Secret Sharing and PIR

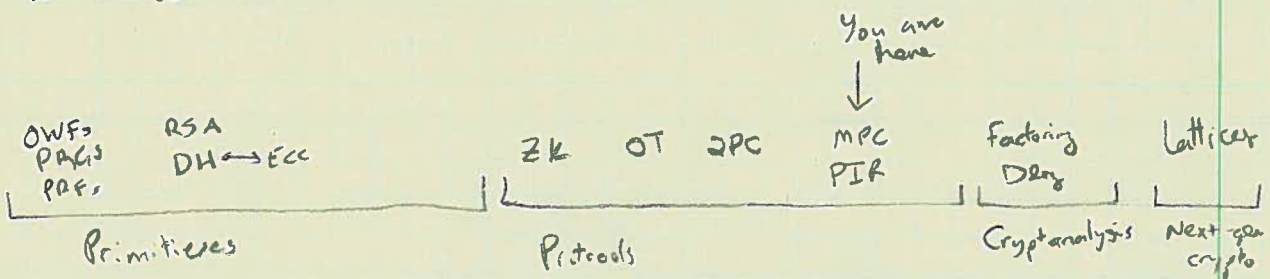
May 24

→ We are not going to cover n-party computation.

→ We are going to cover secret sharing
  ↳ a key ingredient in MPC ... important for everyone to know.

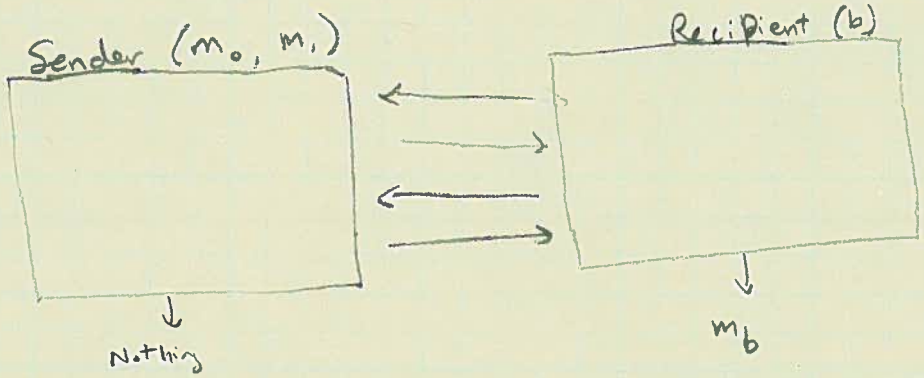
→ Also PIR ... also important for you to know (uses secret sharing)

To review the structure of the course:



Review

Oblivious transfer - powerful primitive. Enough to build MPC.



Receiver Privacy (Sender doesn't learn b)

$\forall\ m_0, m_1$ $\quad View_S((m_0, m_1), 0) \overset{c}{\approx} View_S((m_0, m_1), 1)$

Sender Privacy (receiver learns only m_b)

$\forall$ receivers R $\exists$ ppt Sim s.t.

$View_R((m_0, m_1), b) \overset{c}{\approx} Sim(b, m_b)$

"what leaks"

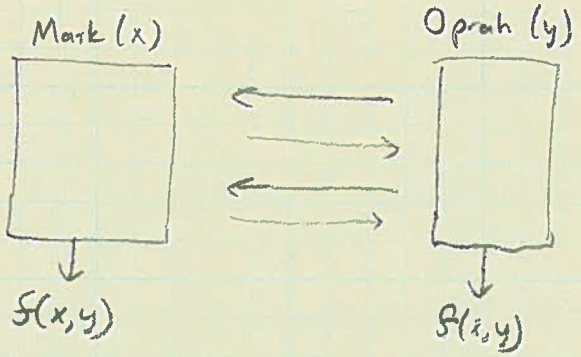
⇒ Simulation !!! ⇐

→ Can build from DDH, RSA, ...

→ Seems unlikely to come generically from k.E.

