# EECS 575: Advanced Cryptography
# Fall 2021
# Lecture 20

**Anthony Opipari**

# Agenda for this lecture

- Announcements
- Recap from last time
- Formalizing interactive proofs
  - For graph nonisomorphism
- Zero-knowledge proofs
  - Honest-verifier zero-knowledge (HVZK)
  - Full zero-knowledge
  - For graph isomorphism

# Agenda for this lecture

- Announcements
- Recap from last time
- Formalizing interactive proofs
  - For graph nonisomorphism
- Zero-knowledge proofs
  - Honest-verifier zero-knowledge
  - Full zero-knowledge
  - For graph isomorphism

# Announcements

- HW 6 is due next Monday, November 22
- Final exam will be released following Thanksgiving break

# Recap from last time

Zero Information: From probabilistic, if two R.Vs $X, Y$ are independent then $X$ reveals no information about $Y$!

Question: How to extend to the computational setting?

Answer: Zero knowledge

- If $Y$ can be simulated efficiently, up to computational indist, without simulator access to $X$, then $X$ "reveals zero knowledge" about $Y$

Idea: use ZK definition to model security when adversary is an active participant

Scheme will reveal ZK about $X$ if adversary's "view" can be simulated without access to $X$

# Recap from last time

Formalize what a proof is:

1) Sequence of logically/mechanically verifiable statements

2) A "dialogue" between interacting prover and verifier

Idea: Combine 1) and 2) to efficiently prove claims which we don't know how to prove with 1) alone.

↳ "Interactive Proof Systems"

# Agenda for this lecture

- Announcements
- Recap from last time
- Formalizing interactive proofs
  - For graph nonisomorphism
- Zero-knowledge proofs
  - Honest-verifier zero-knowledge (HVZK)
  - Full zero-knowledge
  - For graph isomorphism

# Formalizing interactive proofs

$IP = \langle P, V \rangle$: $P, V$ are "interactive" algorithms (communicate)

$P, V$ are randomized (can flip random coins)

"public coin" if $V$ publishes random coins

"private coin" if $V$ hides some coins from $P$

$P(\cdot) \longleftrightarrow V(\cdot)$ Denotes execution of interactive system

$\text{out}_V[\cdot]$ Denotes output of $V$ in the interaction

Language, $L \subseteq \{0,1\}^*$, represents the set of true statements

# Formalizing interactive proofs

**Definition:** An interactive proof system (IP) for language $L$ consists of (a possibly unbounded) alg. $P$ and a PPT alg. $V$

1) completeness: $\forall x \in L$, $\text{out}_V[P(x) \leftrightarrow V(x)] = 1$ w/ prb. 1

2) soundness: $\forall x \notin L$, and unbounded $P^*$,

$$\Pr[\text{out}_V[P^*(x) \leftrightarrow V(x)] = 1] \leq s < 1$$

where $s$ is the soundness error

**Observations:** Soundness ensures no "cheating" prover can convince the verifier of a false statement

By repeating $k$ independent trial, soundness error $s^k \approx 0$ if $k$ is large

# Interactive proof for graph nonisomorphism

Graph $G=(V,E)$ with vertices $V$, edges $E \subseteq V \times V$

Definition: Graphs $G$ and $H$ are isomorphic ($G=(V_G,E_G)$ $H=(V_H,E_H)$)
if they are the same graph up to vertex labels
if $\exists$ a bijection $\rho: V_G \to V_H$ s.t. $(u,v) \in E_G \iff (\rho(u),\rho(v)) \in E_H$

An IP system for GNI:

$P(G_0, G_1)$

$V(G_0, G_1)$

$H = \pi(G_b)$ ←

Choose $b \leftarrow \{0,1\}$

Choose a unif. Permutation $\pi$ of $V_{G_b}$

Challenge: "which graph is this"

