CSCI 1515 Applied Cryptography

This Lecture:

- SNARGs from PCP (continued)
- SNARGs from Linear PCP
- Introduction to MPC

Succinct Non-Interactive Argument (SNARG)

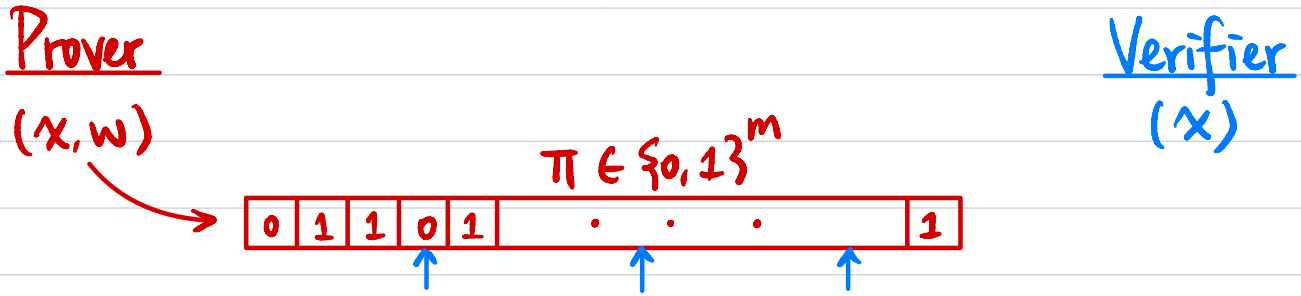
$\forall P^*$ $\forall PPT\ P^*$ (in soundness)

Def A non-interactive proof/argument system is succint if

- The proof π is of length $|\pi| = poly(\lambda, \log|C|)$
- The verifier runs in time $poly(\lambda, |x|, \log|C|)$

- SNARK: Succinct Non-Interactive Argument of Knowledge
- zk-SNARG/zk-SNARK: SNARG/SNARK + Zero-Knowledge

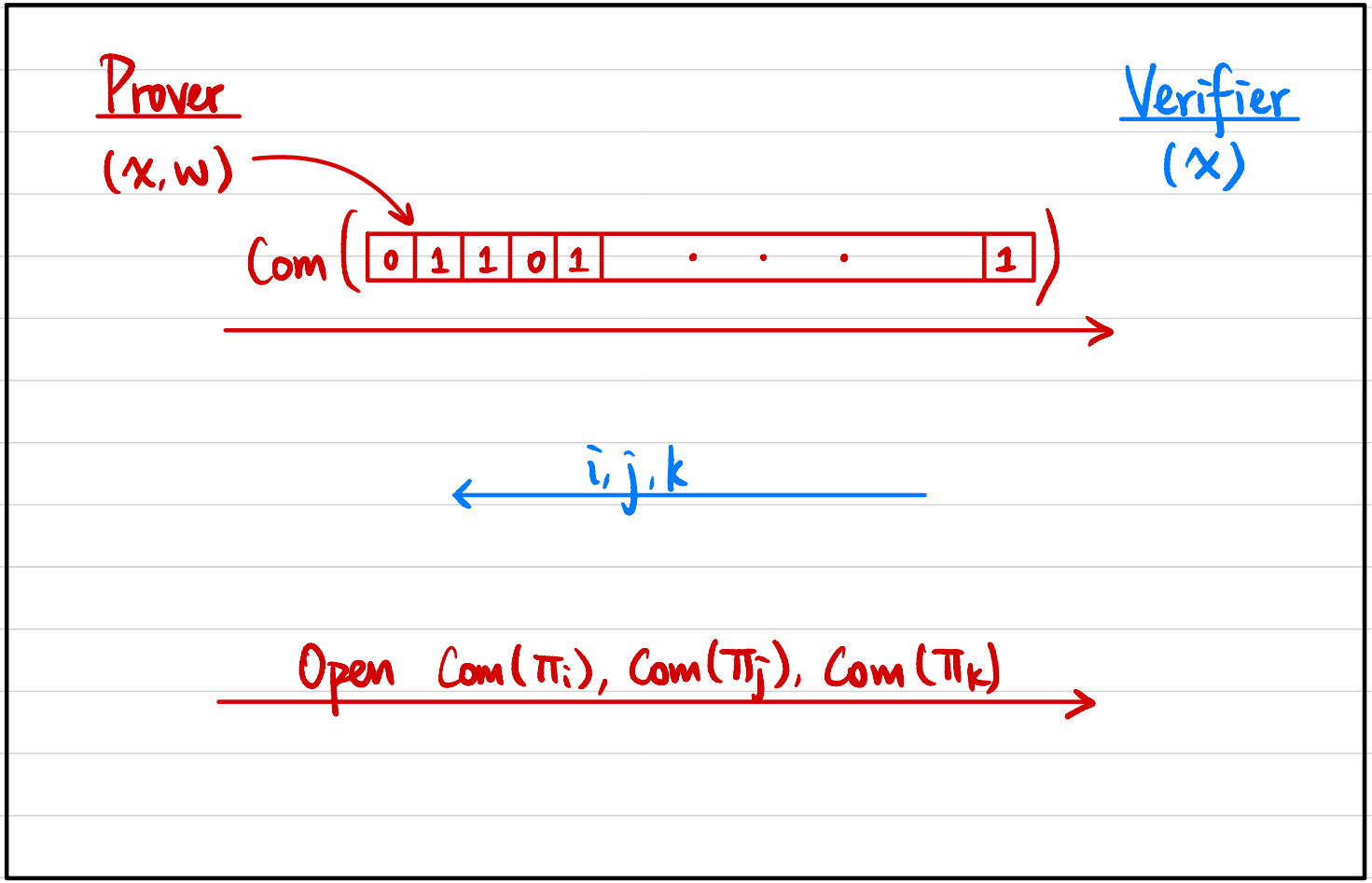
Probabilistically Checkable Proof (PCP)



