CISC859: Topics in Advanced Networks & Distributed Computing: Network & Distributed System Security

Advanced Cryptographic Primitives

Advanced Crypto Techniques

- Secret Sharing
- Information Dispersal
- Blind signature
- Identity-based encryption
- Attribute-based encryption
- Homomorphic encryption
- Secure multi-party computation
  - Ex: private set intersection

Secret Sharing Schemes

- Q: How would you distribute a secret among n parties, such that only t or more of them together can reconstruct it.
  A: "(t, n)-threshold scheme"
- Physical world analogy: A safe with a combination of locks, keys.
- Some applications:
  - Storage of sensitive cryptographic keys
  - Command & control of nuclear weapons

Secret Sharing Schemes (cont'd)

E.g. An (n, n)-threshold scheme:

To share a k-bit secret K, the dealer D

- generates $n-1$ random k-bit numbers, $y_i, i = 1, 2, \ldots, n-1$,
- $y_n = K \bigoplus y_1 \bigoplus y_2 \bigoplus \cdots \bigoplus y_{n-1}$,
- gives the "share" y_i to party P_i

This is a "perfect" SSS: A coalition of less than n can obtain *no* information about the secret.

Q: How to generalize to arbitrary (t, n)?

Shamir's Threshold Scheme

Shamir's (t, n)-threshold scheme:

- D chooses prime p such that $p \geq n+1, K \in Z_p$
- generates distinct, random, non-zero $x_i \in Z_p, i = 1, 2, \ldots, n$
- generates random $a_i \in Z_p, i = 1, 2, \ldots, t-1$
- $a_0 = K$, the secret;
- $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \mod p$
- P_i's share is $(x_i, f(x_i)), i = 1, 2, \ldots, n$

- Fact: Shamir's scheme is a perfect SSS.

Background: Lagrangian Interpolation

Fact: There exists a unique polynomial of degree $\leq m$

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots a_1 x + a_0$$

over any $m+1$ points $(x_0, y_0), (x_1, y_1), \ldots, (x_m, y_m)$

Proof: The Vandermonde matrix $V = [x_{ij} = x_i^j | i, j = 0, 1, \ldots, m]$ non-singular for distinct x_is. Hence, $V \cdot a = y$ has a unique solution for any $y = [y_1, y_2, \ldots, y_m]$

Lagrange's Interpolation Formula: The unique polynomial through $(x_0, y_0), (x_1, y_1), \ldots, (x_m, y_m)$ is given by

