DC-Net, Mixnet

Slides adapted from here, here

Logistics

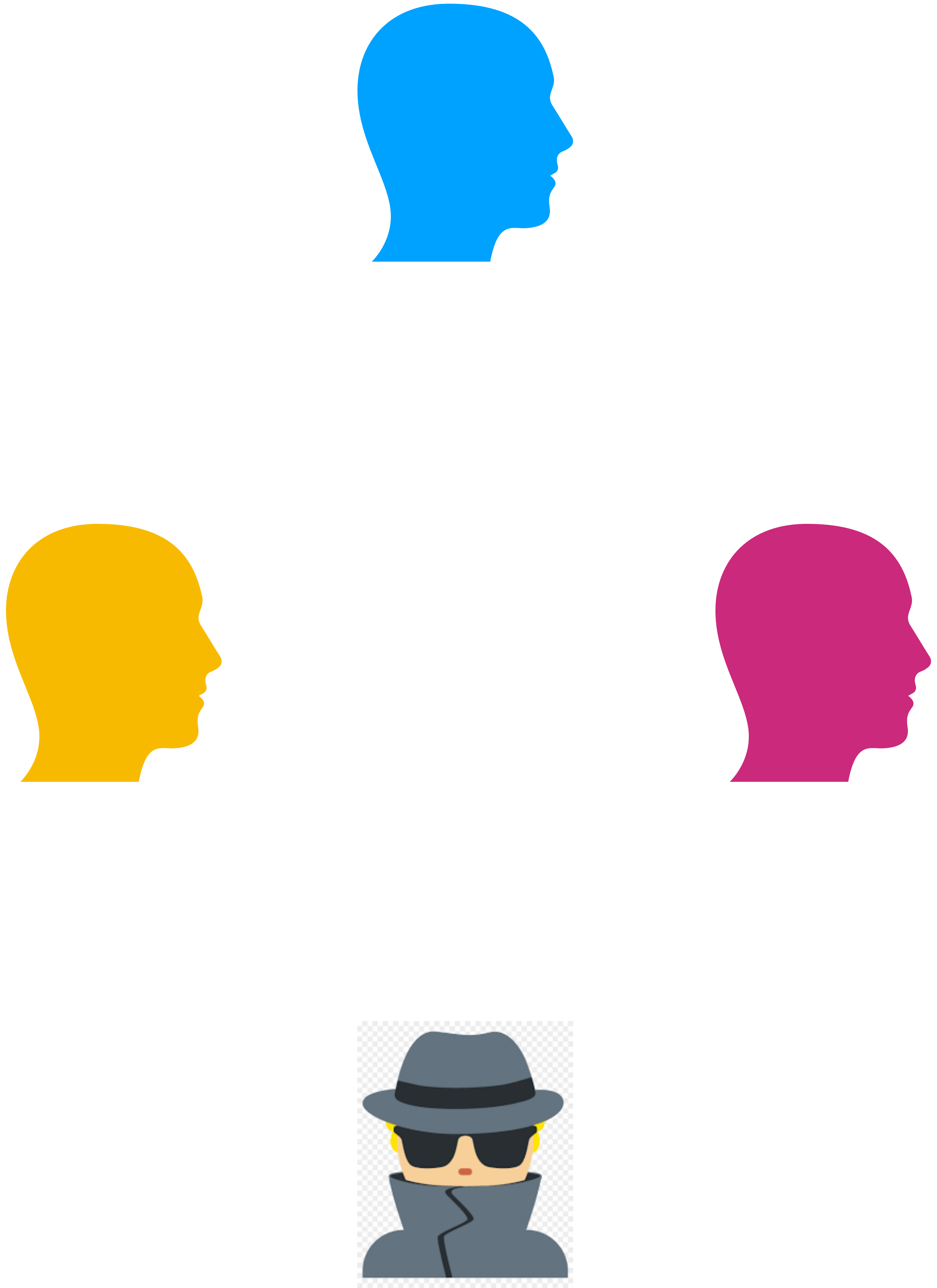
- Project proposal due midnight tonight!
  - 1 - 2 pages, in the same latex format
  - Write about 3 things
    - Problem being tackled
    - Technical approach
    - Evaluation plan
- Mid-project check-in will be during the week of October 25

