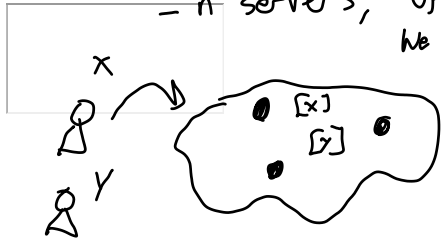


Recap of MPC:

- n servers, up to k of them can fail.
We use secret sharing of degree k .



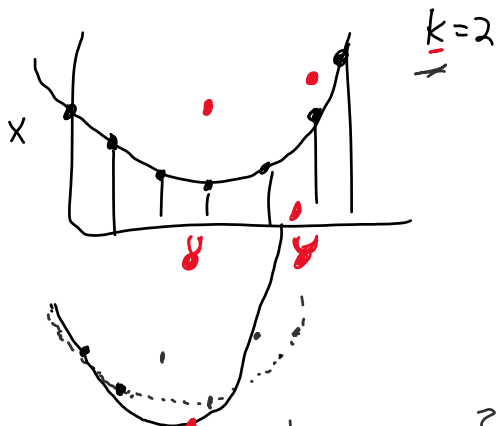
$$a_0 = x \quad a_1 \in \mathbb{F}_p \quad \dots \quad a_k \in \mathbb{F}_p$$

- Secret Share(x): as client
Sampling φ uniformly (among p^k possibilities)
s.t. $\varphi(0) = x$ and φ is degree- k
Send $\varphi(i)$ to server S_i as $[x]^{(i)}$

- Open($[x]$): as server S_i
Send $[x]^{(i)}$ to each other server
Receive shares from other servers $[x]^{(j)}$
Interpolate a polynomial φ s.t. $\varphi(i) = [x]^{(i)}$
output $x = \varphi(0)$.

Robust Reconstruction.

- Suppose the servers that fail (up to k) they return invalid shares during Open
- Can we ensure we get the correct value anyway?

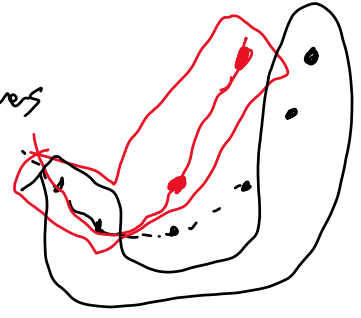


- * Only one subset of 4 that works?
- Find a large enough subset ~~4~~? 5?
- Potentially two subsets each intersecting 4 points
- If we can find a \deg^k poly intersecting

$2K+1$ received shares.
 \Rightarrow we can conclude it is the correct one.

Among $2K+1$ received shares, at most K are from malicious nodes, so at least $K+1$ are correct shares. And $K+1$ correct shares uniquely determine the correct ϕ .

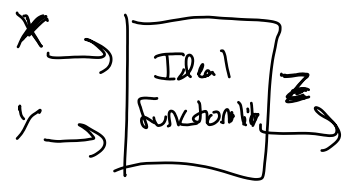
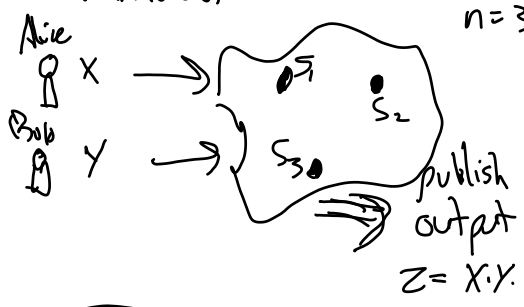
$n \geq 2K+1$ is the minimum number of servers relative to fault tolerance K for which robust interpolation works.



$n \geq 3K+1$ is the minimum to ensure guaranteed output (if the $2K+1$ first shares aren't successful, wait for more and try again).
 Output once some subset of $2K+1$ received shares lie on a deg K poly.

Simulation Security for MPC.

$n=3, K=1$, Semihonest



View can be simulated given just Z

Input: X, Y
 Preproc: $[a], [b], [ab]$.

Procedure:
 Alice Secretshares(X)
 Bob Secretshares(Y)

Servers:
 $D := \text{Open}([X] - [a])$
 $E := \text{Open}([Y] - [b])$
 $[Z] := DE + \dots + [ab]$
 output $Z := \text{Open}([Z])$

★ All interpretation:
 flip a bit, either produce the real view or the simulation. These are indistinguishable.

★ For all X_L, Y_L, X_R, Y_R , st. $Y_L \cdot X_L = Y_R \cdot X_R$.

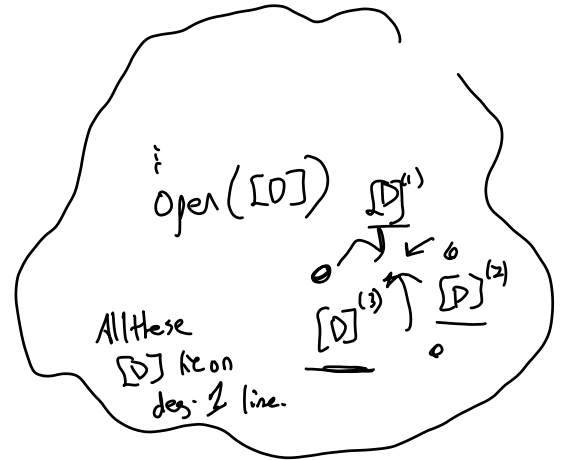
$\text{View}_L \approx \text{View}_R$

\Rightarrow What is the view of S_i in this protocol?
 on inputs everything S_i sees.

