CS 594 – Advanced Crypto (Spring 2026)

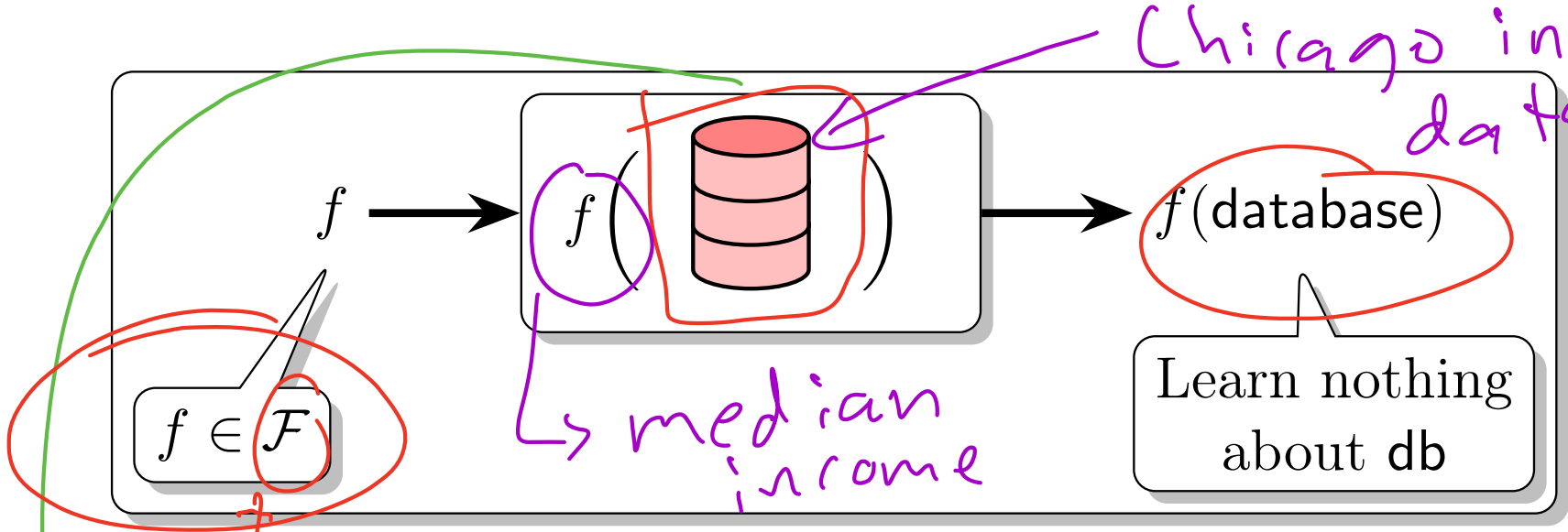
Alex Block

Lecture 15

March 09, 2026

What is Secure Computation?

