This Lecture:

- Zero-Knowledge Proofs for All NP
- Succinct Non-Interactive Arguments (SNARGs)
Zero-Knowledge Proof for Graph 3-Coloring (All NP)

NP language \( L = \{ G : G \text{ has 3-coloring} \} \)

NP relation \( R_L = \{ (G, 3\text{COL}) \} \)

If \( G \in L \), \( \Pr[ P^* \text{ is caught}] \geq ? \)

How to amplify soundness?
Commitment Scheme

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ m \in {0,1} ]</td>
<td>[ m ]</td>
</tr>
<tr>
<td>Commit:</td>
<td>[ r \in {0,1}^k ]</td>
</tr>
<tr>
<td>[ C := \text{Com}(m; r) ]</td>
<td>[ C ]</td>
</tr>
<tr>
<td>Open:</td>
<td>[ (m, r) \rightarrow ]</td>
</tr>
<tr>
<td>Verify:</td>
<td>[ C = \text{Com}(m; r) ]</td>
</tr>
</tbody>
</table>

Example: Pedersen Commitment

Cyclic group \( G \) of order \( q \) with generator \( g \), \( h \in G \)

\[ r \in \mathbb{Z}_q^* \]

\[ \text{Com}(m; r) = g^m \cdot h^r \]
Commitment Scheme

• Hiding: \( \text{Com}(0; r) \approx \text{Com}(1; s) \)
  - Perfectly hiding: \( \text{Com}(0; r) \equiv \text{Com}(1; s) \)
  - Computationally hiding: \( \text{Com}(0; r) \approx \text{Com}(1; s) \)

• Binding: Hard to find \( r, s \) s.t. \( \text{Com}(0; r) = \text{Com}(1; s) \)
  - Perfectly binding: \( \forall r, s, \text{Com}(0; r) \neq \text{Com}(1; s) \)
  - Computationally binding: Any PPT sender cannot find \( r, s \) s.t. \( \text{Com}(0; r) = \text{Com}(1; s) \)

What does Pedersen commitment scheme satisfy?

Can a commitment scheme be both perfectly hiding & perfectly binding?
Zero-Knowledge Proof for Graph 3-Coloring

Input: $G = (V, E)$

Witness: $\phi: V \rightarrow \{0, 1, 2\}$

Given a computationally hiding, perfectly binding commitment scheme.

<table>
<thead>
<tr>
<th>Prover</th>
<th>Verifier</th>
</tr>
</thead>
</table>
| Randomly sample $\pi: \{0, 1, 2\} \rightarrow \{0, 1, 2\}$

For $v \in V$, $r_v \in \{0, 1, 2\}$, $C_v = \text{Com}(\pi(\phi(v)); r_v)$

$(C_v)_{v \in V}$

Randomly pick an edge $(u, v) \in E$

$(u, v)$

Open commitments $C_u$ & $C_v$

$\alpha = \pi(\phi(u)), r_u$

$\beta = \pi(\phi(u)), r_v$

Verify:

$C_u = \text{Com}(\alpha; r_u)$

$C_v = \text{Com}(\beta; r_v)$

$\alpha, \beta \in \{0, 1, 2\}, \alpha \neq \beta$
Circuit Satisfiability (NP Complete)

NP language $L_c = \{x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ st. } C(x,w) = 1\}$

NP relation $R_c = \{ (x,w) : C(x,w) = 1 \}$

Example: pre-image of hash function

$$C(x,w) = H(w) - x + 1$$
ZKP for Circuit Satisfiability
# Proof Systems for Circuit Satisfiability

NP relation $R_{lc} = \exists (x, w): C(x, w) = 1$

<table>
<thead>
<tr>
<th></th>
<th>NP</th>
<th>Σ-Protocol</th>
<th>(Fiat-Shamir) NIZK</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(x, w) $\xrightarrow{w} V(x)$</td>
<td>P(x, w) $\xleftrightarrow{} V(x)$</td>
<td>P(x, w) $\xrightarrow{\Pi} V(x)$</td>
<td></td>
</tr>
<tr>
<td>Zero-Knowledge</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Non-Interactive</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Communication</td>
<td>$O(</td>
<td>w</td>
<td>)$</td>
</tr>
<tr>
<td>V's computation</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Can we have communication complexity & verifier's computational complexity sublinear in $|c|$ & $|w|$?
**Succinct Non-Interactive Argument (SNARG)**

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

- The proof $\Pi$ is of length $|\Pi| = \text{poly}(\lambda, \log |C|)$
- The verifier runs in time $\text{poly}(\lambda, |x|, \log |C|)$

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

**Why Succinct Proofs?**

**Is it possible?**
Verifiable Computation

**Server**

\[ x \leftarrow \text{compute } f \]

\[ y \]

**Client**

\[ y = f(x) \]
Anonymous Transactions on Blockchain

Alice's Account A → 2BTC → Bob's Account B

VK_A (public)  
SK_A (private)  

VK_B (public)  
SK_B (private)  

Transaction: VK_A, VK_B, 2BTC, s = Sign_SK_A

Anonymous Transaction:

Com(VK_A, VK_B, 2BTC, s = Sign_SK_A)

NIZK: valid transaction
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

Prover

\((x, w)\)

\[
\text{com}(01101\cdots1)
\]

Verifier

\((x)\)

\(i, j, k\)

Open \(\text{com}(ti), \text{com}(tj), \text{com}(tk)\)
Merkle Tree

Why (computationally) binding?

Can we make it hiding?
Is it ZK?

**Prover**: $(x, w)$

$MT(\text{Com}(01101\cdots1))$

**Verifier**: $(x)$

Open $\text{Com}(\Pi_i), \text{Com}(\Pi_j), \text{Com}(\Pi_k)$

$i, j, k$