Logistics

- Paper reviews
  - A new optional section for asking clarifying questions (presenters should answer these in the presentation)
- In-class discussion:
  - Will dedicate more time (should start around 11 am)
  - Will keep group discussion
  - In-class participation is 30% of the grade

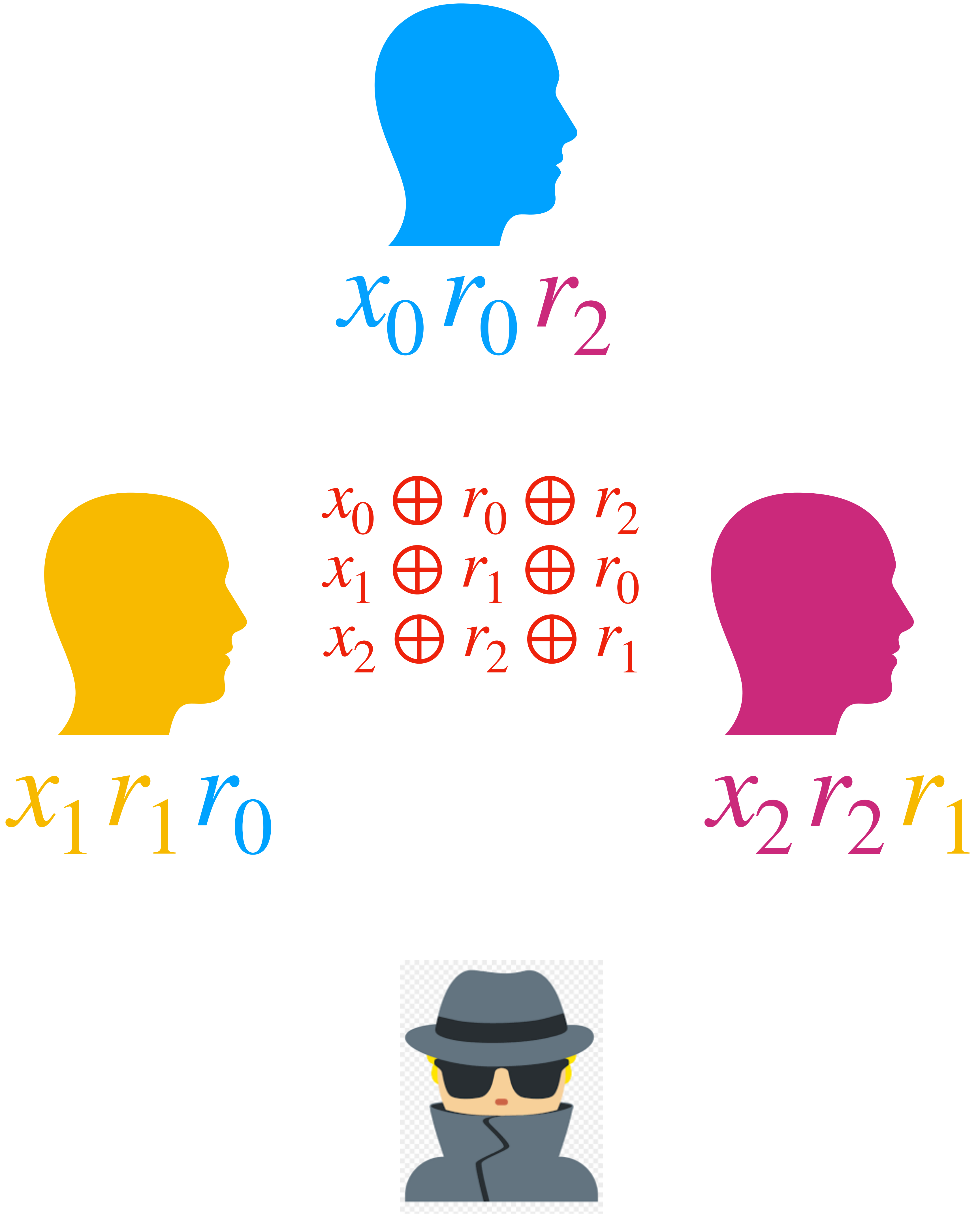
Dining cryptographers problem

- David Chaum's protocol from 1988
- Three cryptographers sitting around a table at dinner
- After dinner, they're told that *someone* has paid for dinner
- However, they wish to find out some information about who has paid for dinner
    - Is it one of the cryptographers, or the NSA?
    - They respect each other's privacy



