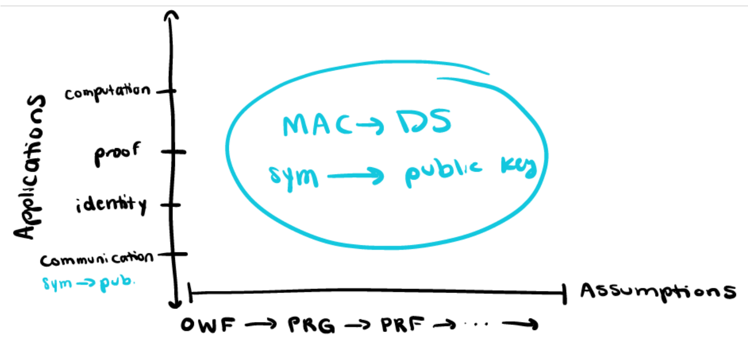
# Lesson 17 Lecture Notes

CS555

## 1 So far...




## 2 NP Proofs

### 2.1 Classical Proof

- Prover $P$ has a witness (e.g., a 3-coloring of graph $G$).
- Verifier $V$ checks:
  1. Only 3 colors are used.
  2. Any two connected vertices have different colors.
- Goal: Verify coloring property on $G$ without revealing the exact coloring.
- Computationally Zero-Knowledge Solution:
  - Prover commits on random permutation of graph with coloring.
  - Verifier challenges on one random edge.
  - Prover shows coloring of that edge.
- Properties:
  - **Completeness:** If $G \in$ 3-colorable, $V$ accepts $P$'s proof.
  - **Soundness:** If $G \notin$ 3-colorable, any cheating prover is rejected with probability $1 - \text{negl}$.
  - **Zero-Knowledge:** A cheating verifier cannot learn more than validity; simulator can reproduce verifier's view.

### 2.2 Proof it is Zero-Knowledge

- **Hybrid $H0$:**
  1. Pick random edge $(i, j)$.
  2. Color edge $(i, j)$ randomly; color all other edges red.
  3. Give commitments to $V$ and get edge challenge.
  4. If challenge $\neq (i, j)$, repeat.
  5. If challenge $= (i, j)$, output edge commitment and openings as simulated transcript.
- **Hybrid $H1$:**
  - Permute legal coloring and color all edges correctly.
  - $H0$ and $H1$ are indistinguishable.
- **Hybrid $H2$:**
  - No repeat step; always give colorings when queried.
  - $H1$ samples from same distribution as $H2$ with probability $1 - 1/|E|$.

### 2.3 Sequential vs. Parallel Repetition

- Zero-knowledge proof requires multiple queries to achieve negligible soundness error.
  - ie. is all edges but one is correct, it can pass with high probability on just a few runs but with many it will not
- Sequential repetition enforces randomness of verifier queries $\to$ could it be replaces with parallel repetition
- Parallel repetition does not preserve zero-knowledge property (cannot enforce truly random queries).

# 3 Proofs of Knowledge

- Beyond decision problems: show knowledge of solution (e.g., discrete log).
- Example: Discrete Logarithm Zero-Knowledge System
  - Parameters: $p = 2q + 1$, $y = g^x \pmod p$.
  - Protocol:
    1. Alice sends $z = g^r \pmod p$.
    2. Bob sends challenge $c \in \{0, 1\}$.
    3. Alice responds $s = r + cx \pmod q$.
    4. Bob verifies: $g^s \stackrel{?}{=} z \cdot y^c \to g^r \cdot g^{x^c} \to g^{r+xc} = g^s$
  - **Proof: Extractor**
    * If prover convinces verifier with prob $> 1/2 + \text{poly}^{-1}$, extractor rewinds to obtain two transcripts with different $c$.
    * From $s_0, s_1$, $g^{s_1 - s_0} = y$ so $s_1 - s_0$ is DLOG

# 4 Non-Interactive Zero-Knowledge Proofs

- Motivation: Interactive proofs require both parties online; want proofs usable later.
- Need a judge that makes usre both prover and verifier aren't cheating
- Fully non-interactive proofs only exist for $P$ problems; these are trivial because they are verifiable in polytime.
- Two approaches to NIZK:
  1. Random Oracle Model: both sides access unbiased random oracle.
  2. Common Random String Model: shared random string used for challenges and permutations.