Advanced Tools from Modern Cryptography

Lecture 1

Basics: Indistinguishability

Manoj Prabhakaran

IIT Bombay

Outline

- Independence
- Statistical Indistinguishability
- Computational Indistinguishability

A Game

- A "dealer" and two "players" Alice and Bob (computationally unbounded)
- Dealer has a message, say two bits m_1m_2
- She wants to "share" it among the two players so that neither player by herself/himself learns <u>anything</u> about the message, but together they can find it
- Bad idea: Give m_1 to Alice and m_2 to Bob
- Other ideas?

Sharing a bit

- To share a bit m, Dealer picks a uniformly random bit b and gives $a := m \oplus b$ to Alice and b to Bob
  - Together they can recover m as $a \oplus b$
  - Each party by itself learns nothing about m: for each possible value of m, its share has the same distribution
  - i.e., Each party's "view" is independent of the message

$a = \text{Share}_A(m;r) = m \oplus r$

$b = \text{Share}_B(m;r) = r$.

$m = 0 \rightarrow (a,b) = (0,0)$ or $(1,1)$ w.p. 1/2 each

$m = 1 \rightarrow (a,b) = (1,0)$ or $(0,1)$ w.p. 1/2 each

Secrecy

- Is the message m really secret?
- Alice or Bob can correctly find the bit m with probability ½, by randomly guessing
  - Worse, if they already know something about m, they can do better (Note: we didn't say m is uniformly random!)
- But they could have done this without obtaining the shares
  - The shares didn't leak any additional information to either party
- Typical crypto goal: preserving secrecy
  - What Alice (or Bob) knows about the message after seeing her share is the same as what she knew a priori

Secrecy

- What Alice knows about the message a priori: probability distribution over the message
  - For each message m, Pr[msg=m]
- What she knows after seeing her share (a.k.a. her view)
  - Say view is v. Then new distribution: Pr[msg=m | view=v]
- Secrecy: ∀ v, ∀ m, Pr[msg=m | view = v] = Pr[msg = m]
  - i.e., view is independent of message
  - Equivalently, ∀ v, ∀ m, Pr[view=v | msg=m] = Pr[view=v]
  - i.e., for all possible values of the message, the view is distributed the same way
  - i.e., $\forall\ m_1, m_2 \ \ \{\ \mathrm{Share}_A(m_1;r)\ \}_r \equiv \{\ \mathrm{Share}_A(m_2;r)\ \}_r$

Secrecy

> Doesn't involve message distribution at all.

- Equivalent formulations:
  - For all possible values of the message, the view is distributed the same way
    - $\forall$ v, $\forall m_1, m_2$, Pr[view=v | msg=m_1] = Pr[view=v | msg=m_2]
  - View and message are independent of each other
    - $\forall$ v, $\forall$ m, Pr[msg=m, view = v] = Pr[msg = m] × Pr[view = v]
  - View gives no information about the message
    - $\forall$ v, $\forall$ m, Pr[msg=m | view=v] = Pr[msg = m]

> Require a message distribution (with full support)

- Important: can't say Pr[msg=m_1 | view=v] = Pr[msg=m_2 | view=v] (unless the prior is uniform)

Exercise

- Consider the following secret-sharing scheme
  - Message space = { Jan, Feb, Mar }
  - Jan $\longrightarrow$ (00,00), (01,01), (10,10) or (11,11) w/ prob 1/4 each
  - Feb $\longrightarrow$ (00,01), (01,00), (10,11) or (11,10) w/ prob 1/4 each
  - Mar $\longrightarrow$ (00,10), (01,11), (10,00), (11,01), (00,11), (01,10), (10,01) or (11,00) w/ prob 1/8 each
  - Reconstruction: Let $\beta_1\beta_2$ = $\text{share}_{\text{Alice}} \oplus \text{share}_{\text{Bob}}$. Map $\beta_1\beta_2$ as follows: 00 $\longrightarrow$ Jan, 01 $\longrightarrow$ Feb, 10 or 11 $\longrightarrow$ Mar
- Is it secure?

Onetime Encryption

The Syntax

- Shared-key (Private-key) Encryption
  - **Key Generation**: Randomized
    - $K \leftarrow \mathcal{K}$, uniformly randomly drawn from the key-space (or according to a key-distribution)
  - **Encryption**: Deterministic
    - Enc: $\mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$
  - **Decryption**: Deterministic
    - Dec: $\mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$

Needs randomisation for more-than-once encryption

Onetime Encryption

Perfect Secrecy

- **Perfect secrecy:** $\forall$ m, m′ $\in \mathcal{M}$
  - $\{Enc(m,K)\}_{K \leftarrow KeyGen} = \{Enc(m',K)\}_{K \leftarrow KeyGen}$

Distribution of the ciphertext

- … defined by the randomness in the key
- In addition, require **correctness**
  - $\forall$ m, K,   Dec(Enc(m,K), K) = m
- E.g. One-time pad: $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^n$ and Enc(m,K) = m⊕K, Dec(c,K) = c⊕K
  - More generally $\mathcal{M} = \mathcal{K} = \mathcal{C} = \mathcal{G}$ (a finite group) and Enc(m,K) = m+K, Dec(c,K) = c–K

$\mathcal{M}$ \ $\mathcal{K}$	0	1	2	3
a	x	y	y	z
b	y	x	z	y

Assuming K uniformly drawn from $\mathcal{K}$

Pr[Enc(a,K)=x] = ¼,
Pr[Enc(a,K)=y] = ½,
Pr[Enc(a,K)=z] = ¼

Same for Enc(b,K).

