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 <u>random</u> bit b and gives $a := m \oplus b$ to Alice and b to Bob

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

  - 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

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

  - i.e., Each party's "view" is <u>independent</u> of the message

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., $\forall\ m_1,m_2\ \ \{\ \mathrm{Share}_A(m_1;r)\ \}_r \equiv \{\ \mathrm{Share}_A(m_2;r)\ \}_r$

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] $\times$ Pr[view = v]
  - View gives no information about the message
    - $\forall$ v, $\forall$ m, Pr[msg=m | view=v] = Pr[msg = m]

Doesn't involve message distribution at all.

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 $\rightarrow$ (00,00), (01,01), (10,10) or (11,11) w/ prob 1/4 each
  - Feb $\rightarrow$ (00,01), (01,00), (10,11) or (11,10) w/ prob 1/4 each
  - Mar $\rightarrow$ (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 $\rightarrow$ Jan, 01 $\rightarrow$ Feb, 10 or 11 $\rightarrow$ Mar
- Is it secure?

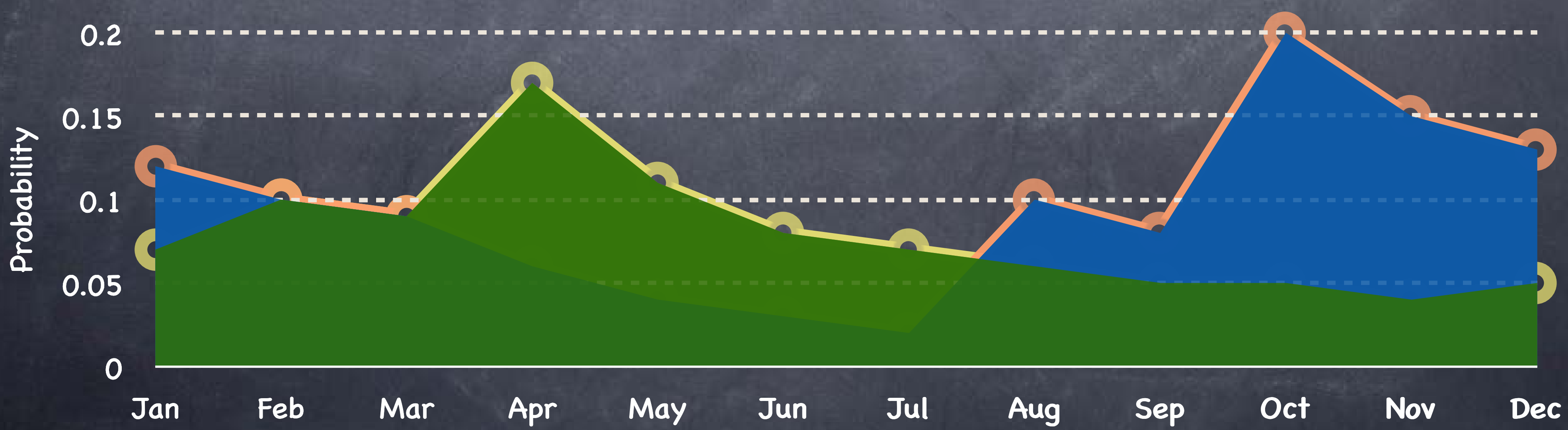
Relaxing Secrecy Requirement

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



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 ν s.t. $\forall k$ $\Delta(A_k,B_k) \leq \nu(k)$
  - where $\Delta(A_k,B_k) := \max_T \mid \Pr_{x\leftarrow A_k}[T(x)=0] - \Pr_{x\leftarrow B_k}[T(x)=0] \mid$
  - i.e. if $\exists$ negligible ν s.t. $\forall$ tests T, $\forall k$

    $\mid \Pr_{x\leftarrow A_k}[T_k(x)=0] - \Pr_{x\leftarrow B_k}[T_k(x)=0] \mid \leq \nu(k)$
- Distribution ensembles $\{A_k\}$, $\{B_k\}$ **computationally indistinguishable** if $\exists$ negligible ν s.t. $\forall$ "efficient" tests T, $\forall$ sufficiently large k