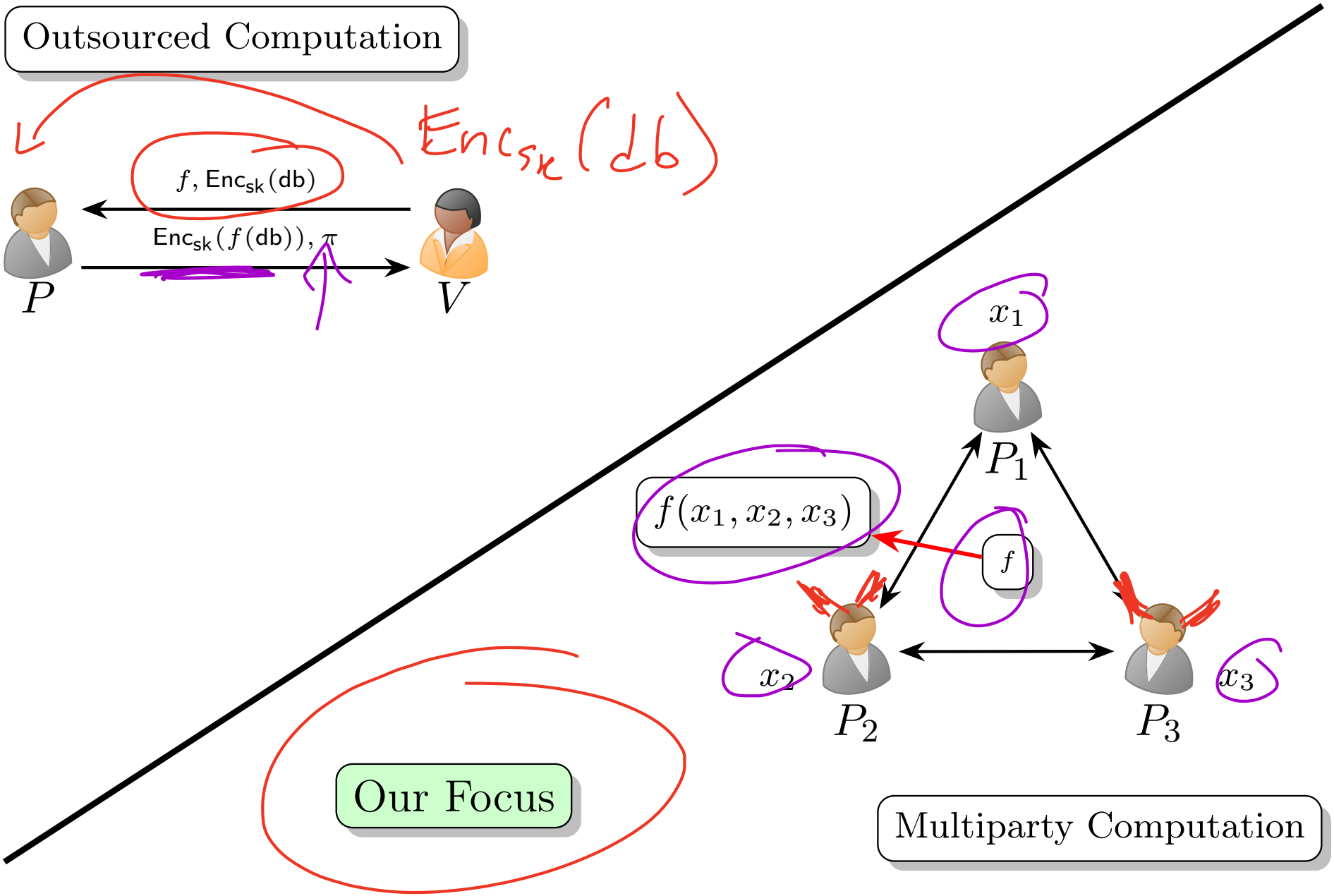
Secure Computation, Informally



Secure Computation

- Secure Computation = computing over data while keeping that data secret.