$$f(x) = \sum_{i=0}^{m} L_{m,i}(x) y_i$$

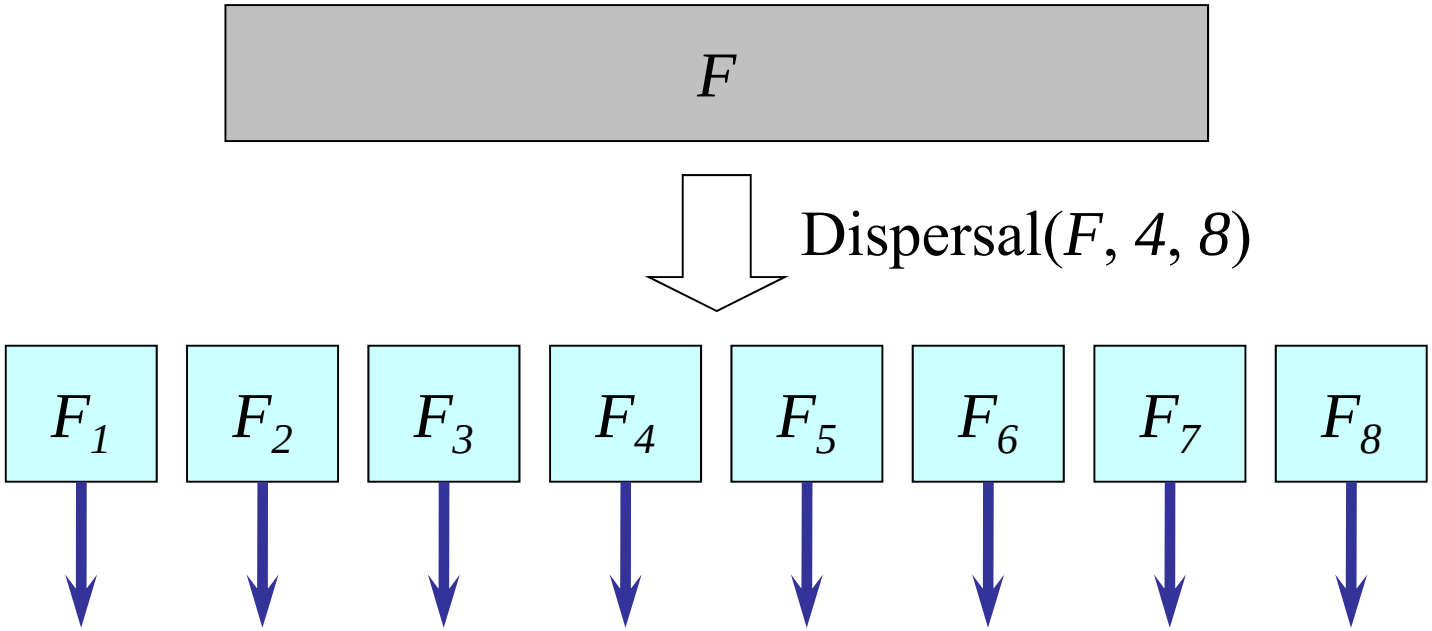
where $$L_{m,i}(x) = \prod_{j \neq i} (x - x_j) / \prod_{j \neq i} (x_i - x_j)$$

Rabin's Information Dispersal

- IDA was developed to provide safe and reliable transmission of information in distributed systems.
- Let F be a data of size N in byte $(|F| = N)$
- m should be less than or equal to $n (m \leq n)$
- Dispersal (F, m, n):
  - splitting the data F with some amount of redundancy resulting in n pieces $F_i (1 \leq i \leq n)$
  - $|F_i| = |F|/m$
    - ➔ Thus, the size of F, N, should be a multiple of m

Dispersal(F, m, n) – Example 1

- $|F| = 32$ bytes, $m = 4, n = 8$



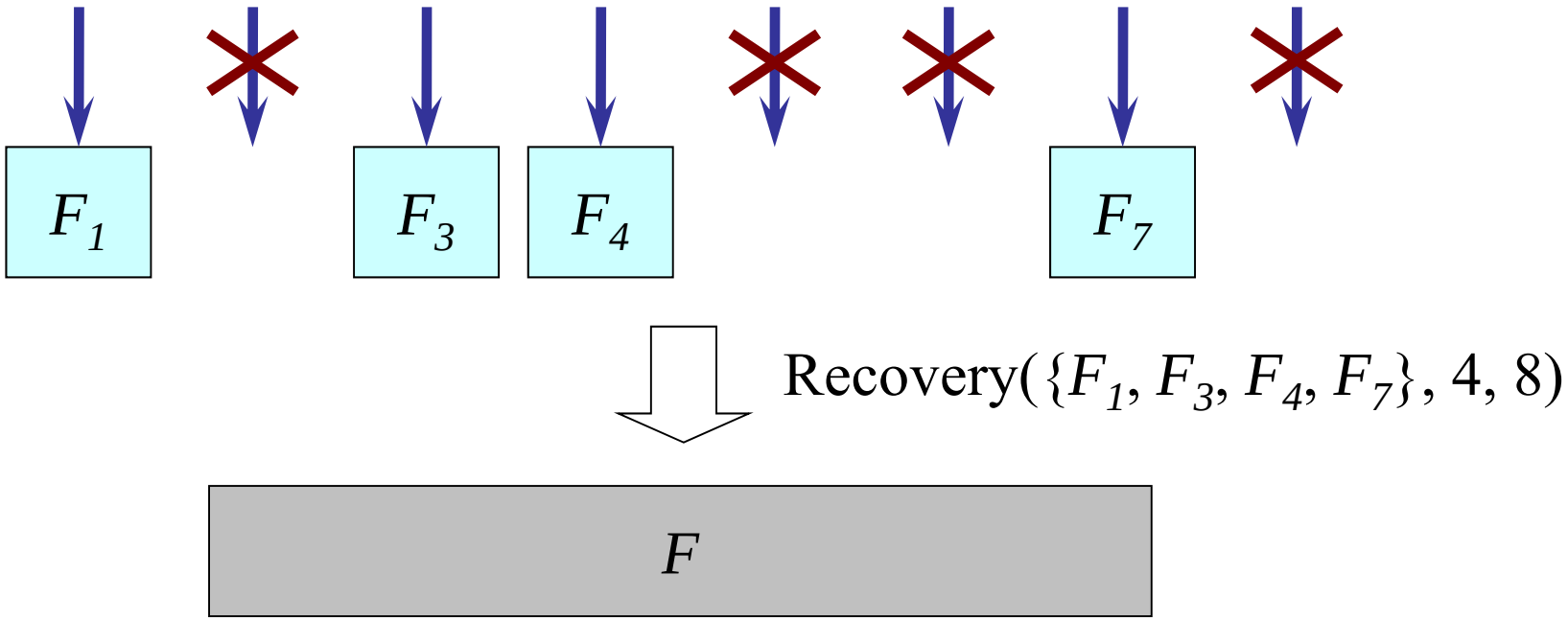
– $|F_i| = 32/4 = 8$ bytes $(1 \le i \le n)$