Review: Garbled Circuits

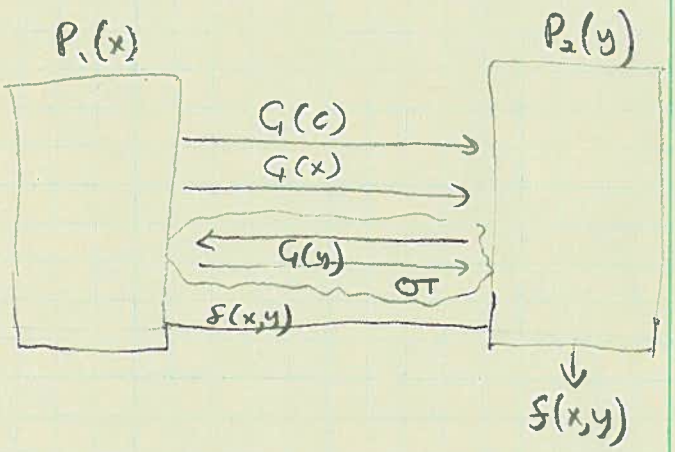
- Means to implement 2PC
- Both learn $f(x,y)$ and "nothing else"



Yao's GC — Built on OT

- P_1 creates "garbled" version of ckt C implementing $f(\cdot)$
- P_1 sends $G(c)$ to P_2
- P_1 sends $G(x)$ to P_2
- P_2 uses OT to get $G(y)$ from P_1
- P_2 can use

$$G(c), G(x), G(y) \Rightarrow f(x,y)$$



Use simulation to define & prove security

Secret Sharing

- The story I heard from my undergrad crypto prof
- Board of directors from Occi Cola wants to encrypt the secret recipie s.t.
  - * each director gets a secret key (n directors)
  - * if any $\frac{2}{3}n$ of the hundred directors combine their secret key $\Rightarrow$ can decrypt (fault-tolerance in case someone disappears)
  - * any smaller subset learns nothing about secret recipe. (defectors can't steal recipe unilaterally)

Naïve solution:
- give each director a key k_i, $1 \le i \le n$.
- ct is, for each subset $S \subseteq \{1, \dots, n\}$ s.t. $|S| = \frac{2}{3}n$

$ct = \langle ct_{S_1}, ct_{S_2}, \dots \rangle$

$ct_{\{k_1, \dots, k_{\frac{2}{3}n}\}} = E(k_1, E(k_2, E(k_3, \dots E(k_{\frac{2}{3}n}, m)\dots)$

$ct_{\{k_2, \dots, k_{\frac{2}{3}n+1}\}}$

$\vdots$

Problem: ct contains many subsets!

How many?

$\binom{n}{\frac{2}{3}n}$ = "n choose $\frac{2}{3}n$"

Useful life fact: $\left(\frac{n}{k}\right)^k \le \binom{n}{k} \le \left(\frac{ne}{k}\right)^k$

So ct size grows like

$\binom{n}{\frac{2}{3}n} \ge \left(\frac{3}{2}\right)^{\frac{2}{3}n}$ ← Exponential in # of board members!

Secret Sharing

Formally....

A (t,n)-secret sharing scheme over msg space $\mathcal{M}$ and secret space $\mathcal{S}$ consists of p.p.t. algs

$$\text{Share}: \mathcal{M} \rightarrow \mathcal{S}^n$$
$$\text{Recon}: \mathcal{S}^t \rightarrow \mathcal{M}$$

s.t.

Correctness "Given any t shares, can recover m."

$\forall\, m \in \mathcal{M}$, $\{s_1, \dots, s_n\} \leftarrow \text{Share}(m)$

$\forall\, S \subseteq \{s_1, \dots, s_n\}$ of size t: $m = \text{Recon}(S)$.

Secrecy "Need at least t shares to learn anything about m"

$\forall\, m_0, m_1\ \forall\, I \subseteq \{1, \dots, n\}$ of size t

$$\left\{ \{s_i \mid i \in I\} : \{s_1, \dots, s_n\} \leftarrow \text{Share}(m_0) \right\} = \left\{ \{s_i \mid i \in I\} : \{s_1, \dots, s_n\} \leftarrow \text{Share}(m_1) \right\}$$

Identical distributions.

The secret sharing scheme we will see is information theoretic. Secure against advs running in unbounded space & time.

↳ Contrast w/ PRG-style security

We would like shares to be as small as possible.

Shamir Secret Sharing (1979)

↳ Also see Blakley (1979)

Idea: * Any $d+1$ distinct points on a degree-d polynomial allow you to recover the entire polynomial.

* Only d points are insufficient.

