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 $m_1 m_2$
- She wants to "share" it among the two players so that neither player by herself/himself learns anything 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⊕b to Alice and b to Bob
  - Together they can recover m as a⊕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

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

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

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

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

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: $\forall$ v, $\forall$ m, Pr[msg=m | view = v] = Pr[msg = m]
  - i.e., view is independent of message
  - Equivalently, $\forall$ v, $\forall$ 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 $\{ \text{Share}_A(m_1;r) \}_r \equiv \{ \text{Share}_A(m_2;r) \}_r$

Doesn't involve message distribution at all.

Secrecy

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

Doesn't involve message distribution at all.

Require a message distribution (with full support)

Exercise

- Consider the following secret-sharing scheme
  - Message space = { Jan, Feb, Mar }
  - Jan → (00,00), (01,01), (10,10) or (11,11) w/ prob 1/4 each
  - Feb → (00,01), (01,00), (10,11) or (11,10) w/ prob 1/4 each
  - Mar → (00,10), (01,11), (10,00), (11,01), (00,11), (01,10), (10,01) or (11,00) w/ prob 1/8 each
  - Reconstruction possible as the 3 sets of shares are disjoint
    - Let $\beta_1\beta_2 = \text{share}_{\text{Alice}} \oplus \text{share}_{\text{Bob}}$. Map $\beta_1\beta_2$ as follows: 00 → Jan, 01 → Feb, 10 or 11 → Mar
- Is it secure?

Homework!

A Puzzle

- Alice and Bob hold secret numbers x and y in {0,..,n} resp.
- Carol wants to learn x+y. Alice and Bob are OK with that.
- But they don't want Carol/each other to learn anything else!
  - i.e., Alice should learn nothing about y, nor Bob about x. Carol shouldn't learn anything else about x,y "other than" x+y
- Can they do it, just by talking to each other (using private channels between every pair of parties)?

How would you formalise this?

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

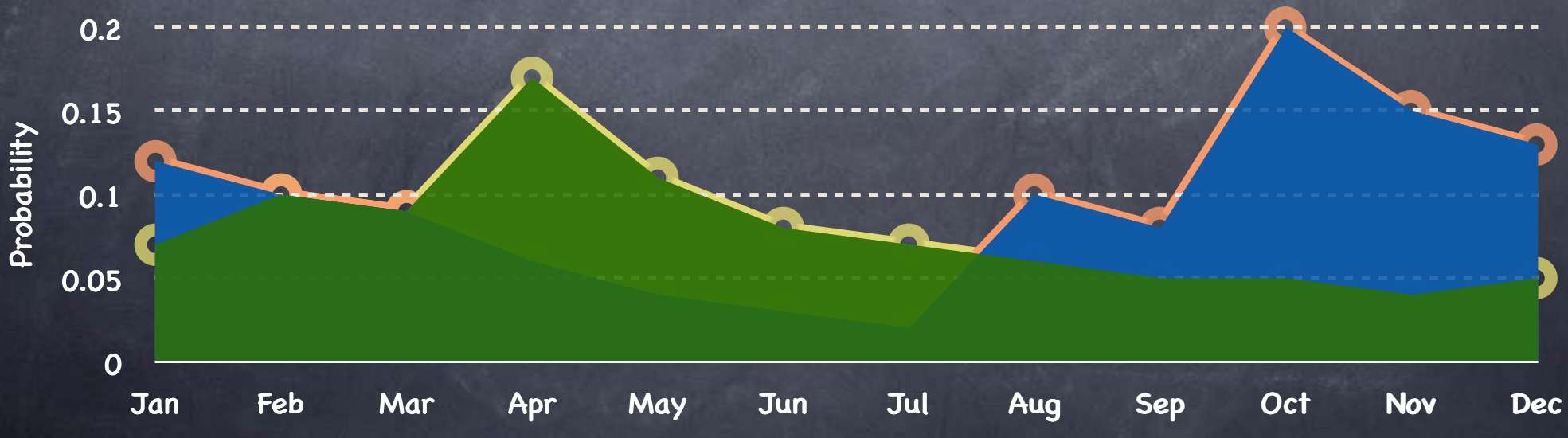
Statistical Difference

- 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?
- $\Delta(A,B) := \max_T | Pr_{x \leftarrow A}[T(x)=0] - Pr_{x \leftarrow B}[T(x)=0] |$
- $Pr_{x \leftarrow A}[T(x)=0] - Pr_{x \leftarrow B}[T(x)=0] = \sum_x (A(x)-B(x))p(x)$, where p(x) stands for Pr[T(x)=0], and A(x), B(x)
- Maximised when p(x)=1 for A(x)>B(x) and p(x)=0 for A(x)<B(x)
- Equals $\sum_{x:A(x)>B(x)} A(x)-B(x) = \sum_{x:A(x)<B(x)} B(x)-A(x) = \frac{1}{2} \sum_x |A(x)-B(x)|$

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?
- $\Delta(A,B) := \max_T | \Pr_{x \leftarrow A}[T(x)=0] - \Pr_{x \leftarrow B}[T(x)=0] |$