Recovery($\{F_{i_j} | (1 \leq j \leq m), (1 \leq i_j \leq n)\}, m, n$)

- Recovery$(\{F_{i_j} | (1 \leq j \leq m), (1 \leq i_j \leq n)\}, m, n)$:
  - reconstructing the original data *F* from any *m* pieces among *n* pieces (F_i $(1 \leq i \leq n)$)

Recovery($\{F_{i_j} | (1 \le j \le m), (1 \le i_j \le n)\}$, m, n) – Example 2

- $|F|$=32 bytes, m=4, n=8, $|F_i|$=8 bytes ($1 \le i \le 8$)
- Let us assume that the following 4(=m) pieces are received.



Detailed Operations

Dispersal(*F*, *m*, *n*)

- $F = b_1, b_2, \cdots, b_N$
  - $(|F| = N)$, and b_i represents each byte in *F* ($0 \le b_i \le 255 (=2^8-1)$).
  - All computations should be done in GF(2^8).
    - GF(2^8) is closed under addition and multiplication.
    - Every nonzero element in GF(2^8) has a multiplicative inverse.

- $F = (b_1, \cdots, b_m), (b_{m+1}, \cdots, b_{2m}), \ldots, (b_{N-m+1}, \cdots, b_N)$
  - $S_i = (b_{(i-1)m+1}, \cdots, b_{im})^T, 1 \le i \le N/m$

- The matrix, $\mathbf{M}(m \times N/m)$, is constructed as follows:
  - $\mathbf{M} = [S_1 \quad S_2 \quad \ldots \quad S_{N/m}]$

Dispersal(*F*, *m*, *n*)

- The matrix, $\mathbf{A}(n \times m)$, is constructed as follows:

$$\mathrm{A}=\begin{bmatrix}\mathbf{a}_1\\ \mathbf{a}_2\\ \vdots\\ \mathbf{a}_n\end{bmatrix}$$

  - $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{im}), 1 \leq i \leq n$
    - ➔ chosen such that every subset of *m* different vectors are linearly independent.

Dispersal(*F*, *m*, *n*)

- The following *Vandermonde matrix* satisfies the property required for $\mathbf{A}$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{m-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{m-1} \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} \end{bmatrix}$$

  - $m \le n$, and all x_i's are nonzero elements in GF(2^8) and pairwise different.
  - Any *m* different rows are linearly independent, so any matrix composed of a set of any *m* different rows is invertible.

