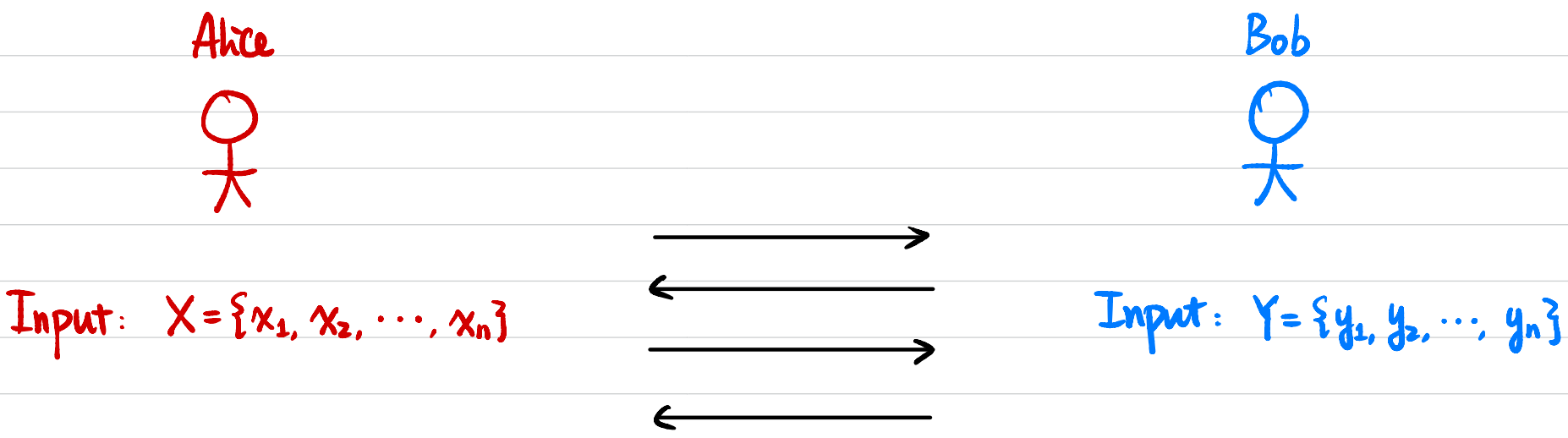
CSCI 1515 Applied Cryptography

This Lecture:

- Private Set Intersection
- Introduction to Fully Homomorphic Encryption
- Somewhat Homomorphic Encryption over Integers

Private Set Intersection (PSI)



PSI: $f(X, Y) = X \cap Y$

PSI-CA: $f(X, Y) = |X \cap Y|$

Applications:

- Password Breach Alert (Chrome, Edge, Firefox, iOS keychain, ...)
- Ads Conversion Measurement (Google)
- Privacy-Preserving Inventory Matching (J.P. Morgan)
- Private Contact Discovery (Signal)

Private Set Intersection (PSI)

Input: $X = \{x_1, x_2, \cdots, x_n\}$

$H(x_1), \cdots, H(x_n)$ →

? ← $H(y')$

Input: $Y = \{y_1, y_2, \cdots, y_n\}$

$Y' = \{y_1', y_2', \cdots, y_n'\}$

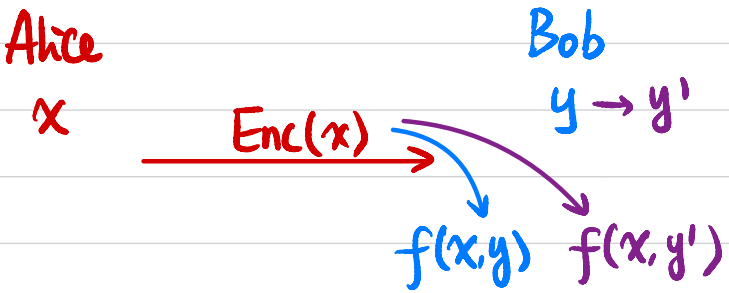
$H(y_1), \cdots, H(y_n)$ ⇓ $X \cap Y$

$H(y_1'), \cdots, H(y_n')$ ⇓ $X \cap Y'$

Is it (semi-honest) secure?

No!

Is it possible to achieve 2PC / MPC with 1 round of communication? No!



DDH-based PSI

Cyclic group G of order q with generator g, where DDH holds.