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

Asking for $\forall k$ makes it as strong as statistical indistinguishability

Indistinguishability

$A_k \approx B_k$

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

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

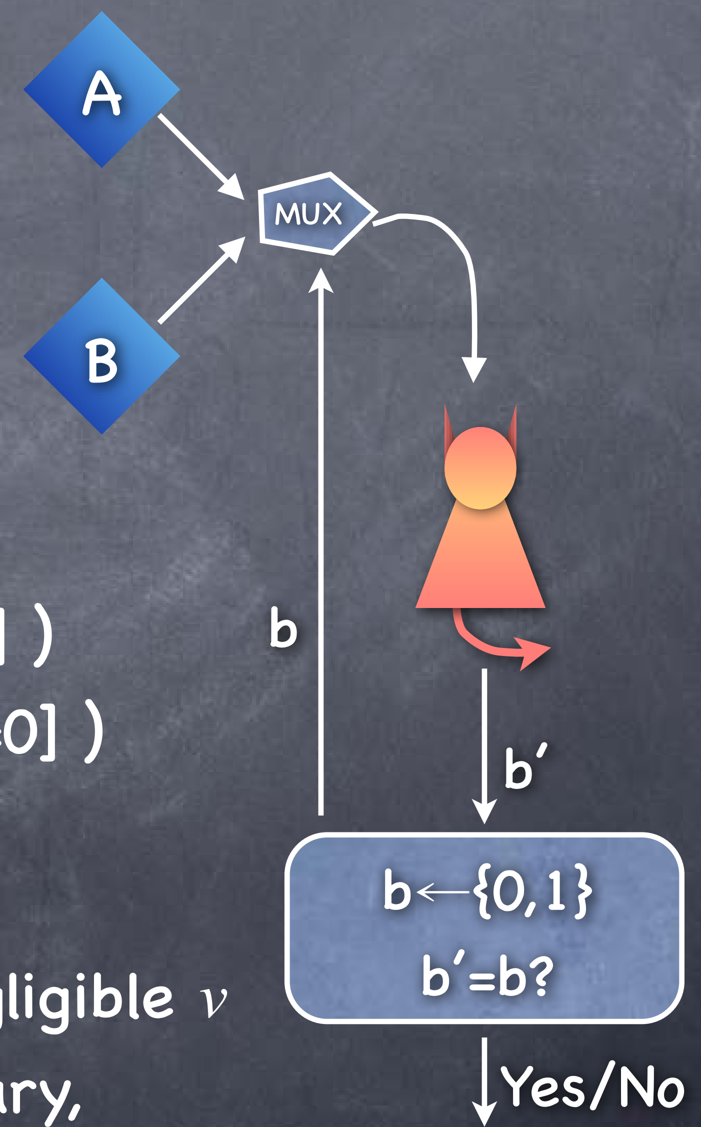
- Efficient: Probabilistic Polynomial Time (PPT)

  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
  - $\Pr[b'=b] = ?$
    - $\Pr[b'=b=0] + \Pr[b'=b=1]$
      $= \tfrac{1}{2}\cdot\Pr[b'=0|b=0] + \tfrac{1}{2}\cdot\Pr[b'=1|b=1]$
      $= \tfrac{1}{2}\left(\Pr[b'=0|b=0] + 1-\Pr[b'=0|b=1]\right)$
      $= \tfrac{1}{2} + \tfrac{1}{2}\left(\Pr[b'=0|b=0] - \Pr[b'=0|b=1]\right)$
      $= \tfrac{1}{2} + \tfrac{1}{2}\left(\Pr_{x\leftarrow A}[T(x)=0] - \Pr_{x\leftarrow B}[T(x)=0]\right)$
  - Maximum $\Pr[b'=b] = \tfrac{1}{2} + \Delta(A,B)/2$
  - A,B computationally statistically indistinguishable if $\exists$ negligible ν large enough in the above game, for every PPT adversary, $\forall k$, Advantage(k) $:= \Pr[b'=b] - \tfrac{1}{2} \le \nu(k)$



Pseudorandomness Generator (PRG)

- Takes a short seed and (deterministically) outputs a long string
  - G_k: $\{0,1\}^k \rightarrow \{0,1\}^{n(k)}$ where $n(k) > k$
- Security definition: Output distribution induced by random input seed should be "pseudorandom"
  - i.e., Computationally indistinguishable from uniformly random
  - $\boxed{\{G_k(x)\}_{x \leftarrow \{0,1\}^k} \approx U_{n(k)}}$
  - 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