Dispersal(*F*, *m*, *n*)

- *n* pieces, F_i ($1 \le i \le n$), are computed as follows:

$$\mathbf{A} \cdot \mathbf{M} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} \cdot \begin{bmatrix} S_1 & S_2 & \cdots & S_{N/m} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{a}_1 \cdot S_1 & \mathbf{a}_1 \cdot S_2 & \cdots & \mathbf{a}_1 \cdot S_{N/m} \\ \mathbf{a}_2 \cdot S_1 & \mathbf{a}_2 \cdot S_2 & \cdots & \mathbf{a}_2 \cdot S_{N/m} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{a}_n \cdot S_1 & \mathbf{a}_n \cdot S_2 & \cdots & \mathbf{a}_n \cdot S_{N/m} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix}$$

  - $\mathbf{a}_i \cdot S_j = (a_{i1} b_{(k-1)m+1} + \cdots + a_{im} b_{km})$

Dispersal(F, m, n) – Example 3

- $|F|$=32 bytes, m=4, n=8
  - $F = b_1, b_2, \ldots, b_{32}$
  - $F = (b_1, \ldots, b_4), (b_5, \ldots, b_8), \ldots, (b_{29}, \ldots, b_{32})$
  - $\mathbf{M}(4 \times 8)$

$$\mathbf{M} = [S_1 \quad S_2 \quad \ldots \quad S_8] = \begin{bmatrix} b_1 & b_5 & \cdots & b_{29} \\ b_2 & b_6 & \cdots & b_{30} \\ b_3 & b_7 & \cdots & b_{31} \\ b_4 & b_8 & \cdots & b_{32} \end{bmatrix}$$

Dispersal(F, m, n) – Example 3

- $\mathbf{A}(8 \times 4)$

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_8 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x_8 & x_8^2 & x_8^3 \end{bmatrix}$$

Dispersal(F, m, n) – Example 3

- F_i ($1 \le i \le 8$) are computed as follows:

$$\mathbf{A} \cdot \mathbf{M} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_8 \end{bmatrix} \cdot \begin{bmatrix} S_1 & S_2 & \dots & S_8 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{a}_1 \cdot S_1 & \mathbf{a}_1 \cdot S_2 & \cdots & \mathbf{a}_1 \cdot S_8 \\ \mathbf{a}_2 \cdot S_1 & \mathbf{a}_2 \cdot S_2 & \cdots & \mathbf{a}_2 \cdot S_8 \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{a}_8 \cdot S_1 & \mathbf{a}_8 \cdot S_2 & \cdots & \mathbf{a}_8 \cdot S_8 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_8 \end{bmatrix}$$

Recovery $(\{F_{i_j}|(1 \leq j \leq m), (1 \leq i_j \leq n)\}, m, n)$:

- Given *m* pieces $F_{i_j}(1 \leq j \leq m), (1 \leq i_j \leq n)\})$

$$\begin{bmatrix} F_{i_1} \\ F_{i_2} \\ \vdots \\ F_{i_m} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{i_1} \\ \mathbf{a}_{i_2} \\ \vdots \\ \mathbf{a}_{i_m} \end{bmatrix} \cdot \mathbf{M} = \mathbf{A}' \cdot \mathbf{M}$$

- **M** can be recovered from the given *m* pieces Fi_j ((1≤ j ≤m), (1≤ i_j ≤n)) because A' is invertible.

$$\mathbf{M} = \begin{bmatrix} \mathbf{a}_{i_1} \\ \mathbf{a}_{i_2} \\ \vdots \\ \mathbf{a}_{i_m} \end{bmatrix}^{-1} \cdot \begin{bmatrix} F_{i_1} \\ F_{i_2} \\ \vdots \\ F_{i_m} \end{bmatrix}$$

Recovery– Example 4

- $|F|$=32 bytes, m=4, n=8
- In example 3, F_i ($1 \le i \le 8$) pieces of 8 bytes are resulted.
- Assume that $\{F_1, F_3, F_4, F_7\}$ are received among them.