Beamer ml

Server i sees i	Total fp.
• $[x], [y]$	$2x$
• $[a], [b], [ab]$	$3x$
• ϕ_D from reconstruction	$2x$ (deg/poly)
• ϕ_E	$\phi_D(i) = [x-a] \quad 2x$
• ϕ_Z	$2x$

Total $11x$ fp elems.



Give a Pr distribution	Dof
$\phi_D(i) = [x-a]$	-1
$\phi_E(i) = [y-b]$	-1
$\phi_Z(i) = DF + \dots [ab]$	-1
$\phi_Z(o) = Z$	-1

In total, P possible views, all equally likely.

How can it be simulated given Z .

given prob dist. above, sample in any order that satisfies constraints.

$S(Z)$:
 ~~$\phi_Z \neq$ deg-1 poly.~~

Constant Round Vector Product.

Input: $[x_1] \dots [x_m]$ Precondition: $x_i \neq 0$.

Output: $[x_1 \cdot x_2 \dots x_m] = [\prod x_i]$

Preprocess: any.

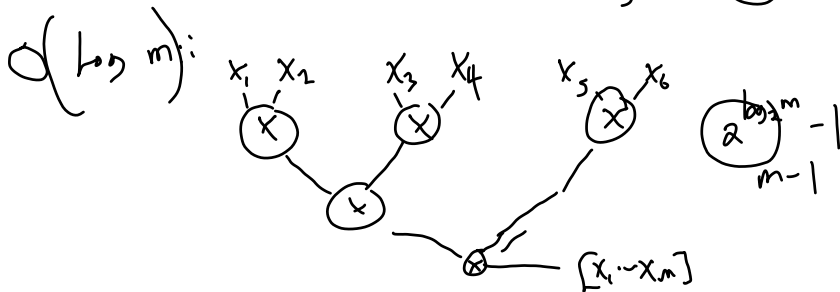
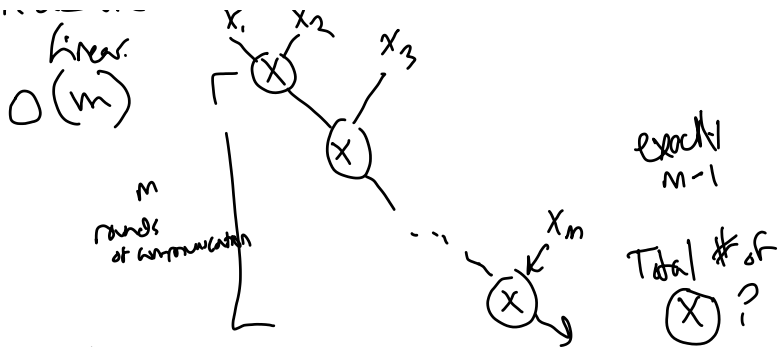
Perf goal: constant # of rounds
 Init: $\Delta^x [x]$



• $open(D) \cdot open(E)$

adapt $\rightarrow [xy] \rightarrow xy$

Procedure: ...



$O(1)$: lookup table? and overhead is $O(m)$.

related: $[x_1], \dots, [x_k]$

input: $[i]$ output: $[X_i]$

$\sum_{j=1}^k ([i] \rightarrow j) \cdot [X_j]$

$x_1 \quad \times \quad x_2 \quad \times \quad \dots \quad x_m$

\Downarrow "mask" $D = (x - a)$ \leftarrow 2x base rings

$r_0, r_1, \dots, r_m \in \mathbb{F}_p$

$[r_1] \times [x_1] \times [r_1^{-1}] \times [r_2] \times [x_2] \times [r_2^{-1}] \dots$

\downarrow open $([r_i, x_i, r_i^{-1}])$

$(r_0 \cdot x_1 \cdot r_1^{-1}) \quad (r_1 \cdot x_2 \cdot r_2^{-1}) \quad \dots \quad (r_{m-1} \cdot x_m \cdot r_m^{-1})$

$x_i \neq 0 \Rightarrow (r_0 \cdot x_i \cdot r_i^{-1})$ uniformly distributed.

all $r_i x_{i+1} r_{i+1}^{-1}$ uniformly distributed

$(r_0^{-1}) \times \prod_{i=0}^m (r_i \cdot x_i \cdot r_i^{-1}) \times [r_m] = [x_1 \dots x_m]$

possible to do w/ only $r_0 \dots r_m$? choose $[r_m]$ so that $r_m \cdot \prod_{i < m} r_i = 1$.

• prepare $[r_i] [r_i^{-1}]$

+ 2x base rings

• Total cost: $\sim 2.5x$ overhead?

DFT over Finite Fields.

$$\text{DFT (over roots)} \\ X_k = \sum_{j=0}^{n-1} x_j \left(e^{(2\pi i/n) \cdot k \cdot j} \right) \quad \sum x_j \omega^{kj}$$

Primitive n th roots of unity

- n th root of unity is $\omega^n = 1$
- primitive $\Rightarrow n$ is the smallest > 0 for which is true.

$$\omega = e^{2\pi i/n}$$

time \leftrightarrow freq.

coeff \leftrightarrow eval. point.
(w/basis ω, ω^2, \dots)



$$\text{DFT}_{\omega}[f] = \{f(0), f(\omega), f(\omega^2), \dots, f(\omega^{n-1})\}$$

$$f(\omega^j) = a_0 + a_1(\omega^j) + a_2(\omega^{j^2}) + \dots + a_{n-1}(\omega^{j^{n-1}})$$

• This can be computed in $O(n \log n)$

• n is a power of 2.

• Finding ω

• check if ω is an n th root of unity?

$$\omega^n \stackrel{?}{=} 1$$

• Lagrange $\Rightarrow \omega$ is an n th root of unity

ex. $p = (2^k \cdot q) + 1 \quad n = 2^k$

