18734 Recitation

Statistical Distance
Basic Cryptography

Amit Datta

Distance

- L1 distance

- Between two points
  - $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$

  - $\sum_i |x_i - y_i|$

Distance between functions

- Between two discrete functions
  - $m_1(x)$, $m_2(x)$
  - $x \in \{x_1, x_2, \ldots, x_n\}$
  - $\sum_i |m_1(x_i) - m_2(x_i)|$
- Between two continuous functions
  - $n_1(y)$, $n_2(y)$
  - $y \in [y_1, y_2]$
  - $\int_{y1}^{y2} |n_1(y) - n_2(y)| dy$

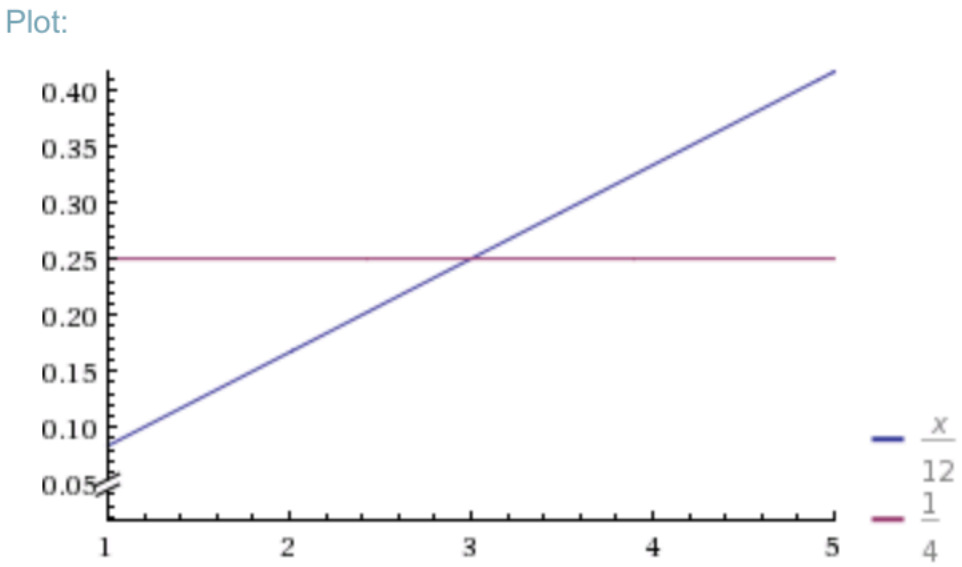
Distance between probability distributions

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

- Between two continuous distributions
  - PDFs $f_1(y)$, $f_2(y)$
  - $y \in [y_1, y_2]$
  - $\int_{y1}^{y2} |f_1(y) - f_2(y)| dy$

Exercise

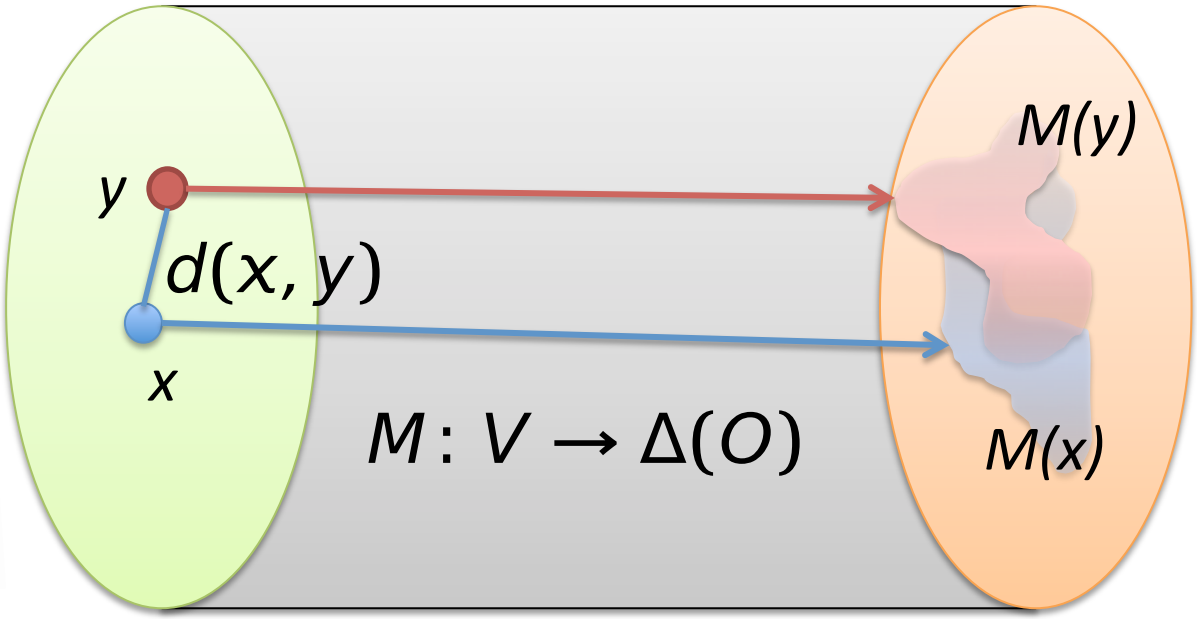
- Find L1 distance between the following distributions:
  - $f_1(x) = x/12 \quad x \in [1, 5]$
  - $f_2(x) = 1/4 \quad x \in [1, 5]$