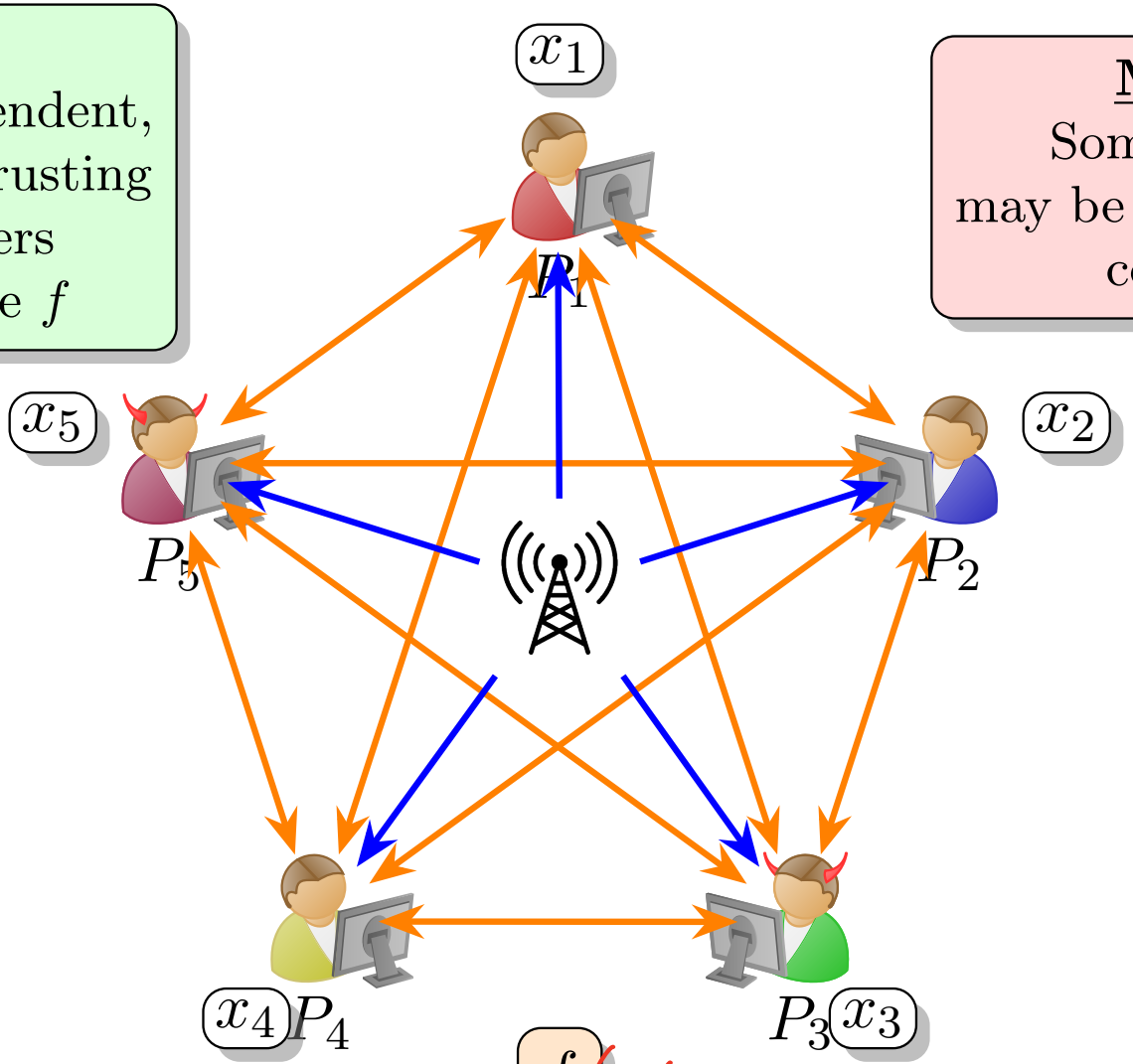
Outsourced vs. Multiparty Computation



Secure Multiparty Computation (MPC)

Goal
Enable independent, mutually distrusting data owners to compute f

Model
Some parties may be (maliciously) corrupt!



Brief History of MPC

- Idea introduced by Andrew Yao in early 1980s.
  - Introduced general idea of MPC.
  - m parties $P_1, \ldots, P_m$ wish to jointly compute $f(x_1, \ldots, x_m)$.
  - Party P_i has private input x_i.
- Yao developed the famous Garbled Circuits Protocol in a series of talks over the following few years.
  - Remains the basis for many of the most efficient MPC implementations.
- MPC remained purely theoretical until early 2000s.
  - Fairplay 2PC system in 2004 was first notable implementation of general-purpose MPC.
  - Still limited at the time: only could compute the median of two sorted arrays, each with ten 16-bit numbers.