$H: \{0,1\}^* \to G$ (modeled as Random Oracle)

$(g^x, g^y, g^{xy}) \stackrel{c}{\approx} (g^x, g^y, g^z)$

Alice

Bob

Input: $X = \{x_1, x_2, \cdots, x_n\}$

$a \xleftarrow{\$} \mathbb{Z}_q$

Input: $Y = \{y_1, y_2, \cdots, y_n\}$

$b \xleftarrow{\$} \mathbb{Z}_q$

g^{xy} (under g^x, y)

$H(Y)^b := \{H(y_1)^b, \cdots, H(y_n)^b\}$ ←

$H(Y)^{b \cdot a} := \{H(y_1)^{b \cdot a}, \cdots, H(y_n)^{b \cdot a}\}$ →

$H(X)^a := \{H(x_1)^a, \cdots, H(x_n)^a\}$

$H(y')^{b \cdot a} = ?$

$H(Y)^{b \cdot a} \cap H(X)^{a \cdot b}$

$\Downarrow$

$X \cap Y$

Is it (semi-honest) secure?

PSI-CA?

PSI-CA: $f(X, Y) = |X \cap Y|$

Input: $X = \{x_1, x_2, \cdots, x_n\}$

$a \xleftarrow{\$} \mathbb{Z}_q$

Input: $Y = \{y_1, y_2, \cdots, y_n\}$

$b \xleftarrow{\$} \mathbb{Z}_q$

$$\xleftarrow{H(Y)^b := \{H(y_1)^b, \cdots, H(y_n)^b\}}$$

$$\xrightarrow{H(Y)^{b \cdot a} := \{H(y_1)^{b \cdot a}, \cdots, H(y_n)^{b \cdot a}\}_{\text{shuffle}}}$$

$$H(X)^a := \{H(x_1)^a, \cdots, H(x_n)^a\}$$

$$\{H(Y)^{b \cdot a}\}_{\text{shuffle}} \cap H(X)^{a \cdot b}$$

$$\downarrow$$

$$|X \cap Y|$$

Fully Homomorphic Encryption (FHE)

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 + m_2)$

Additively Homomorphic

↑

Exponential ElGamal / Pallier / Regev

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 \cdot m_2)$

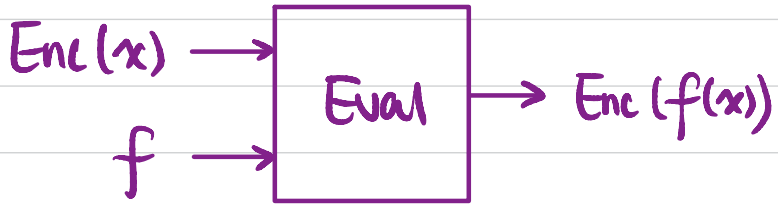
Multiplicatively Homomorphic

↑

RSA / ElGamal

Fully Homomorphic: Additively & Multiplicatively Homomorphic

Homomorphic Evaluation:



Application: Outsourcing Storage & Computation

Server

Client

Data x

Key sk

$ct \leftarrow Enc(x)$

ct

f

$ct' \leftarrow Eval(f, ct)$

ct'

$f(x) \leftarrow Dec_{sk}(ct')$

Application: Privacy-Preserving Query

Server

Client

Query x

Key sk

$ct \leftarrow Enc(x)$

ct

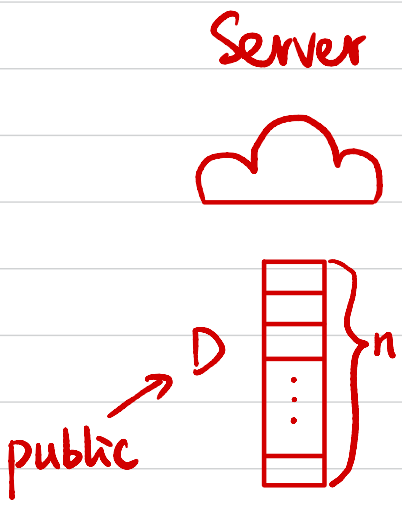
Search/ML/LLM/...

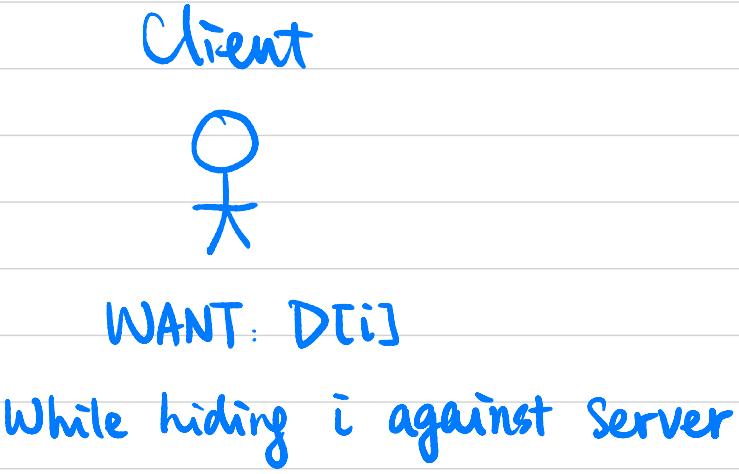
$ct' \leftarrow Eval(f, ct)$

ct'