$$\begin{bmatrix} F_1 \\ F_3 \\ F_4 \\ F_7 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \cdot S_1 & \mathbf{a}_1 \cdot S_2 & \cdots & \mathbf{a}_1 \cdot S_8 \\ \mathbf{a}_3 \cdot S_1 & \mathbf{a}_3 \cdot S_2 & \cdots & \mathbf{a}_3 \cdot S_8 \\ \mathbf{a}_4 \cdot S_1 & \mathbf{a}_4 \cdot S_2 & \cdots & \mathbf{a}_4 \cdot S_8 \\ \mathbf{a}_7 \cdot S_1 & \mathbf{a}_7 \cdot S_2 & \cdots & \mathbf{a}_7 \cdot S_8 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_8 \end{bmatrix} \cdot \mathbf{M}$$

Recovery($\{F_{i_j} | (1 \leq j \leq m), (1 \leq i_j \leq n)\}$, m, n) – Example 4

- The original data M can be recovered by the following computation:

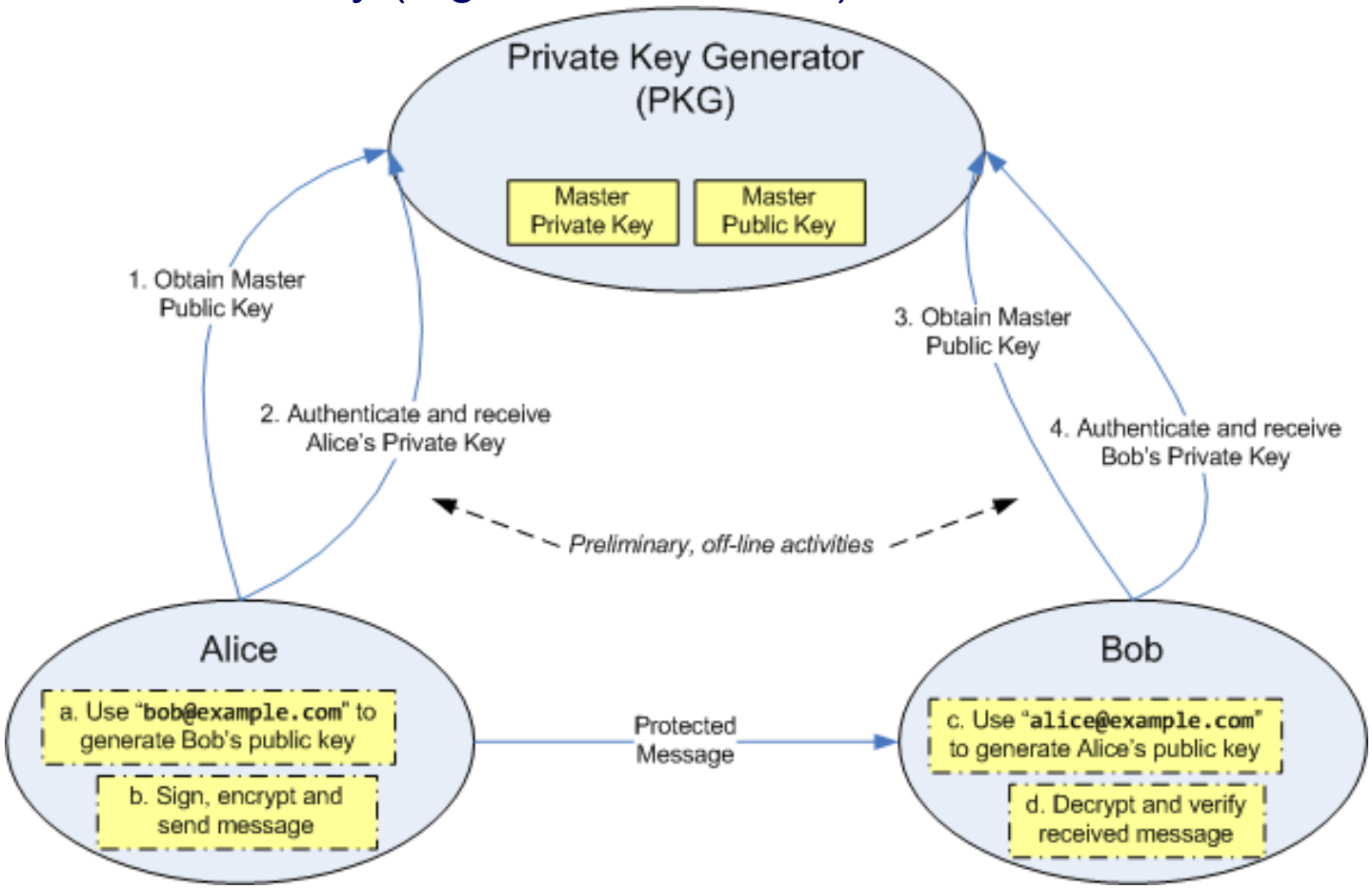
$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_8 \end{bmatrix}^{-1} \cdot \begin{bmatrix} F_1 \\ F_3 \\ F_4 \\ F_7 \end{bmatrix} = \mathbf{M}$$

