18734 Recitation

Statistical Distance
Basic Cryptography

Amit Datta
Distance

• L1 distance

• Between two points
  \(- (x_1, x_2, ..., x_n) \text{ and } (y_1, y_2, ..., y_n)\)

\[- \sum_i |x_i - y_i|\]
Distance between functions

• Between two discrete functions
  – \( m_1(x), m_2(x) \)
  – \( x \in \{x_1, x_2, ..., x_n\} \)
  – \( \sum_i |m_1(x_i) - m_2(x_i)| \)

• Between two continuous functions
  – \( n_1(y), n_2(y) \)
  – \( y \in [y_1, y_2] \)
  – \( _{y_1}^{y_2} \int |n_1(y) - n_2(y)| \, dy \)
Distance between probability distributions

• Between two discrete distributions
  – PMFs $p_1(x)$, $p_2(x)$
  – $x \in \{x_1, x_2, ..., x_n\}$
  – $\sum_i |p_1(x_i) - p_2(x_i)|$

• Between two continuous distributions
  – PDFs $f_1(y)$, $f_2(y)$
  – $y \in [y_1, y_2]$  
  – $\int_{y_1}^{y_2} |f_1(y) - f_2(y)| \, dy$
Exercise

• Find L1 distance between the following distributions:
  
  \(- f_1(x) = \frac{x}{12} \quad x \in [1, 5] \)
  
  \(- f_2(x) = \frac{1}{4} \quad x \in [1, 5] \)
Fairness through Awareness

Metric \( d : V \times V \rightarrow \mathbb{R} \)

Lipschitz condition \( \| M(x) - M(y) \| \leq d(x, y) \)

\( M : V \rightarrow \Delta(O) \)

\( V: \text{Individuals} \quad O: \text{outcomes} \)
Fairness through Awareness: Example

Santa is distributing blue and red candies
Basic Crypto Concepts
Basic Cryptographic Concepts

• Encryption scheme (symmetric and public key)
• Signature scheme
• Message authentication code
• Hash function

• A network protocol like SSL is built using these primitives
Symmetric Encryption Scheme

• **Key generation algorithm**  
  – Produces a key that is used for encryption and decryption

• Algorithm to *encrypt* a message

• Algorithm to *decrypt* a ciphertext

• **Correctness:**  
  – Decrypting a ciphertext obtained by encrypting message $m$ with the corresponding key $k$ returns $m$
    \[
    dec(enc(m,k),k) = m
    \]

• **(Symbolic) Security:**  
  – A ciphertext cannot be decrypted without access to the key
Symmetric Encryption Scheme

• **Key generation algorithm**
  – generate random bits

• **Algorithm to encrypt a message**
  – \( \text{enc}(m,k) = m \oplus k \)

• **Algorithm to decrypt a ciphertext**
  – \( \text{dec}(c,k) = c \oplus k \)

• **Correctness:**
  – \( \text{dec}(\text{enc}(m,k),k) = m \). Satisfied?

• **(Symbolic) Security:**
  – A ciphertext cannot be decrypted without access to the key. Satisfied?
Public-Key Encryption Scheme

- **Key generation algorithm**
  - Produces private decryption & public encryption key pair

- Algorithm to *encrypt* a message

- Algorithm to *decrypt* a ciphertext

- **Correctness:**
  - Decrypting a ciphertext obtained by encrypting message $m$ with the corresponding encryption key returns $m$
    
    $\text{dec(}\text{enc}(m, pk(A)), sk(A)) = m$

- **(Symbolic) Security:**
  - A ciphertext cannot be decrypted without access to the private decryption key
Public-Key Encryption Scheme

Hello Alice!  \rightarrow  Encrypt  \rightarrow  Alice's public key  

6EB6957008E03CE4

Hello Alice!  \rightarrow  Decrypt  \rightarrow  Alice's private key
Public-Key Encryption Scheme

- **Key generation algorithm**
  - Generate random public key: e, secret key: d=1/e

- **Algorithm to encrypt a message**
  - \( \text{enc}(m,e) = m^e \)

- **Algorithm to decrypt a ciphertext**
  - \( \text{dec}(c,d) = c^d \)

- **Correctness:**
  - \( \text{dec(\text{enc}(m, pk(A)), sk(A))} = m. \) Satisfied?

- **(Symbolic) Security:**
  - A ciphertext cannot be decrypted without access to the private decryption key. Satisfied?

*Why would we want public-key encryption?*

Signature Scheme

• *Key generation* algorithm
  – Produces private signing & public verification key pair

• Algorithm to *sign* data

• Algorithm to *verify* signature

• Correctness:
  – Message signed with a signing key verifies with the corresponding verification key
    \[ \text{verify}(m, \text{sign}(m, sk(A)), pk(A)) = \text{ok} \]

• Security:
  – A signature cannot be produced without access to the private signing key
Signature Scheme

• Key generation algorithm
  – private signing & public verification key pair (e, d=1/e)

• Algorithm to sign data
  – $\text{sign}(m, e) = m^e$

• Algorithm to verify signature
  – $\text{verify}(m, c, d) = \text{return ok iff } m == c^d$

• Correctness:
  – $\text{verify}(m, \text{sign}(m, sk(A)), pk(A)) = ok$. Satisfied?

• Security:
  – A signature cannot be produced without access to the private signing key. Satisfied?
Message Authentication Code (MAC)

- **Key generation algorithm**
  - Produces a key

- **Algorithm to mac a message**

- **Algorithm to verify a mac on a message**

- **Correctness:**
  - Message mac-ed with key verifies with the same key
    \[\text{verify}(k, m, \text{mac}(k,m)) = \text{ok}\]

- **Security:**
  - A MAC cannot be produced without access to the key

Similar to signature, but uses symmetric key

*What property does a signature have, but a MAC does not?*
Hash Functions

• Algorithm to hash a message m to a fixed length output hash(m)

• Security (Collision resistance)

Given hash function hash: X → Y, cannot find a collision, i.e. x, x' ∈ X s.t. x ≠ x' and hash(x) = hash(x')
Hash Functions

• Algorithm to hash a message $m$ to a fixed length output $\text{hash}(m)$
  – $\text{hash}(m) = m \% 10$, where $m$ is an integer

• Security (Collision resistance)

Given hash function $\text{hash}: X \rightarrow Y$, cannot find a collision, i.e. $x, x' \in X$ s.t. $x \neq x'$ and $\text{hash}(x) = \text{hash}(x')$. Satisfied?