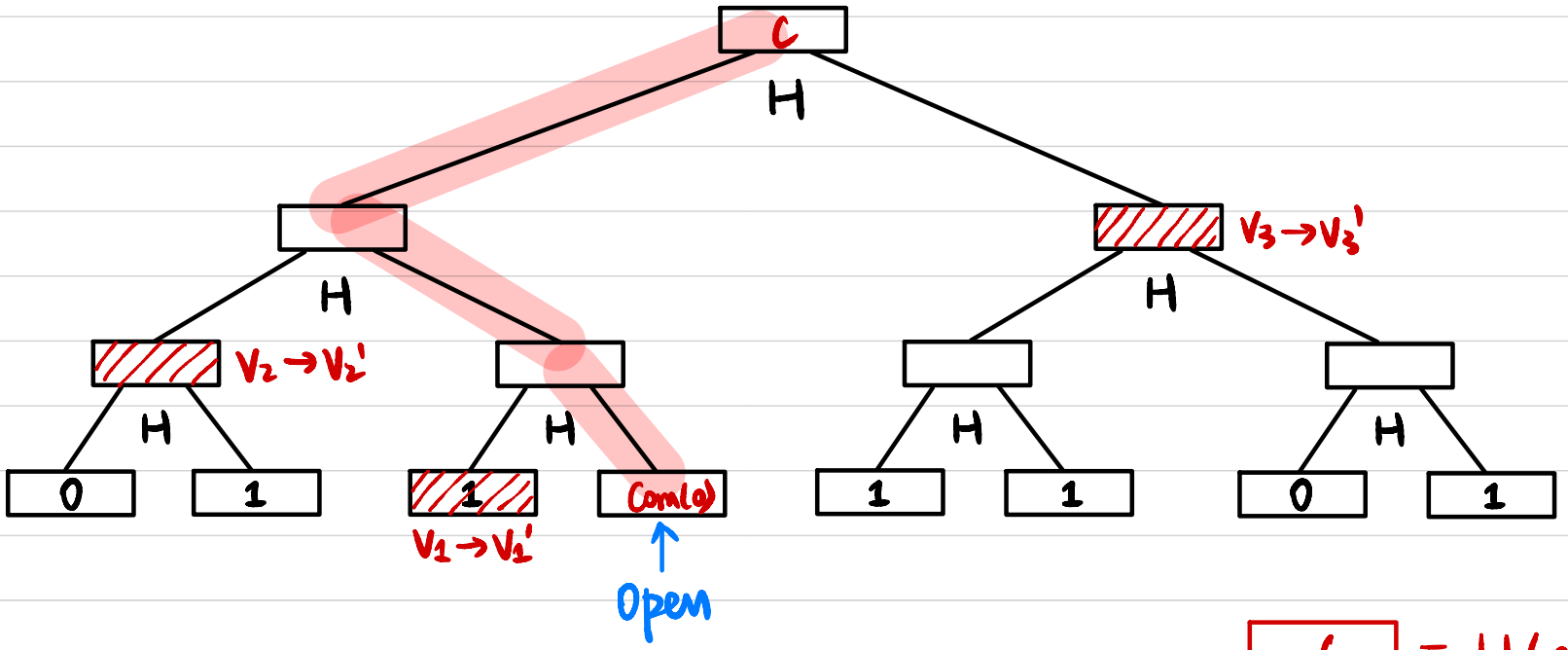
PCP Theorem (Informal):