Blind Signature

- Basic idea (http://en.wikipedia.org/wiki/Blind_signature)
  - It is a form of digital signature in which the content of a message is disguised (blinded) before it is signed
  - It provide unlinkability, which prevents the signer from linking the blinded message it signs to a later un-blinded version that it may be called upon to verify.
  - Applications: cryptographic election systems and digital cash schemes
  - Example: Blind RSA signature
    - Assume a signer's public and private keys are (e,n) and (d,n), respectively.
    - User to bank: $mr^e \bmod n$
    - Signer to user: $(mr^e)^d = m^d r \bmod n$
    - User computes $m^d r * 1/r = m^d \bmod n$

Identity-based Encryption (IBE)

- Basic idea (http://en.wikipedia.org/wiki/ID-based_encryption)
  - the public key of a user is some unique information about the user's identity (e.g., email address)



Identity-based Encryption (IBE)

- Key Advantages
  - No need for public-key distribution or certificates
  - The possibility to encode additional information into the recipient's ID (or public key)
    - ➔ e.g., ID+expiration time
- Key Drawbacks
  - The compromise of the PKG exposes all private keys
  - The PKG can sign on behalf of all users: no non-repudiation
  - A secure channel is needed to transfer the private key from the PKG to each user

Attribute-Based Encryption

- Basic idea (http://en.wikipedia.org/wiki/Attribute-based_encryption)
  - A type of public-key encryption in which the private key of a user and the ciphertext are dependent about attributes (e.g. the country he lives, or the kind of subscription he has).
  - the decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext
- Applications
  - Broadcast encryption: only the users with certain attributes can decrypt the ciphertext

Homomorphic Encryption

- Basic idea (http://en.wikipedia.org/wiki/Homomorphic_encryption)
  - It allows specific types of computations to be carried out on ciphertexts and obtain an encrypted result whose decryption matches the result of desired operations performed on the plaintext.

$$\mathbf{E}(m_1) \circ \mathbf{E}(m_2) = \mathbf{E}(m_1 \diamond m_2)$$

  - The desired operation on plaintext can be addition or multiplication
  - Multiplicative example: RSA

$$\mathbf{E}(m_1) \times \mathbf{E}(m_2) = m_1^e m_2^e \mod n = (m_1 m_2)^e \mod n = \mathbf{E}(m_1 m_2)$$

  - Additive example: Paillier cryptosystem
  - Fully homomorphic encryption published in 2009

Additive Example: Paillier cryptosystem

- Encryption

$$\mathbf{E}(m, r) = g^m r^N \mod N^2$$

- Homomorphic

$$\mathbf{E}(m_1, r_1) \times \mathbf{E}(m_2, r_2) = \mathbf{E}(m_1 + m_2, r_1 r_2) \mod N^2$$