Generic vs. Specialized MPC Protocols

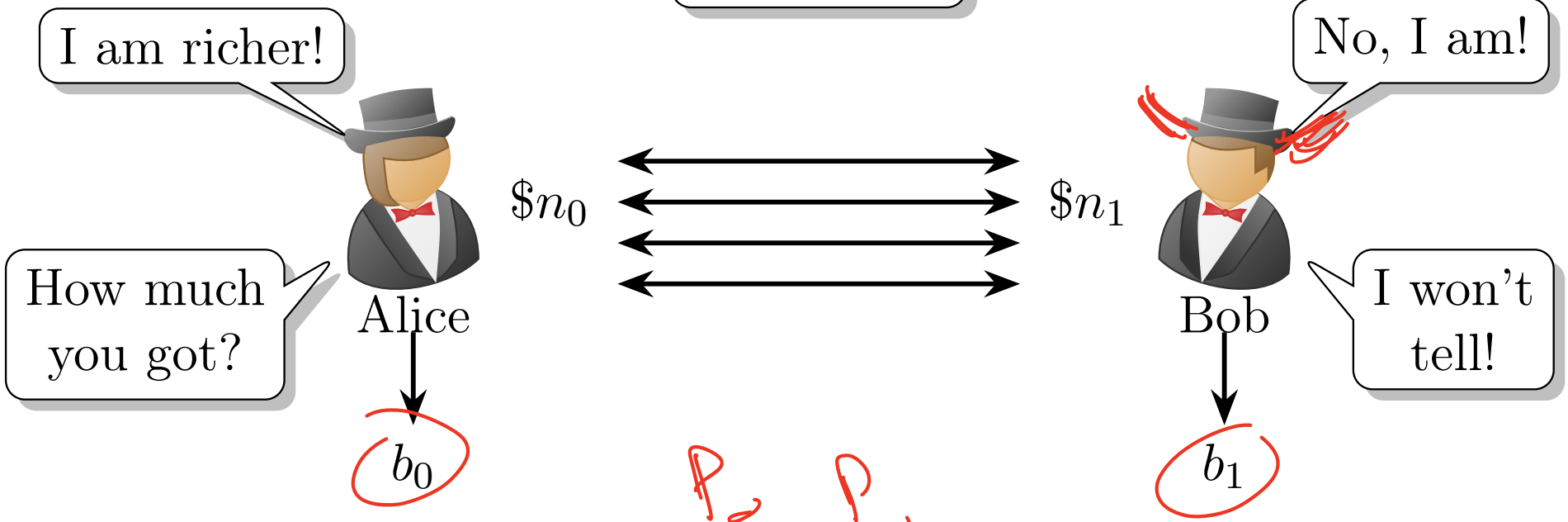
- Yao's GC protocol (and many other MPC protocols) are *generic* or *general purpose*.
  - I.e., they can compute any discrete function with a fixed-sized circuit representation.

- Important MPC sub-area: protocols for *specialized functionalities*.
  - E.g., private set intersection (PSI).

- Though many "specialized" MPC protocols do not outperform general ones, we'll discuss PSI protocols later, as these protocols can outperform generic ones.

MPC Applications

- Yao's Millionaires Problem.