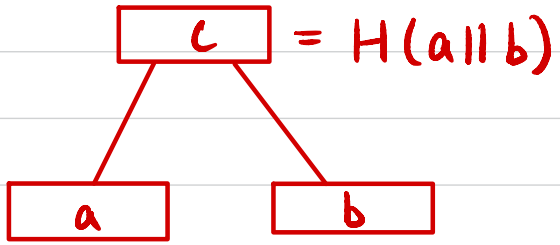
Every NP language has a PCP where the Verifier reads only a constant number of bits of the proof.

First Attempt



Merkle Tree



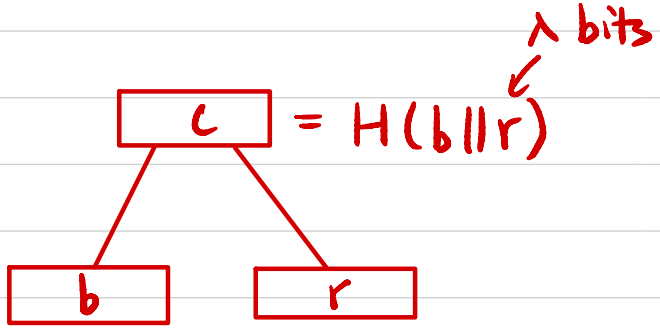


Why (computationally) binding ?