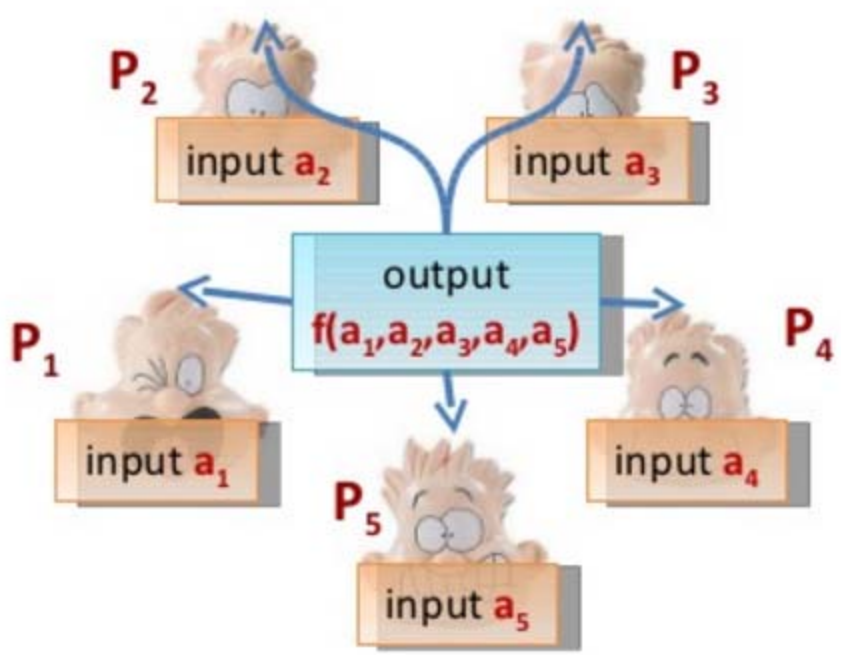
$$\mathbf{E}^{m_2}(m_1, r_1) = \mathbf{E}(m_1 m_2, r_1^{m_2}) \mod N^2$$

- Self-blinding

$$\mathbf{E}(m_1, r_1) r_2^N \mod N^2 = \mathbf{E}(m_1, r_1 r_2)$$

Secure Multi-Party Computation

- A subfield of cryptography
- Multiple parties jointly compute a function over their inputs while keeping those inputs private



Private Set Intersection (PSI)

- Problem setting
  - Two parties, say Alice and Bob, each have a private data set
- A PSI protocol allows Alice and Bob to find out the intersection of their data sets without disclosing additional information to each other
- A PSI Cardinality (PSI-CA) protocol allows Alice and Bob to find out the intersection cardinality of their data sets without disclosing additional information to each other

An Example

Alice

$A = \{a_1, a_2, \ldots, a_n\}$

Bob

$B = \{b_1, b_2, \ldots, b_m\}$

$$f(x) = \prod_{i=1}^{m}(x - b_i) = \sum_{i=0}^{m} f_i x^i$$

$$\mathbf{E}(f_0), \mathbf{E}(f_1), \mathbf{E}(f_2) \cdots, \mathbf{E}(f_m) \longleftarrow$$

$$\mathbf{E}(r_1 f(a_1) + a_1), \mathbf{E}(r_2 f(a_2) + a_2), \cdots, \mathbf{E}(r_n f(a_n) + a_n) \longrightarrow$$

$$r_1 f(a_1) + a_1, r_2 f(a_2) + a_2, \cdots, r_n f(a_n) + a_n \longleftarrow$$

$r_i f(a_i) + a_i = a_i? \quad 1 \leq i \leq n$

In-class Exercise

- Alice has $\mathbf{a} = (a_1, a_2, \ldots, a_k)$
- Bob has $\mathbf{b} = (b_1, b_2, \ldots, b_k)$

- How to securely compute $\mathbf{a} \cdot \mathbf{b}$?

- How to securely compute $||\mathbf{a} - \mathbf{b}||_2^2 = \sum_{i=1}^{k} (a_i - b_i)^2$