$f(x) \leftarrow Dec_{sk}(ct')$

Application: Private Information Retrieval (PIR)





Query i
Key sk
$ct \leftarrow Enc(i)$

ct ←

$ct' \leftarrow Eval(f, ct)$

$f_D(i) = D[i]$

→ ct'

$D[i] \leftarrow Dec_{sk}(ct')$

Fully Homomorphic Encryption (FHE)

- **Syntax:** A (public-key) homomorphic encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ w.r.t. function family $\mathcal{F}$:
  - $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
  - $ct \leftarrow \mathsf{Enc}_{pk}(m) \quad m \in \{0,1\}$
  - $m \leftarrow \mathsf{Dec}_{sk}(ct)$
  - $ct_f \leftarrow \mathsf{Eval}(f, ct_1, \cdots, ct_n)$ → output $ct_f = (f, ct_1, \cdots, ct_n) \xrightarrow{\mathsf{Dec}} f(m_1, \cdots, m_n)$ $\quad f: \{0,1\}^n \to \{0,1\}$

- **Correctness:** $\forall f \in \mathcal{F}, \quad \forall m_1, m_2, \cdots, m_n \in \{0,1\}$

  $\forall i \in [n], \; ct_i \leftarrow \mathsf{Enc}_{pk}(m_i), \quad ct_f \leftarrow \mathsf{Eval}(f, ct_1, \cdots, ct_n).$

  $\mathsf{Dec}_{sk}(ct_f) = f(m_1, \cdots, m_n)$

- **(CPA) Security:** $(pk, \mathsf{Enc}_{pk}(m_0)) \overset{c}{\approx} (pk, \mathsf{Enc}_{pk}(m_1))$.

- **Compactness:** $|ct_f| \leq$ fixed poly (λ) ← Why do we need this?

  Independent of circuit size of f.

- If $\mathcal{F}$ contains **all** poly-sized Boolean circuits, then Π is **fully** homomorphic.

FHE Constructions

Step 1: Somewhat Homomorphic Encryption (SWHE)

- over Integers
- from RLWE (BFV)

Step 2: Bootstrapping

SWHE over Integers

Attempt 1 (secret-key)

- secret key: odd number p ← Why odd?
- $\mathrm{Enc}(m)$: $m \in \{0,1\}$
  Sample a random q.
  Output $ct = p \cdot q + m$
  Encryption of 0 is a multiple of p.
- $\mathrm{Dec}(ct)$: $ct \bmod p$
- Eval ADD: $ct \leftarrow ct_1 + ct_2$
  Eval MULT: $ct \leftarrow ct_1 \cdot ct_2$

$ct_1 = p \cdot q_1 + m_1$

$ct_2 = p \cdot q_2 + m_2$

$ct_1 + ct_2 = p \cdot (q_1 + q_2) + (m_1 + m_2)$

$ct_1 \cdot ct_2 = p^2 \cdot q_1 q_2 + p \cdot (q_1 \cdot m_2 + q_2 \cdot m_1) + m_1 \cdot m_2$

Why is it homomorphic?

(CPA) Security?

$\mathrm{Enc}(0)$: $\gcd \begin{pmatrix} p \cdot q_1 \\ p \cdot q_2 \\ \vdots \end{pmatrix} = p$

SWHE over Integers

Attempt 2 (secret-key)

- secret key: odd number p
- $\mathrm{Enc}(m)$: $m \in \{0,1\}$

  Sample a random q. Sample a random $e \ll p$ (noise)

  Output $ct = p \cdot q + m + 2e$

  Encryption of 0 is small and even modulo p.

- $\mathrm{Dec}(ct)$: $[ct \bmod p] \bmod 2$
- Eval ADD: $ct \leftarrow ct_1 + ct_2$ (ADD mod 2) (XOR)

  Eval MULT: $ct \leftarrow ct_1 \cdot ct_2$ (MULT mod 2) (AND)

$ct_1 = p \cdot q_1 + m_1 + 2e_1$

$ct_2 = p \cdot q_2 + m_2 + 2e_2$

Why is it homomorphic?

$ct_1 + ct_2 = p \cdot (q_1 + q_2) + (m_1 + m_2) + 2(e_1 + e_2)$

(CPA) Security?

$ct_1 \cdot ct_2 = p \cdot (\cdots) + 2 \cdot (\cdots) + m_1 \cdot m_2$

How homomorphic is it?

Approximate GCD Problem

> Given polynomially many $\{x_i = p \cdot q_i + s_i\}$, find p.

Example parameters:

$$p \sim 2^{O(\lambda^2)}, \quad q_i \sim 2^{O(\lambda^5)}, \quad s_i \sim 2^{O(\lambda)}$$

Best known algorithms take $\sim 2^{\lambda}$ time