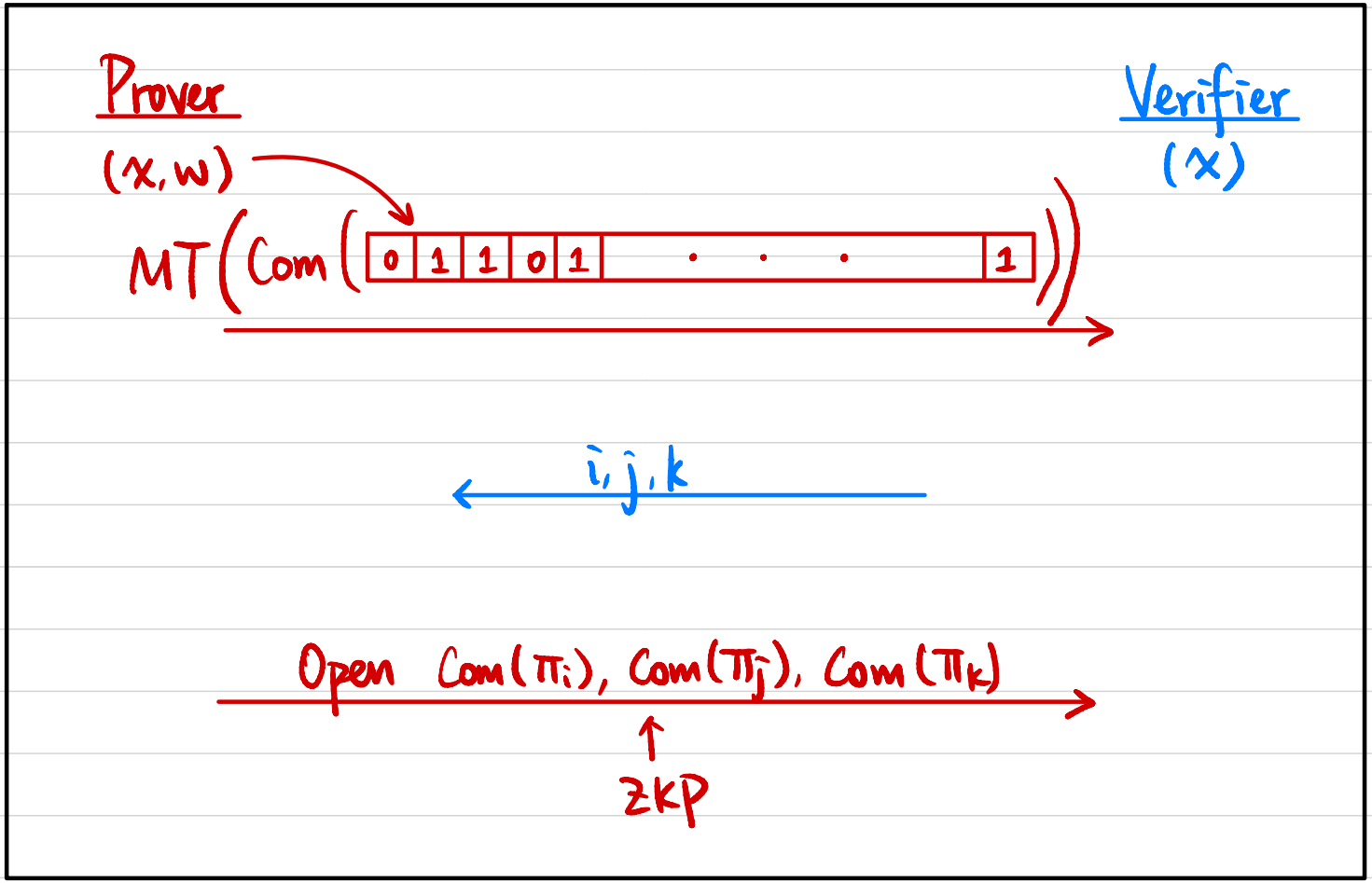
Collision Resistance of Hash

Can we make it hiding ?

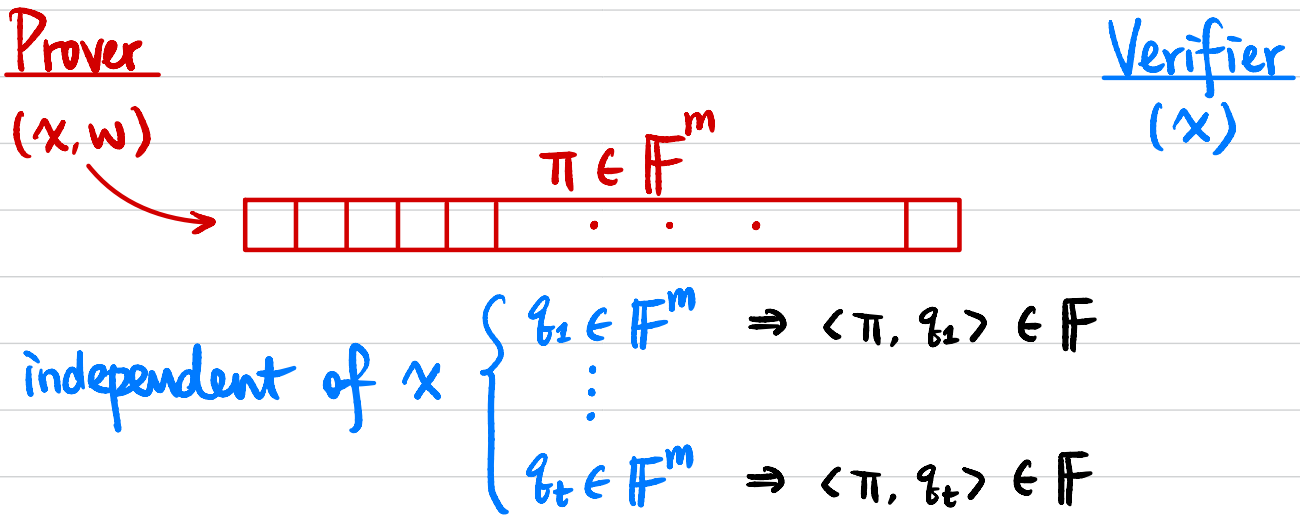
Commitment to b



Is it ZK?