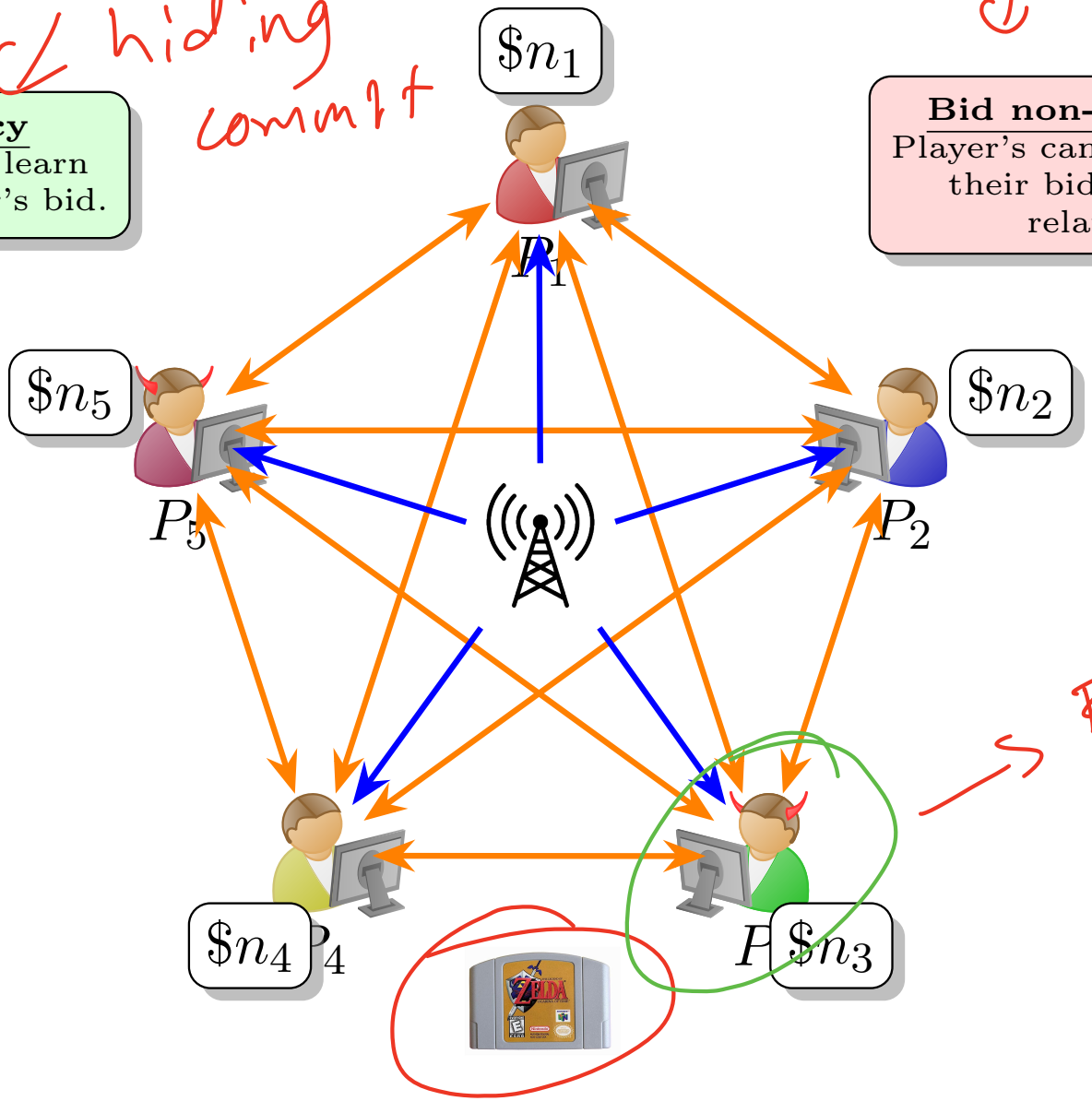


MPC Applications

- Secure Auctions.

Bid Privacy
No player may learn any other player's bid.

Bid non-malleability
Player's cannot manipulate their bid to generate related bid.



MPC Applications

- Voting.
  - Functionality: simply tallies votes for each candidate in election.
  - Privacy and non-malleability of the votes are *essential*.
  - Not standard MPC definition but needed: *coercion resistance*.

- Secure Machine Learning.
  - *Oblivious model inference*: client $\mathcal{C}$ requests to a server $\mathcal{S}$ which has a pretrained model.
    - Request should be kept private from $\mathcal{S}$, model should be kept private from $\mathcal{C}$.
  - Useful in *model training* as well.
    - Groups of users can jointly train a model with their own private data without revealing their data.

MPC Applications

- Other Applications.
  - Privacy-preserving network security monitoring.
  - Privacy-preserving genomics.
  - Private stable matching.
  - Contact discovery.
  - Ad conversion.
  - Spam filtering on encrypted email.