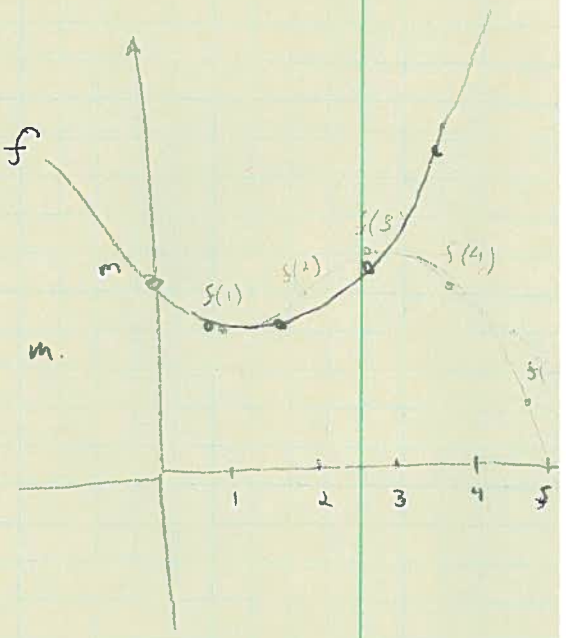
These facts hold over a finite field $\mathbb{F}$ (modulo prime p)

Share: $\mathbb{F} \longrightarrow (\mathbb{F}, \mathbb{F})^n$ s.t. $|\mathbb{F}| > n$.

($\mathbb{F}$: msg; $(\mathbb{F}, \mathbb{F})^n$: shares)

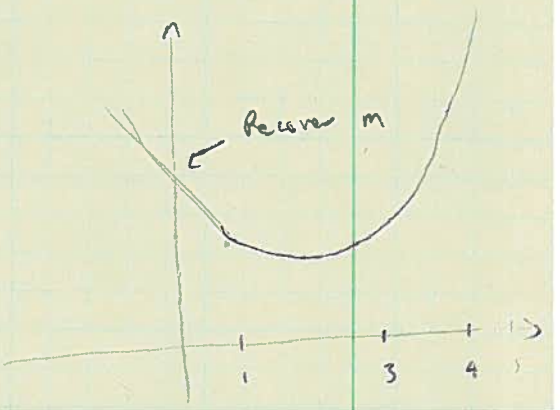
Share(m):

- Choose random poly $f \in \mathbb{F}[x]$ s.t.
  - $f(0) = m$
  - f has degree $t-1$.
- Output $\begin{bmatrix} S_1 = (1, f(1)) \\ S_2 = (2, f(2)) \\ S_3 = (3, f(3)) \\ \vdots \\ S_n = (n, f(n)) \end{bmatrix}$ as shares of m.



Reconstruct $((x_1, y_1), \ldots, (x_t, y_t))$

- Use polynomial interpolation to recover f
- Output $f(0)$.



Shamir Secret Sharing

Correctness: By Uniqueness of interpolating polynomial.

Security: There $\exists$ an equal # of polynomials that interpolate only $t-1$ shares and pass through $f(0)=m$ for all $m \in \mathcal{M}$.

Efficiency.

Share: Choose random coefficients and shift to get $f(0)=m$.

Recon: Given $(x_1, y_1) \dots, (x_t, y_t)$, want to get f of degree $t-1$ s.t. $y_i = f(x_i)$. $\forall\, i \in \{1, \dots, t\}$.

Idea: Say $f(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_{t-1} x^{t-1}$.

Then
$$\begin{aligned} f(x_1) &= \alpha_0 x_1 + \alpha_2 x_1^2 + \dots + \alpha_{t-1} x_1^{t-1} &= y_1 \\ f(x_2) &= \alpha_0 x_2 + \alpha_2 x_2^2 + \dots + \alpha_{t-1} x_2^{t-1} &= y_2 \\ &\vdots & \vdots \\ f(x_t) &= \dots & = y_t \end{aligned}$$

$$\begin{pmatrix} & X & \end{pmatrix} \begin{pmatrix} \vec{\alpha} \end{pmatrix} = \begin{pmatrix} \vec{y} \end{pmatrix}$$

$$X\vec{\alpha} = \vec{y}, \text{ or } \vec{\alpha} = X^{-1}\vec{y}.$$

Use linear algebra to recover $\vec{\alpha}$s, get f, and recover message.

Is X non-zero?

Connection to Error-Correcting Codes

Secret-sharing schemes closely related to ECC.

Recieve: 

→ Want to recover encoded msg given only some "large enough" subset of the blocks

→ Connection is close! Can view Shamir SS sharing as Reed-Solomon encoding (see CS250)

Another related problem:

- Say that the board of directors meets to recover the Coca-Cola recipe from shares.
- All n board members show up
  - $\frac{2}{3}n$ give correct shares
  - $\frac{1}{3}n$ lie and give corrupt shares

⇒ Can you still recover the recipe?!

↳ Brute-force decoding will be exponential time!

↳ Need to be more clever...

What if you have a long secret?

→ Bad: use huge field FF

→ Better: use bit-by-bit sharing

→ Best: use PRG! $\langle \text{Share}(k),\ E(k,m) \rangle$

↳ no longer info-theoretic