Prover
(x, w)
MT(Com(0 1 1 0 1 . . . 1))
Verifier
(x)
i, j, k
Open Com(π_i), Com(π_j), Com(π_k)
ZKP

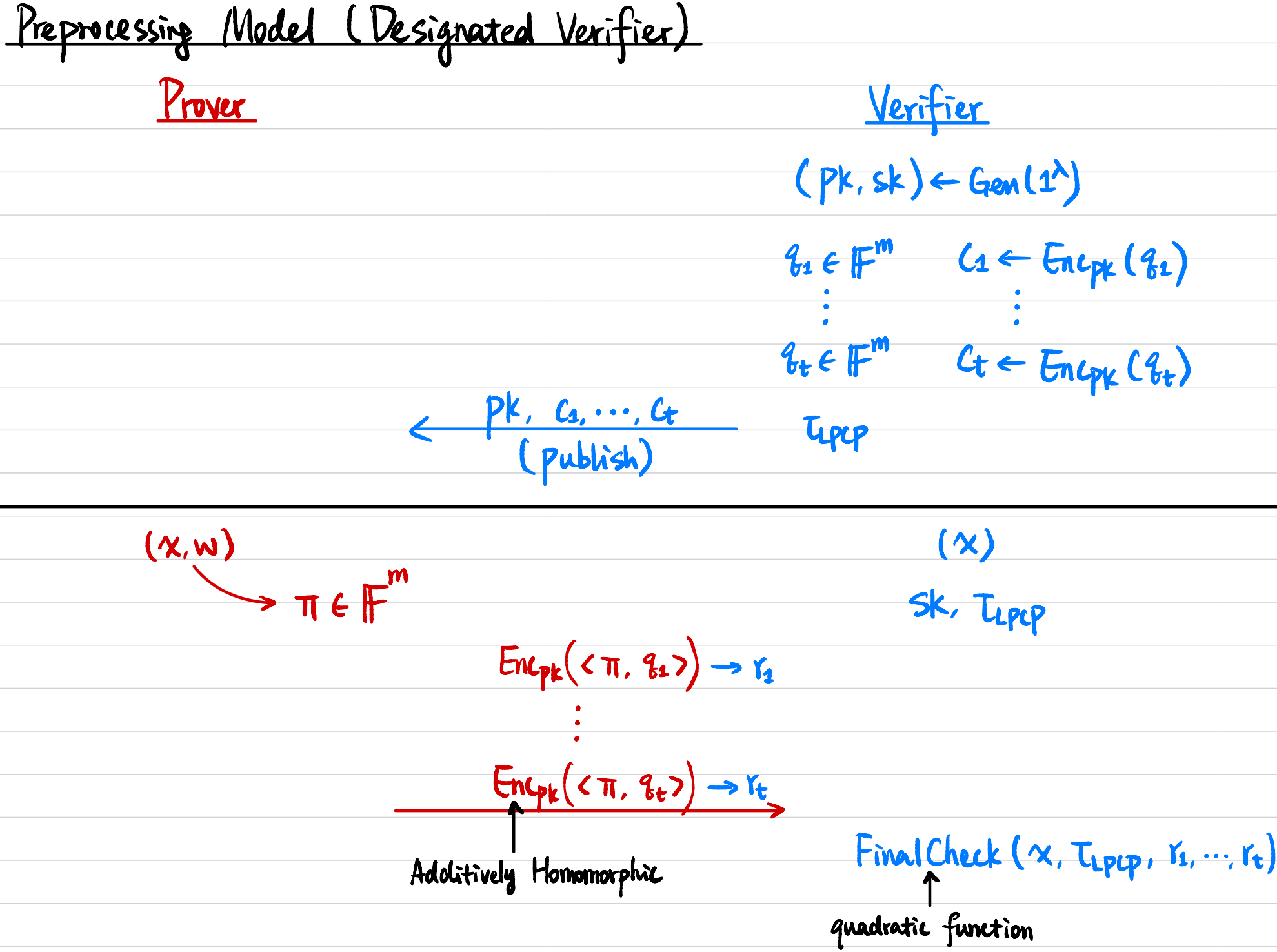