Compute to determine if $G_0 \equiv H$ or $G_1 \equiv H$

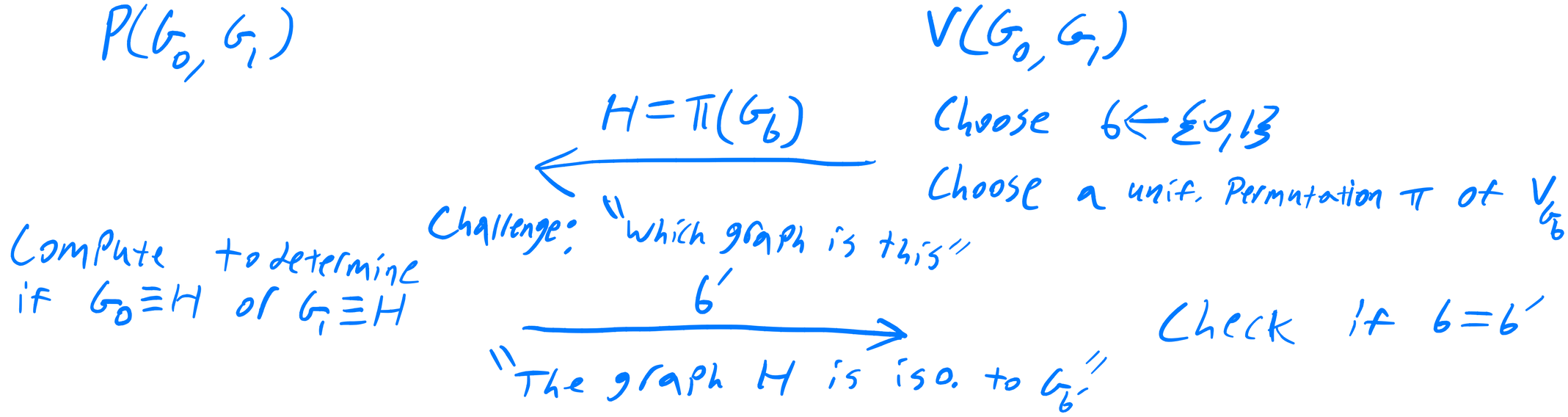
$b'$ →

Check if $b=b'$

"The graph $H$ is iso. to $G_b$"

# Interactive proof for graph nonisomorphism

An IP system for GNI:




1) Complete: If $(G_0, G_1) \in GNI$ $(G_0 \not\equiv G_1)$ then P can always determine $b'$ s.t. $b' = b$ and so V will always accept.

2) Sound: If $(G_0, G_1) \notin GNI$ $(G_0 \equiv G_1)$ then H is iso. to $G_0$ and $G_1$
   ↳ $P^*$ has $\frac{1}{2}$ chance to guess b, $s = \frac{1}{2}$

# Agenda for this lecture

- Announcements
- Recap from last time
- Formalizing interactive proofs
  - For graph nonisomorphism
- Zero-knowledge proofs
  - Honest-verifier zero-knowledge
  - Full zero-knowledge
  - For graph isomorphism

# Honest-verifier zero-knowledge

Recall: In previous GNI example, when $(G_0, G_1) \in GNI$ verifier received $b$, which it already knew: gains "Zero Knowledge"

Definition: An IP for language $L$ is honest-verifier zero-knowledge if $\exists$ PPT simulator $S$, s.t.

$\forall x \in L$, $View_V[P(x) \leftrightarrow V(x)] \approx^s S(x)$

where $View_V$ denotes verifier's entire "view" of the interaction

"whatever verifier would have seen talking to the prover could be generated using only the true statements"

Exercise: Show that the previous GNI protocol is HVZK

# Full zero-knowledge

Observation: Definition of HVZK ignores "malicious" verifier which attempts to extract information from the prover (e.g. sends $H$ w/ unknown relation to $G_0, G_1$ as challenge).

Definition: An IP for language $L$ is zero-knowledge if $\forall$ nuPPT (possibly malicious) $V^*$, $\exists$ nuPPT simulator $S$, s.t.

$\forall x \in L$, $View_{V^*}[P(x) \leftrightarrow V^*(x)] \approx^s S(x)$

Question: Does the previous GNI protocol satisfy zero-knowledge?

To answer, either define a simulator or show how $V^*$ can extract information from $P$.

# Zero-knowledge proof for graph isomorphism

Definition: Graph isomorphism defined as complement of GNI,
$GI = \{(G_0, G_1) : G_0 \equiv G_1\}$

Question: How to define a zero-knowledge interactive protocol for GI?

Idea: Disallow P from serving as an isomorphism oracle

P sends V a graph $H \equiv G_0 \equiv G_1$

V challenges P to demonstrate isomorphism for $H \equiv G_b$ where b is random

P show such an isomorphism

# Zero-knowledge proof for graph isomorphism

Question: How to define a zero-knowledge interactive protocol for GI?

Protocol: