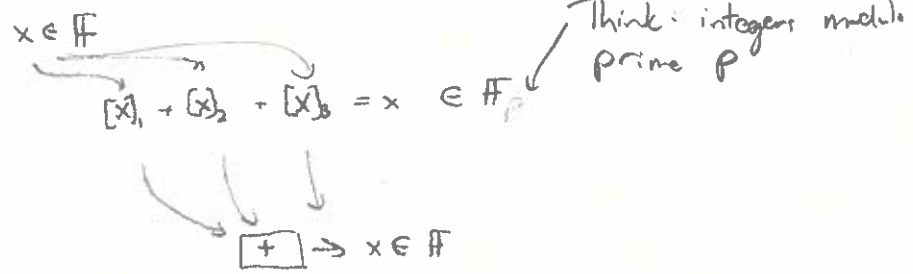
Today: From MPC to Zero Knowledge April 25, 2018

Logistics
- HW 2 due friday 5pm
- Course Staff traveling to Eurocrypt next week:
  - check Piazza for OH changes!
- OH Today & Tomorrow

Plan
- Recap: MPC with correlated randomness
- From MPC to ZK
- Application to PQ sigs

Recap: Secret Sharing and MPC

- Simple secret sharing: additive



- Also covered fancier secret sharing schemes (Shamir)

Want to spend some time reviewing MPC

Most MPC protocols view computation as a circuit (wlog)
Boolean - AND, OR, NOT (mod 2)
Arithmetic - $\times$, + (mod p)

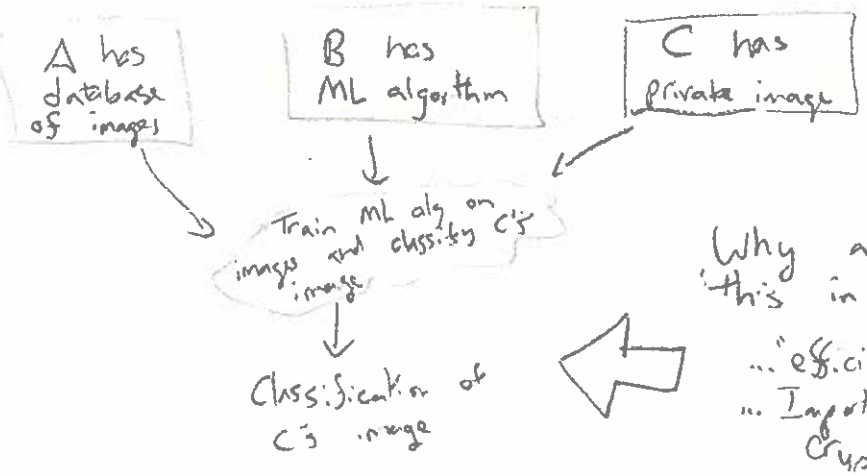
Each party i has input x_i, parties want to jointly compute
$f(x_1, \ldots, x_n)$
such that no coalition of evil players learns anything apart from
- $f(x_1, \ldots, x_n)$
- their private input

[Think about how you would formalize this]
↳ We'll see in a sec

We aim for "semi-honest" security
↳ "honest but curious" evil players
↳ Bogus in reality.... need more work to get "full"/"malicious" security

Implication: Anything that n people can compute (i.e. ppt), can compute securely. (w/ private inputs)
***In theory

e.g



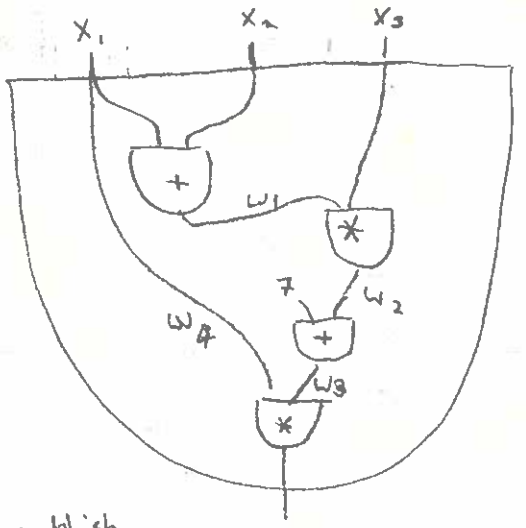
Idea of MPC Protocol David discussed (Beaver ckt randomization)

Players 1, 2, 3 get "correlated randomness"

Example: Player i has private input $x_i \in \mathbb{F}$

Want to compute

$$f(x_1, x_2, x_3) = x_1\left((x_1 + x_2)x_3 + 7\right)$$



Idea: Players jointly walk through each wire in ckt, compute an additive secret sharing of wire

→ Once have share of output, can publish and reconstruct

I. Get share of input

Split x_i into $[x_i]_1, [x_i]_2, [x_i]_3$,

send each share to player i

II. Get share of internal wires

$[w_1]_i = [x_1]_i + [x_2]_i$ ← Each player does this

Use correlated randomness to get share of w_2

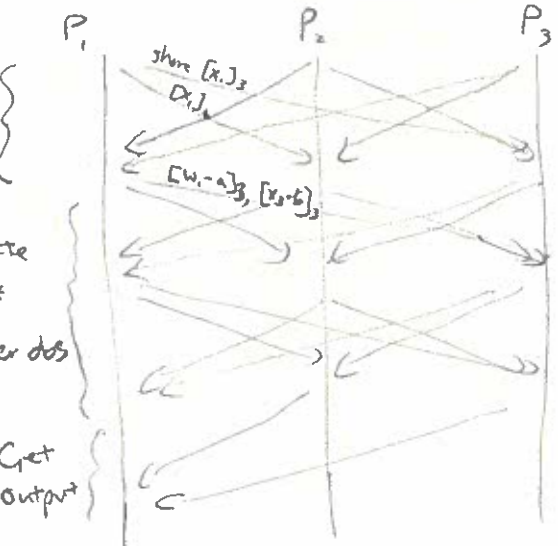
$[w_2]_i = \{$ Protocol David described plus broadcast

$[w_3]_i = [w_2]_i + 7/3 \quad \in \mathbb{F}$

⋮

III. Get output

Players send output shares to P_1



Have $[a]_i, [b]_i, [c]_i$ s.t. $ab = c \in \mathbb{F}$

1) Broadcast $[w_1 - a]_i, [x_3 - b]_i$

2) Compute $d = w_1 - a$
$e = x_3 - b$

3) Set
$[w_2]_i = [w_1 \cdot x_3]_i$
$= [c]_i + [w_1]_i e + [x_3]_i d - e \cdot d$

A couple of notes

1. Where does correlated randomness come from?
   - ↳ Can generate it using SHE or MPC (see HW)
   - ↳ Semi-trusted dealer (see later in course)
2. Protocol cost (in terms of broadcast) scales w/ depth of ckt
3. Must represent fn f as a ckt
   - ↳ Not efficient... lots of optimization

e.g. Recent work on 2PC, semi-honest

256 neuron, 2 hidden layers ⇒ 7 hours on tAN
1000 dim linreg on 1000 pts ⇒ 158sec
↳ 0.1 s on my laptop

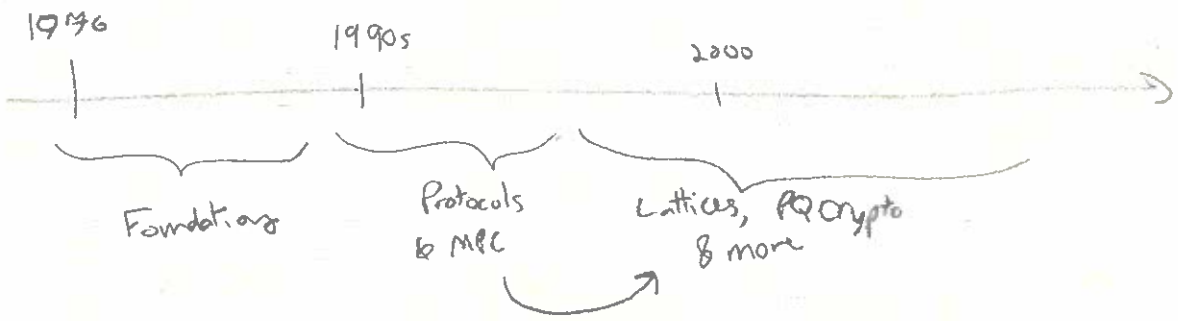
{Zhang & Mohassel '17} Many optimizations

Florian: 1.5 sec on laptop

Bottom line

For simple functions, can actually implement MPC
↳ Allows "best possible" security for computation on secret data

Big Picture



MPC in the Head
- connects MPC and ZK in surprising way
- Has application to ZK crypto
- Nice bridge to next section of course
  ↳ Also just a really nice idea

MPC in the Head (IKOS '07)

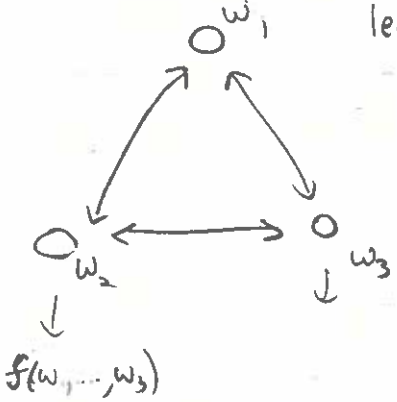
We will show that it is possibly to "efficiently" compile any general MPC protocol into a zeroknowledge proof system for an NP relation.

↳ Surprising/cool connection b/w primitives

Recall: MPC

Each player i has input $w_i \in \{0,1\}^\ell$

At the end of the protocol, each player learns $f(w_1, \dots, w_n)$ "and nothing else."



Security notions for semi-honest MPC, honest majority

1) Correctness: $\forall$ inputs, $\forall$ randomness, all players output $f(w_1, \dots, w_n)$ on input $(w_1, \dots, w_n)$

2) Privacy/Zk: $\exists$ ppt sim s.t. $\forall$ inputs $\forall$ sets $T \subseteq [n]$ of corrupted players s.t. $|T| \leq n/2$

$$\left\{ \text{View of players in } T \right\} \approx \left\{ \text{Sim}\left(T, (w_i)_{i \in T}, f(w_1, \dots, w_n)\right) \right\}$$

(Intuition the only thing that leaks is $f(w_1, \dots, w_n)$.

⇒ Can achieve this information theoretically using BGW protocol
- 5 players/parties
- 2 corruptions tolerated (again – semi-honest model)

↳ Similar to (but slightly diff from) the protocol we've seen

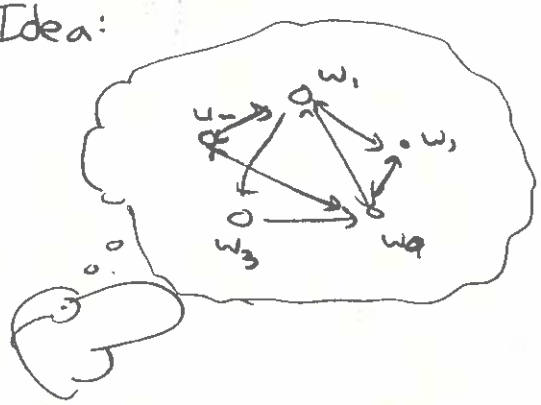
MPC in the Head Construction

e.g.
x = Sat formula ϕ
w = Sat assignment

We build ZK pf system for NP relation $R(x, w)$.
↳ Recall: defn of ZK proofs
Define $f_x(w_1, \ldots, w_n) = R(x, w_1 \oplus w_2 \oplus \cdots \oplus w_n)$.

↳ $f_x(\cdot)$ outputs $\begin{cases} 1 & \text{if fed shares of a witness } w \text{ s.t. } R(x,w)=1 \\ 0 & \text{otherwise} \end{cases}$

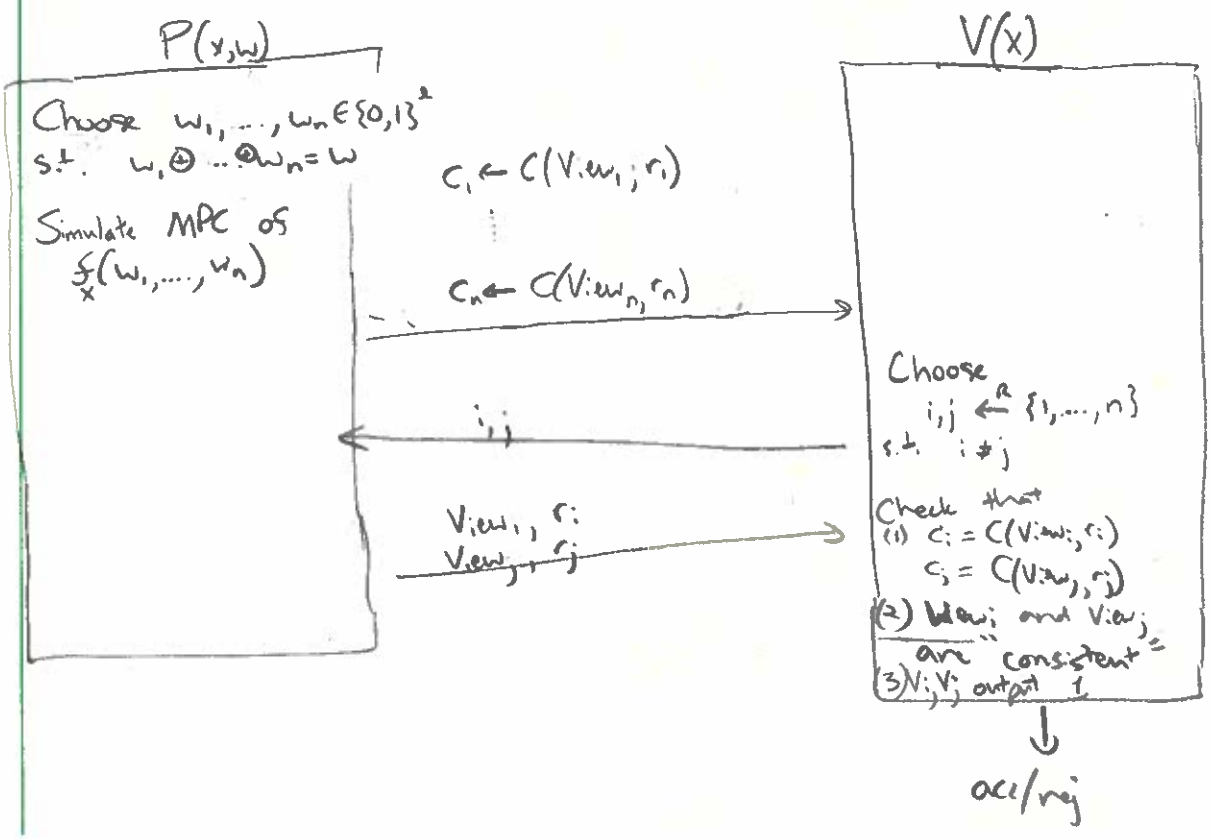
Idea:



Prover for ZK pf system imagines 5 parties running MPC of f_x
↳ Each party i's input is a share w_i of NP witness w

Uses simulated MPC to convince V that $x \in R(\mathcal{L})$

Protocol
- Let $C: \{0,1\}^* \to \mathcal{C}$ be a commitment scheme
- We will use $n=5$ party protocol



MPC in the Head

$View_i$ includes * input w_i to party.
* all msgs sent & recvd by party i
* all randomness of party i

Say that 2 views $View_i$, $View_j$ are consistent if

every msg sent by i to j is identical to msg received by j from i (and vice versa). (And i, j followed protocol.)

Claim MPC in the head ZKP is correct, sound, ZK.

Correctness: By construction.