Fairness through Awareness

Metric $d : V \times V \rightarrow \mathbb{R}$

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

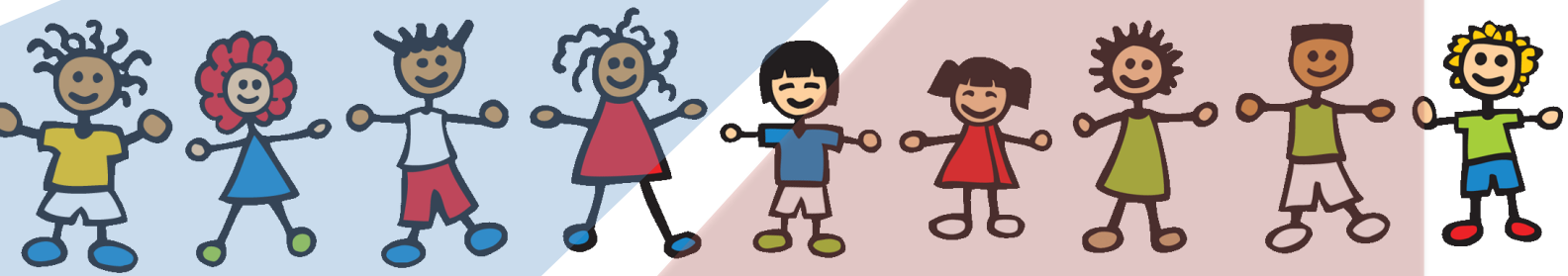


V: Individuals

O: 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
  - enc(m,k) = m $\oplus$ k
- Algorithm to *decrypt* a ciphertext
  - dec(c,k) = c $\oplus$ k
- Correctness:
  - *dec(enc(m,k),k) = m.* Satisfied?
- (Symbolic) Security:
  - A ciphertext cannot be decrypted without access to the key. Satisfied?

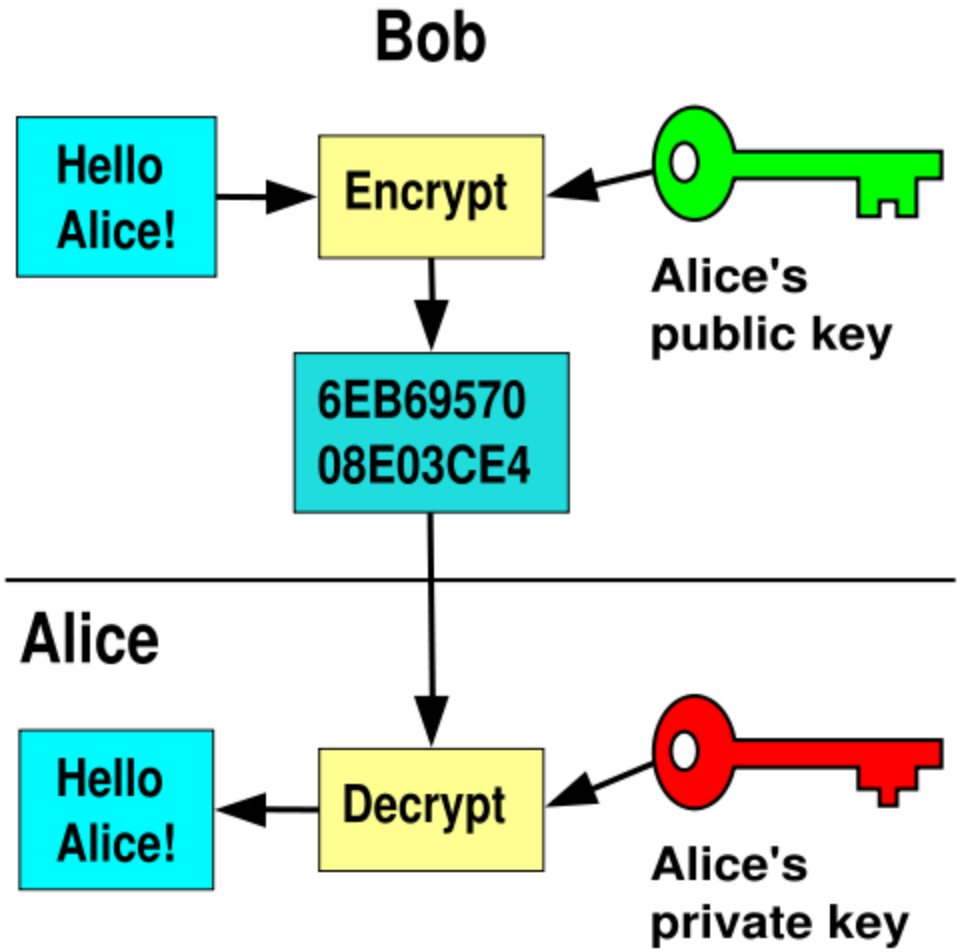
Public-Key Encryption Scheme

- *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 encrytion key returns m

$$dec(enc(m, pk(A)), sk(A)) = m$$

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

Public-Key Encryption Scheme



Public-Key Encryption Scheme

- *Key generation* algorithm
  - Generate random public key: e, secret key: d=1/e
- Algorithm to *encrypt* a message
  - enc(m,e) = m^e
- Algorithm to *decrypt* a ciphertext
  - dec(c,d) = c^d
- Correctness:
  - *dec(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?

RSA: http://en.wikipedia.org/wiki/RSA_(cryptosystem)

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

$$verify(m, sign(m, sk(A)), pk(A)) = ok$$

- 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
  - sign(m, e) = m^e
- Algorithm to *verify* signature
  - verify(m, c, d) = return *ok* iff m == c^d
- Correctness:
  - *verify(m,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

$$verify(k, m, mac(k,m)) = ok$$

- 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:* $X \rightarrow Y$, cannot find a collision, i.e. $x, x' \in X$ s.t. $x \neq x'$ and hash(x) = hash(x')

Hash Functions

- Algorithm to *hash a* message m to a fixed length output *hash(m)*
  - *hash(m) = m % 10, where m is an integer*

- 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'). Satisfied?