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 ν s.t. $\forall k$ $\Delta(A_k,B_k) \leq \nu(k)$
  - where $\Delta(A_k,B_k) := \max_T | Pr_{x \leftarrow A_k}[T(x)=0] - Pr_{x \leftarrow B_k}[T(x)=0] |$
  - i.e. if $\exists$ negligible ν s.t. $\forall$ tests T, $\forall k$
    $| Pr_{x \leftarrow A_k}[T_k(x)=0] - Pr_{x \leftarrow B_k}[T_k(x)=0] | \leq \nu(k)$
  - Equivalently (why?) $\forall$ tests T, $\exists$ negligible ν s.t $\forall k$
    $| Pr_{x \leftarrow A_k}[T_k(x)=0] - Pr_{x \leftarrow B_k}[T_k(x)=0] | \leq \nu(k)$
- Distribution ensembles $\{A_k\}$, $\{B_k\}$ **computationally indistinguishable** if $\forall$ "efficient" tests T, $\exists$ negligible ν s.t.
  $| Pr_{x \leftarrow A_k}[T_k(x)=0] - Pr_{x \leftarrow B_k}[T_k(x)=0] | \leq \nu(k)$

Indistinguishability

$A_k \approx B_k$

- Distribution ensembles $\{A_k\}$, $\{B_k\}$ computationally indistinguishable if $\forall$ "efficient" tests T, $\exists$ negligible ν s.t.

  $| \Pr_{x \leftarrow A_k}[T_k(x)=0] - \Pr_{x \leftarrow B_k}[T_k(x)=0] | \leq \nu(k)$

"$\exists \nu \, \forall T$" makes it as strong as statistical indistinguishability

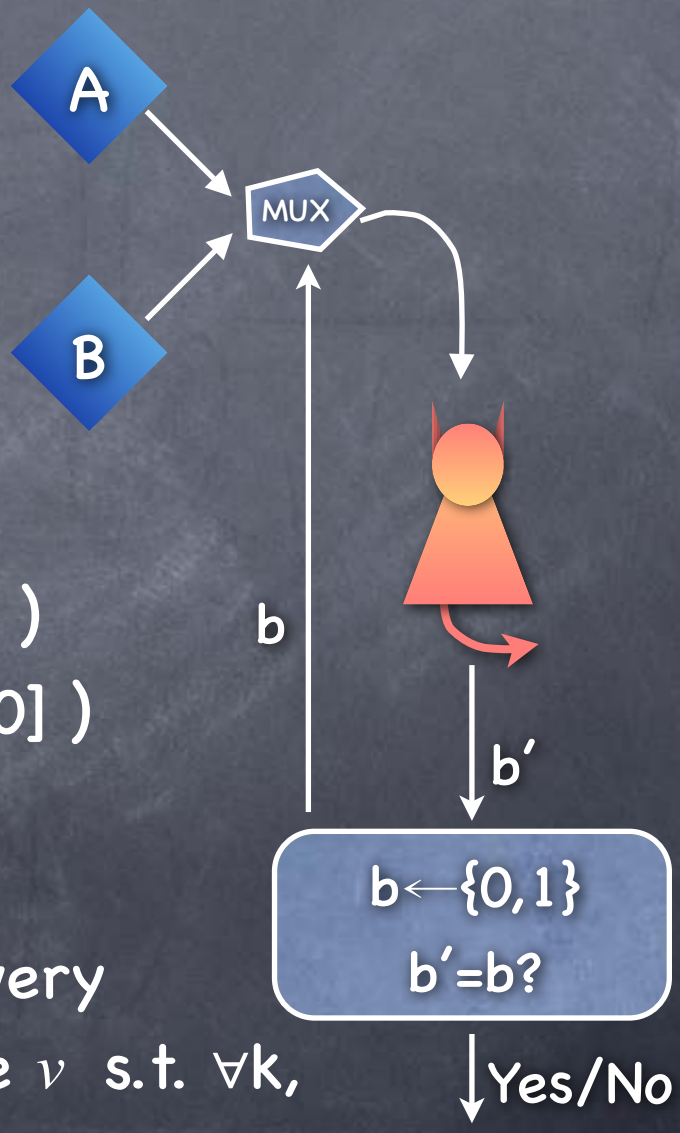
- Efficient: Probabilistic Polynomial Time (PPT)

Non-Uniform

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

Security Games

- Indistinguishability can be defined using a guessing game
  - b chosen uniformly at random
  - Pr[b′=b] = ?
    - Pr[b′=b=0] + Pr[b′=b=1]
      = ½·Pr[b′=0|b=0] + ½·Pr[b′=1|b=1]
      = ½ (Pr[b′=0|b=0] + 1−Pr[b′=0|b=1])
      = ½ + ½ (Pr[b′=0|b=0] − Pr[b′=0|b=1])
      = ½ + ½ ($Pr_{x \leftarrow A}$[T(x)=0] − $Pr_{x \leftarrow B}$[T(x)=0])
  - Maximum Pr[b′=b] = ½ + Δ(A,B)/2
  - A,B computationally indistinguishable if, for every PPT adversary in the above game, ∃ negligible ν s.t. ∀k,
    Advantage(k) := Pr[b′=b] − ½ ≤ ν(k)



Pseudorandomness Generator (PRG)

- Takes a short seed and (deterministically) outputs a long string
  - G_k: $\{0,1\}^k \to \{0,1\}^{n(k)}$ where $n(k) > k$
- Security definition: Output distribution induced by random input seed should be "pseudorandom"
  - i.e., Computationally indistinguishable from uniformly random
  - $\{G_k(x)\}_{x \leftarrow \{0,1\}^k} \approx U_{n(k)}$
  - Note: $\{G_k(x)\}_{x \leftarrow \{0,1\}^k}$ cannot be statistically indistinguishable from $U_{n(k)}$ unless $n(k) \leq k$ (Exercise)
    - i.e., no non-trivial PRG against unbounded adversaries