MPC Deployment

- MPC has been deployed but still faces several key issues.
  - Building trust and confidence in deployed system.
  - Understanding what *sensitive* information may be inferred from the output of an MPC protocol.
  - Informing decision makers who must protect sensitive data but do not have technical crytographic background on the security implications of participating in an MPC protocol.

- Current deployments of MPC act as *enablers* of data sharing.
  - Those using MPC are not doing so to add privacy.
  - MPC being used to enable a feature or an application, which would otherwise be impossible.

MPC Deployment

- Danish sugar beets auction (2009).
  - Widely considered first real-world MPC deployment.
  - Danish researchers collaborated with the Danish government and relevant stakeholders to conduct auction of sugar beets.
  - 3PC between Dansico (only company in Denmark which processes sugar beets), the farmer's association DKS, and the researchers/government.
  - Lead to the creation of company Partisia, which uses MPC to support secure auctions and data exchange.

MPC Deployment

- Estonian Student Study (2015).
  - In 2012, 43% of IT students failed to graduate.
  - Government wanted to see if this was due to aggressive hiring by IT companies.
  - Issue: looking at education and tax records was prevented by privacy legislation.
  - Company Cybernetica deployed a 3PC solution for the data analysis.
    - Found no correlation between graduation failure and working during studies.
    - Did find correlation between more education and more income.

MPC versus ZKPs

Zero-knowledge Proofs

- *Prover* and *Verifier* with different computational power.
- Prover's data protected, Verifier doesn't really have data to protect.
- Can enable 2PC with homomorphic encryption.
- Verifier won't learn output if Prover dishonest.
- Can build ZKPs from MPC: MPC-in-the-head paradigm.

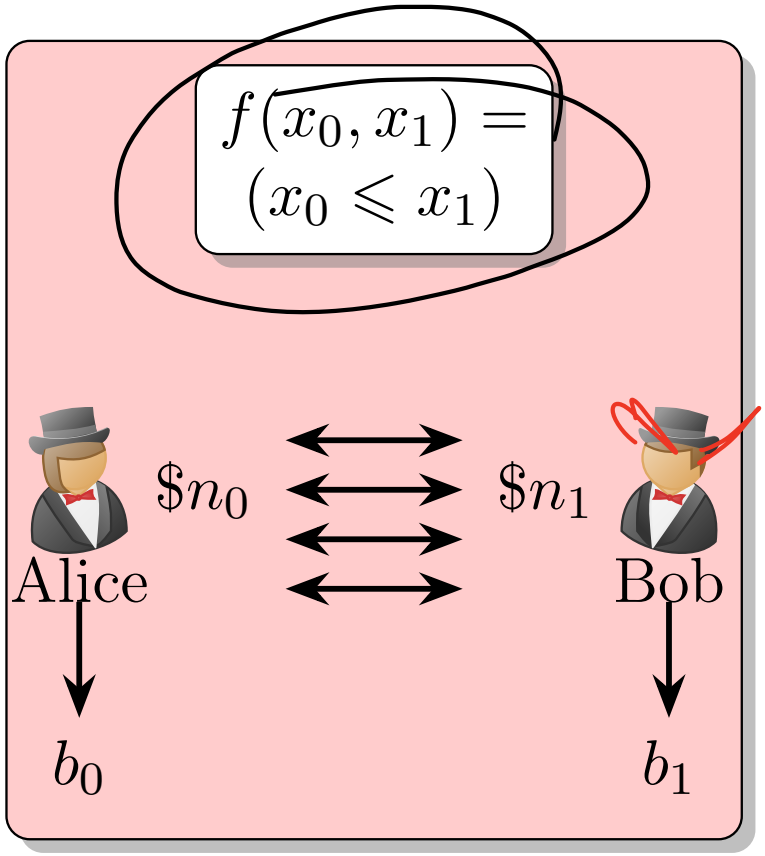
MPC

- $m \geqslant 2$ parties, each with roughly same computational power.
- All parties' data is protected.
- Can still guarantee honest parties receive correct output even with malicious parties (in some cases).
- ZKPs are often used in MPC protocols as a useful primitive.