Linear PCP



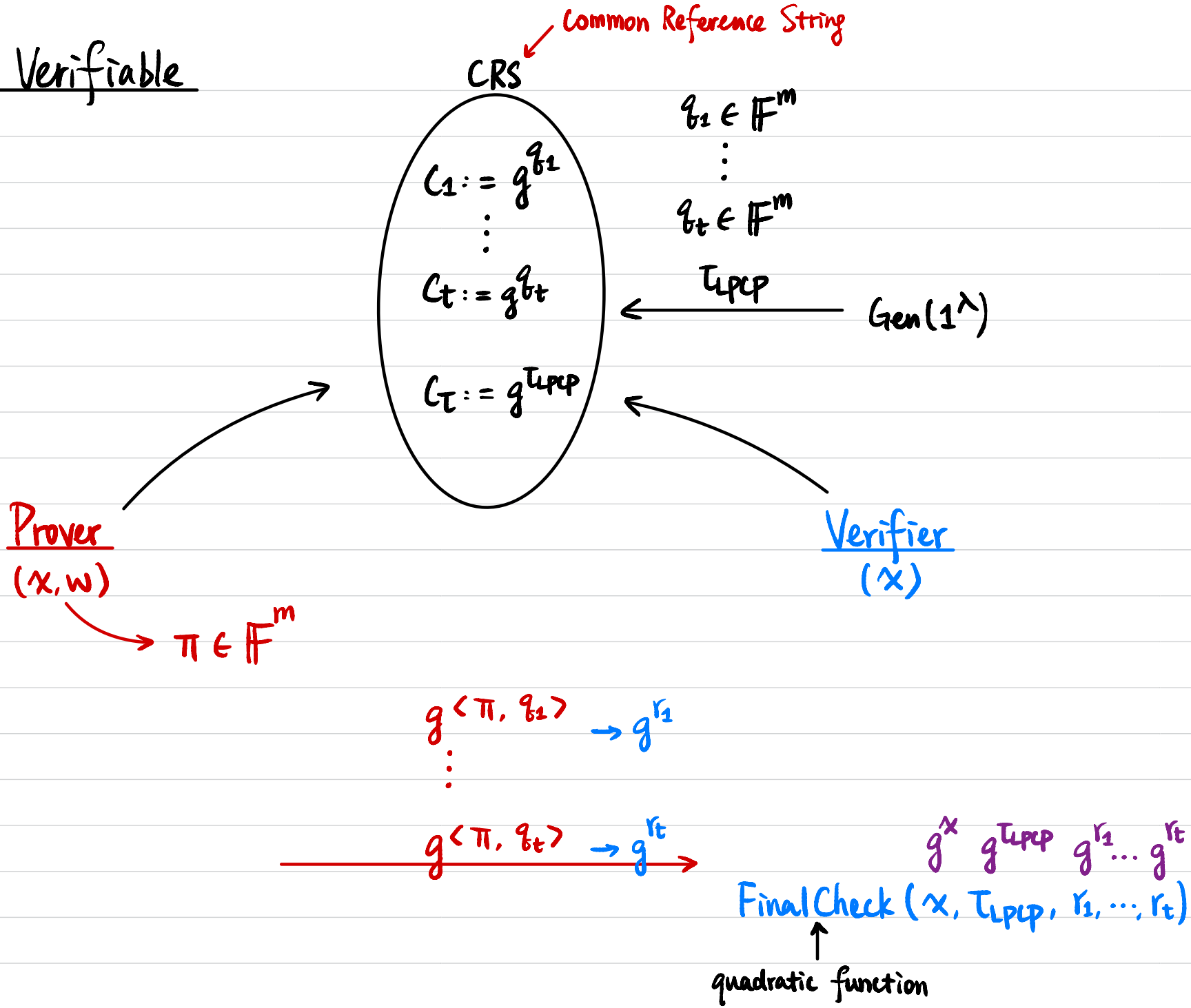
Constructing LPCP for Circuit Satisfiability:

- From Walsh-Hadard code, $m = O(|C|^2)$
- From quadratic span programs, $m = O(|C|)$

Preprocessing Model (Designated Verifier)



Publicly Verifiable



Bilinear Pairings

Asymmetric pairing

Cyclic groups G_1, G_2, G_T with generators g_1, g_2, g_T, respectively.

$$e: G_1 \times G_2 \longrightarrow G_T$$

$$e(g_1^a, g_2^b) = g_T^{ab}$$

Symmetric pairing:

$$e: G \times G \longrightarrow G_T$$

$$e(g^a, g^b) = g_T^{ab}$$

Secure Multi-Party Computation

Alice

x

Bob

y

Second date?

$f(x, y) = x \wedge y$

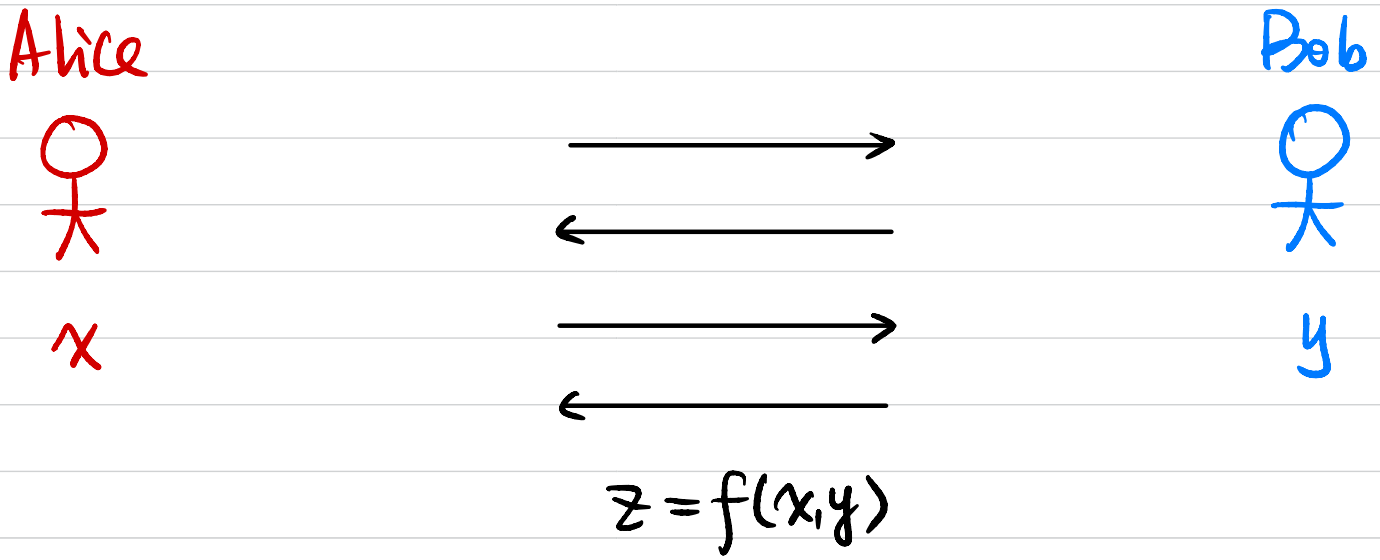
Who is richer?

$$f(x, y) = \begin{cases} 0 & \text{if } x > y \\ 1 & \text{otherwise} \end{cases}$$

Common friends?

