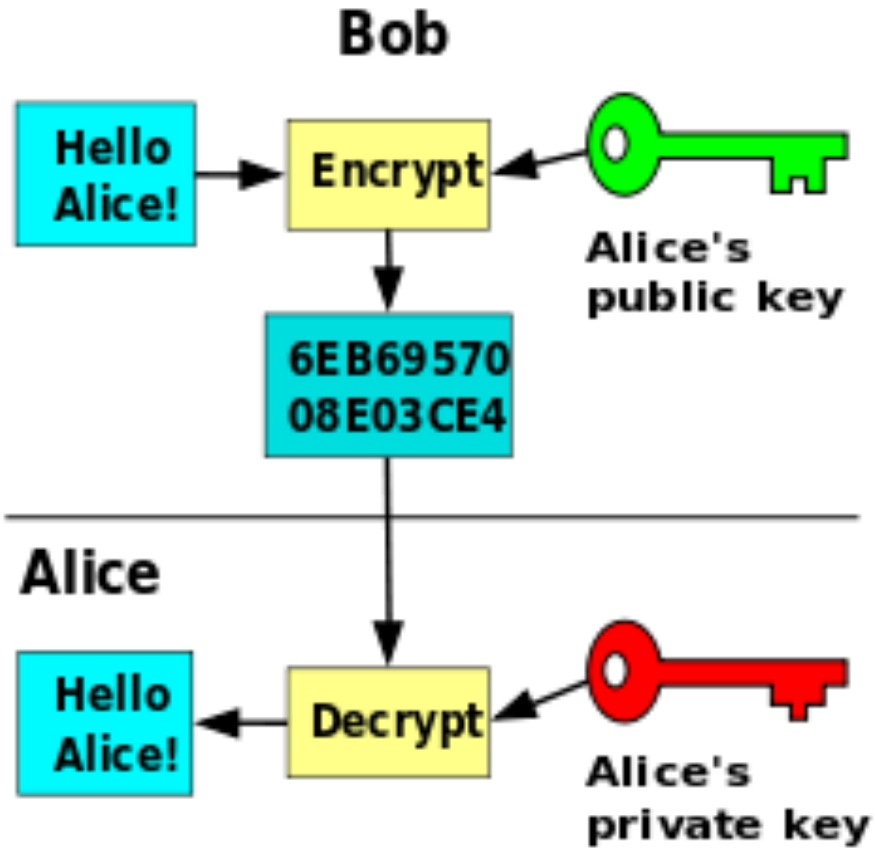
In Class Exercise

- Goal: Design a system in which
  - Individuals have sensitive personal data – set of attributes (medical records)
  - Data is somehow encrypted by the individual and stored at the cloud
  - A third-party wants to do computation on the data (medical center)
  - The third party also has secret inputs and does not want to share those with the cloud
  - Ideally, user is not involved

Paillier Cryosystem



- The public key: (n, g, h = g^x)
- Secret key: x ∈ [1, n^2/2]
- Strong secret:

  Factorization of n = zy (z , y are safe primes)

Paillier Cryptosystem Encryption

- To encrypt a message m ∈ Z_n
  - Select a random r ∈ [1, n/4]
  - Generate the ciphertext pair (C1,C2) such that
  - C1 = g^r mod n^2
  - C2 = h^r(1 + mn) mod n^2
  - [m]=(C1,C2)

The public key: (n, g, h = g^x)
Secret key: x ∈ [1, n^2/2]

Paillier Cryptosystem Decryption

- The message m can be recovered from [m]=(C1,C2) as follows:
  - m = Delta(C2 /C1^x)
  - Delta(u) = [(u−1) mod n^2]/n
    - For all u ∈ {u < n^2 | u = 1 mod n}

The public key: (n, g, h = g^x)
Secret key: x ∈ [1, n^2/2]

Paillier Cryptosystem Threshold Encryption

- Assume we randomly split the secret key in two shares x1 and x2 ,
  - x = x1 + x2
- The Paillier cryptosystem enables an encrypted message (C1,C2) to be partially decrypted to a ciphertext pair (C˜1,C˜2) using x1 as
  - C˜1 = C1
  - C˜2 = C2 /C1^(x1) mod n^2
- Then, (C˜1,C˜2) can be decrypted using x2

The public key: (n, g, h = g^x)
Secret key: x ∈ [1, n^2/2]

Homomorphism

- The product of two ciphertexts is equal to the encryption of the sum of their corresponding plaintexts

- A ciphertext raised to a constant number is equal to the encryption of the product of the corresponding plaintext and the constant

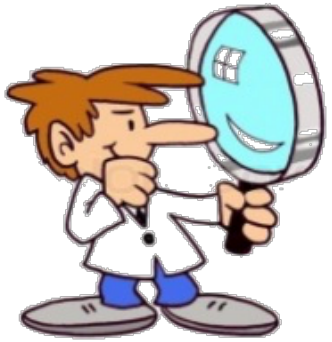
Tasks

- Decide on the system model and parties involved
- Decide on the threat model for all parties involved
- Design the system
  - Initialization: Key generation, key management, encryption
  - Application: SMC
- Comment on the functions that can be supported
- Comment on the security/privacy of the system
- Comment on the performance
- Comment on the user-friendliness

System Model

Threat Model

- Semi-honest adversary vs. Malicious adversary



- Polynomial-time adversary vs. computationally unbounded adversary

- Collusion

Requirements

- Types of supported queries:
  - Weighted Average
  - Multiplication of ciphertexts
  - Division
  - Comparison/Classification
- Access Control
- Access Patterns

Design

- Initialization

- Application(s)