DC-Net

- Each cryptographer has input $x_i \in \{0,1\}$
- Also flips a coin and shows the result to the cryptographer to their right
- Each person sees their own coin, plus one more coin from the person to the left
- If did not pay -> announce “same coins”/ “different coins”, else announce opposite of that
- Odd # of differences = cryptographer paid, otherwise NSA paid
- **Unconditionally secure anonymous broadcast**
  - What are the collusion assumptions?



DC-Net

- Extending to more parties?
  - Each player i has input bit x_i
  - Each player i shares secrets $r_{i1}, \cdots, r_{in} \in \{0,1\}$ with all players
  - XOR gives the sum $\sum_i^n x_i \mod 2$, but can extend to work in a larger modulus instead
- Sending longer messages?
  - Heuristic idea: use the protocol to implement a shared anonymous broadcast channel; if collision detected, use exponential backoff

DC-Net

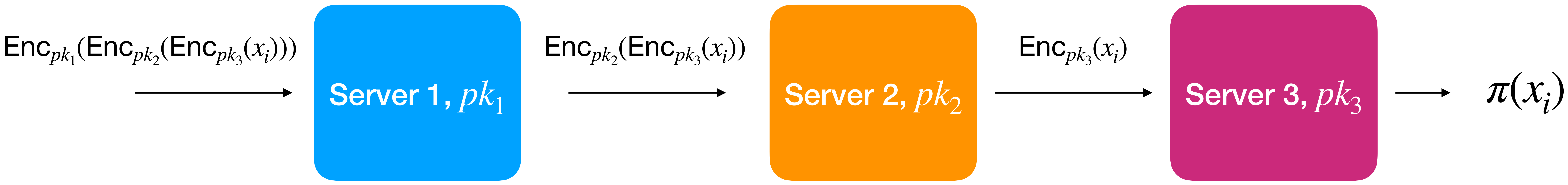
- Using DC-Net in practice for anonymous broadcasting is very expensive
    - Robustness: any party goes offline, then the messages are unrecoverable
    - Communication: each party needs to send n bits
    - Message recovery: n^2 total work (each party needs to reconstruct using n bits)

Mixnet

- Another idea proposed by Chaum
- Each party i wants to broadcast message $x_i \in \{0,1\}^l$
- All parties want to learn $\{x_1, \cdots, x_n\}$, except in *shuffled order*
- Delegate work to k servers that repeatedly shuffle the messages

Mixnet

- Each player i encrypts message x_i using the servers' public keys
  - $c_i = \text{Enc}_{pk_1}(\text{Enc}_{pk_2}(\text{Enc}_{pk_3}(x_i)))$
- Each server shuffles, decrypts, and passes to the next server
- Output is in a shuffled order such that no servers knows the final permutation



Mixnet

- Mixnet vs. DC-net
  - Per user communication: 1 cipher text vs. n bits
  - Total compute: n public key operations vs. n^2 field operations
  - Security: computational vs. information theoretic

Today’s reading: anonymous messaging using Vuvuzela

Next (few) classes: secure multiparty computation

- DC-Net is actually a type of secure multiparty computation!
  - Multiple users, each has a private input
  - Protocol can securely compute the sum (XOR) function given “random shares” of an input bit
- In general, secure multi-party computation (MPC) is a way to jointly compute on different parties’ private inputs, without revealing
  - Each party’s private input
  - Intermediate results of the computation

Next class: secure aggregation for federated learning

- Many techniques for doing so: **secret sharing,** homomorphic encryption, garbled circuits, etc.
- Wednesday's reading
  - A different type of sharing compared to DC-net (robustness)
  - Setting is not quite MPC, and instead is something called "federated learning"
    - What are the security guarantees?
    - Is there any extra leakage?