor
  - Weak and strong extractors
  - Measure of success

$$\Delta(P_{KER}, \mathrm{unif}(|K|) \times P_{RE}) \leq \epsilon$$

  - Basic example

Reading for this lecture

- Lecture notes on statistical distance, min-entropy, and randomness extraction from IVLE (LN1statDistance.pdf, LN2minEntropy.pdf, LN3Randomness.pdf). Ignore references to homework and to quantum information.

- Homework due by Monday 2pm next week