ZK: Need to construct a simulator S s.t. $\forall V^*$

$$\{\text{Verifier's View}\} \stackrel{c}{\approx} \{S^{V^*}(x)\}$$

$S(x)$ {
Guess (i,j) that V^* will ask for
$w_i, w_j \leftarrow \{0,1\}$
$(View_i, View_j) \leftarrow Sim_{MPC}(\{i,j\}, \{w_i, w_j\}, 1)$
$c_i \leftarrow C(View_i)$
$c_j \leftarrow C(View_j)$
c_k for $k \notin \{i,j\}$ is commit to 0^λ
$(i^*, j^*) \leftarrow V^*(c_1, \ldots, c_n)$
if $(i,j) \neq (i^*, j^*)$ abort
else
output
$\langle (c_1, \ldots, c_n), (i,j), \begin{pmatrix} View_i, r_i \\ View_j, r_j \end{pmatrix} \rangle$

MPC in the Head

Has knowledge, but let's just show soundness

* If $x \in \mathcal{L}(R)$ then $\forall w_1, \ldots, w_n \in \{0,1\}^*$ honest MPC players output 0 (by MPC correctness)
* Either
  - all views have output "0"
  - $\exists$ pair (i^*, j^*) of views that are inconsistent.
* $\Pr[\text{open malformed views}] \geq \frac{1}{\binom{n}{2}} \geq$ constant for (S, d) protocol

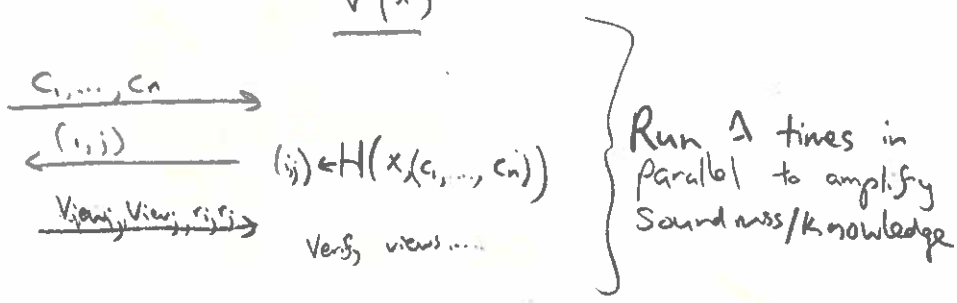
($\binom{n}{2}$: # parties)

MPC in the Head & PQ Crypto

Can use MPC in the head to prove knowledge of preimage of OWF

$$R(x,w) = \begin{cases} 1 & \text{if } x == SHA256(w) \\ 0 & \text{o.w.} \end{cases}$$

MPC-in-head is a 3-round Sigma protocol
↳ Can convert to signature scheme in ROM a la Schnorr!



A signature scheme:

$Gen(1^\lambda) \rightarrow$ sample $w \xleftarrow{R} \{0,1\}^\lambda$
set $x \leftarrow H_1(w)$
output $(pk = x,\ sk = w)$

$Sign(sk, m) \rightarrow$ compute tx of MPCIH protocol
where challenge $(i,j) \leftarrow H(x, (c_1, \ldots, c_n), m)$
for random oracle H

$Verify(pk, m, \sigma) \rightarrow$ Run MPC in head verifier
using $(i,j) \leftarrow H(x, (c_1, \ldots, c_n), m)$ as challenge

$\Rightarrow$ Security essentially follows same argument as we used for Schnorr

TC

Why do we care?!

→ Signatures from only symmetric-key primitives (AES/SHA2)
  ↳ Compare w/ RSA, Schnorr
  ↳ Prove/verify time ≈ 50 ms using SHA2 w/ 80 bit security
- No known polytime Q attacks on these primitives
  ↳ PQ secure signatures from "standard assptions" in ROM

Limitations

Sig size: ~1 MB for 80 bit security
  ↳ Have to open ≈ 80 views
  ↳ Each view is $\lambda(|C|)$ bits long
  ↳ Lots of bits!

This is an active line of research (also happening at Stanford)
  ↳ In theory can get very short PQ signatures in ROM from SHA2 ↳ $O(\text{polylog}(|C|))$
  In practice, most hash-based sigs have relatively long signatures

  See Ligero (CCS '17) w/ $O(\sqrt{|C|})$-sized sig