Defining MPC

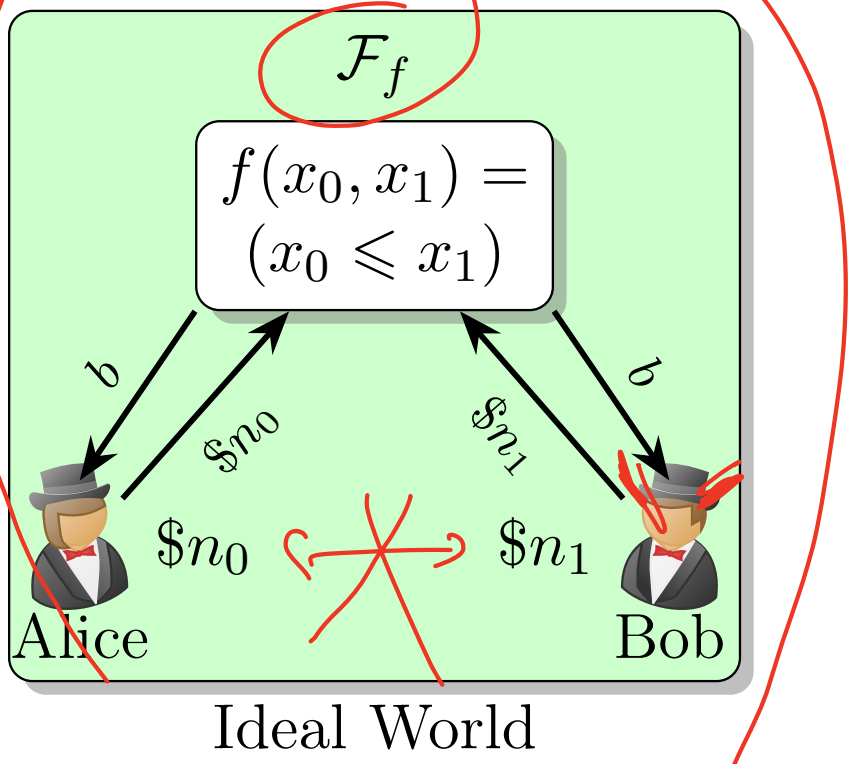
Real-Ideal Security Paradigm

- Idea: execution in the real world is *indistinguishable* from *ideal* world where function f is perfectly computed.
- Indistinguishability happens via *simulation*.



Real World

millionaire's problem



Ideal World

Semi-honest Secure MPC

- Let π be a protocol and $\mathcal{F}$ be a functionality.
- Let $C \subset [m]$ denote the set of corrupt parties in π.
- We say that π *securely realizes* $\mathcal{F}$ *in the presence of semi-honest adversaries* if there exists a simulator Sim such that $\forall C \subset [m]$ and all inputs $x_1 \ldots, x_m$, the distributions $\mathsf{Real}_\pi(C, x_1, \ldots, x_m)$ and $\mathsf{Ideal}_{\mathcal{F},\mathsf{Sim}}(C, x_1, \ldots, x_m)$ are indistinguishable.

allowable corruptions $|C| \leq t$ and $n/2$

what kind? Depends! $t \leq n/2 - 1$ (info-theoretic)

$\approx$

$\mathsf{Real}_\pi(C, x_1, \ldots, x_m)$

- Run protocol π where each party P_i honestly behaves with input x_i. , obtains y_i
- Let V_i denote the final view of P_i.
- Output $\{V_i : i \in C\}, (y_1, \ldots, y_m)$.

$\mathsf{Ideal}(C, x_1, \ldots, x_m)$

- Compute $(y_1, \ldots, y_m) \leftarrow \mathcal{F}(x_1, \ldots, x_m)$.
- Output $\mathsf{Sim}(C, \{(x_i, y_i) : i \in C\}), (y_1, \ldots, y_m)$

$\{\tilde{V}_i : i \in C\}$

Next Time: More Security Definitions, Yao's GC Protocol and the GMW Protocol