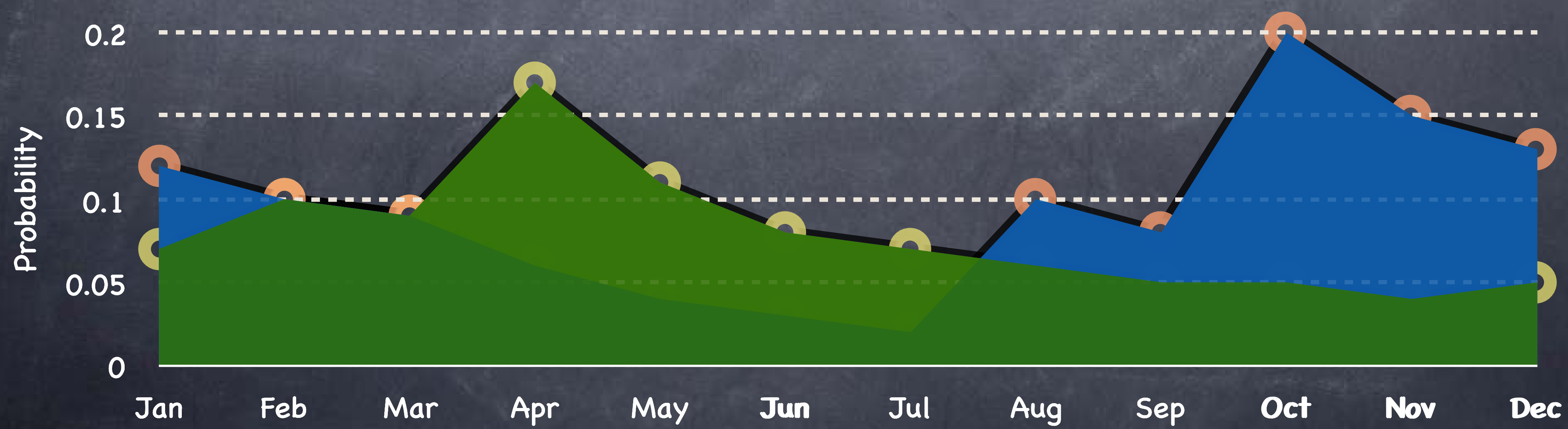
Relaxing Secrecy Requirement

- When view is not exactly independent of the message
  - Next best: view close to a distribution that is independent of the message
  - Two notions of closeness: Statistical and Computational

a.k.a. Statistical Distance or Total Variation Distance

Statistical Difference

- Given two distributions A and B over the same sample space, how well can a test T distinguish between them?
  - T given a single sample drawn from A or B
  - How differently does it behave in the two cases?
- $\Delta(A,B) := \max_T \mid Pr_{x \leftarrow A}[T(x)=1] - Pr_{x \leftarrow B}[T(x)=1] \mid$



Indistinguishability

- Two distributions are **statistically indistinguishable** from each other if the statistical difference between them is "negligible"
- What is negligible? 2^{-20} ? 2^{-40} ? 2^{-80} ? Let the "user" decide!
- Security guarantees will be given asymptotically as a function of the **security parameter**
  - A knob that can be used to set the security level
- Given $\{A_k\}$, $\{B_k\}$, $\Delta(A_k,B_k)$ is a function of the security parameter k
- Negligible: reduces "very quickly" as the knob is turned up
  - "Very quickly": quicker than 1/poly for any polynomial poly
    - So that if negligible for one sample, remains negligible for polynomially many samples
  - $\nu(k)$ is said to be **negligible** if $\forall\ d \geq 0, \exists\ N$ s.t. $\forall\ k>N, \nu(k) < 1/k^d$

Indistinguishability

- Distribution ensembles $\{A_k\}$, $\{B_k\}$ are **statistically indistinguishable** if $\exists$ negligible $\nu(k)$ s.t. $\Delta(A_k,B_k) \leq \nu(k)$
  - $\Delta(A_k,B_k) := \max_T \mid Pr_{x \leftarrow A_k}[T(x)=1] - Pr_{x \leftarrow B_k}[T(x)=1] \mid$
- Can rewrite as: $\forall$ tests T, $\exists$ negligible $\nu(k)$ s.t.

  $\mid Pr_{x \leftarrow A_k}[T_k(x)=1] - Pr_{x \leftarrow B_k}[T_k(x)=1] \mid \leq \nu(k)$

In particular,
T that is best for all k.
(k is also given to T)

- Distribution ensembles $\{A_k\}$, $\{B_k\}$ **computationally indistinguishable** if $\forall$ "efficient" tests T, $\exists$ negligible $\nu(k)$ s.t.

  $\mid Pr_{x \leftarrow A_k}[T_k(x)=1] - Pr_{x \leftarrow B_k}[T_k(x)=1] \mid \leq \nu(k)$

Really need to allow a
different ν for each T

Indistinguishability

- Distribution ensembles $\{A_k\}$, $\{B_k\}$ **computationally indistinguishable** if $\forall$ "efficient" tests T, $\exists$ negligible $\nu(k)$ s.t.

  $$| \Pr_{x \leftarrow A_k}[T_k(x)=1] - \Pr_{x \leftarrow B_k}[T_k(x)=1] | \leq \nu(k)$$

  $A_k \approx B_k$

- **Efficient:** Probabilistic Polynomial Time (PPT)
  - PPT T: a family of randomised programs T_k (one for each value of the security parameter k), s.t. there is a polynomial p with each T_k running for at most p(k) time

    Non-Uniform

  - (Could restrict to uniform PPT, i.e., a single program which takes k as an additional input. But by default, we'll allow non-uniform.)