$f(X, Y) = X \cap Y$

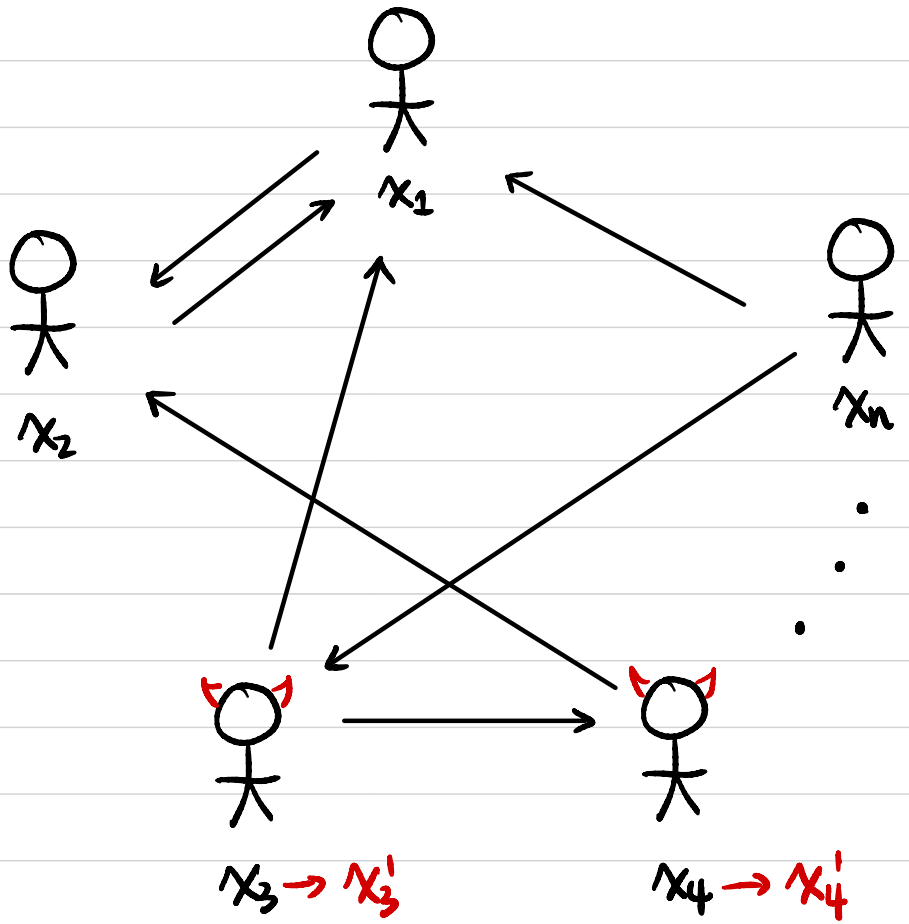
Secure Two-Party Computation (2PC)



Applications:

- Password Breach Alert (Chrome / Firefox / Azure / iOS Keychain)
- Privacy-Preserving Contact Tracing for COVID-19 (Apple & Google)
- Ads Conversion Measurements / Personalized Advertising (Google / Meta)

Secure Multi-Party Computation (MPC)



$$z = f(x_1, \cdots, x_n)$$

Secure Multi-Party Computation (MPC)

Applications:

- Privacy-Preserving Inventory Matching (J.P. Morgan)
- Setup Ceremony to securely generate CRS (Zcash)
- Distributed Key Management (Unbound / Coinbase)
- Federated Learning (Google Keyboard Search Suggestion)
- Auctions (Danish sugar beet auction)
- Boston gender wage gap (Boston Women's Workforce Council)

- Study / Analysis on Medical Data
- Fraud Detection (banks)

Setting

- n parties $P_1, P_2, \cdots, P_n$
  with private inputs $x_1, x_2, \cdots, x_n$

- Jointly compute $f(x_1, x_2, \cdots, x_n)$

- Communication:
  Authenticated secure point-to-point channels between each pair (P_i, P_j)
  (sometimes also assume broadcast channel)

- The adversary can "corrupt" a subset of the parties
  (e.g. at most t parties)

What properties do we want?

General Security Properties

- **Correctness:** The function is computed correctly.
- **Privacy:** Only the output is revealed.
- **Independence of Inputs:** Parties cannot choose inputs depending on others' inputs.

- **Security with Abort:** Adversary may "abort" the protocol.
  (preventing honest parties from receiving the output)
- **Fairness:** If one party receives output, then all receive output.
- **Guaranteed Output Delivery (GOD):** Honest parties always receive output.

Adversary's Power

Allowed adversarial behavior:

- **Semi-honest** / passive / honest-but-curious:

  Follow the protocol description honestly,

  but try to extract more information by inspecting transcript.

- **Malicious** / active:

  Can deviate arbitrarily from the protocol description.

Adversary's Computing Power:

- **Unbounded computing power** ⇒ Information-Theoretic (IT) Security
- **PPT bounded** ⇒ Computational Security