Proving Computation Integrity

CS 276: Introduction to Cryptography

Sanjam Garg

April 6, 2026

Overview

1. Interactive Proofs
2. Zero-Knowledge Proofs

Interactive Proof Systems

Definition 1 (Interactive proof for language L)

An interactive protocol $(\mathcal{P}, \mathcal{V})$ is an interactive proof system for L if $\mathcal{V}$ is PPT and:

- **Completeness**: for all $x \in L$, $\Pr[\mathcal{V} \text{ accepts after interacting with } \mathcal{P}(x)] = 1$.
- **Soundness**: for all $x \notin L$ and all (possibly unbounded) $\mathcal{P}^*$, $\Pr[\mathcal{V} \text{ accepts after } \mathcal{P}^*(x)] \leq \mathsf{negl}(|x|)$.

Interactive Proof Systems

Definition 1 (Interactive proof for language L)

An interactive protocol $(\mathcal{P}, \mathcal{V})$ is an interactive proof system for L if $\mathcal{V}$ is PPT and:

- **Completeness**: for all $x \in L$, $\Pr[\mathcal{V} \text{ accepts after interacting with } \mathcal{P}(x)] = 1$.
- **Soundness**: for all $x \notin L$ and all (possibly unbounded) $\mathcal{P}^*$, $\Pr[\mathcal{V} \text{ accepts after } \mathcal{P}^*(x)] \leq \mathsf{negl}(|x|)$.

We allow probabilistic prover/verifier and interaction; this can prove statements where non-interactive succinct certificates are not known.

GI vs GNI: Why Interaction Matters

Graph Isomorphism (GI)

For $GI = \{(G_0, G_1) \mid G_0 \cong G_1\}$, a prover can send an isomorphism w directly; verifier checks in polynomial time.

GI vs GNI: Why Interaction Matters

Graph Isomorphism (GI)

For $GI = \{(G_0, G_1) \mid G_0 \cong G_1\}$, a prover can send an isomorphism w directly; verifier checks in polynomial time.

Graph Non-Isomorphism (GNI)

For $GNI = \{(G_0, G_1) \mid G_0 \ncong G_1\}$, no short witness is known. Interaction helps:

- Verifier sends a random relabeling H of one graph.
- If $G_0 \ncong G_1$, prover identifies which graph produced H.
- If $G_0 \cong G_1$, even unbounded prover can only guess with probability $1/2$.

Repeat k times to make soundness error 2^{-k}.

Interactive Proof for GNI

One round

1. $\mathcal{V}$ samples $b \leftarrow \{0,1\}$ and random permutation π, sets $H = \pi(G_b)$, sends H.
2. $\mathcal{P}$ outputs b' (which graph H came from).
3. $\mathcal{V}$ accepts iff $b' = b$.

Interactive Proof for GNI

One round

1. $\mathcal{V}$ samples $b \leftarrow \{0,1\}$ and random permutation π, sets $H = \pi(G_b)$, sends H.
2. $\mathcal{P}$ outputs b' (which graph H came from).
3. $\mathcal{V}$ accepts iff $b' = b$.

Analysis

- If $(G_0, G_1) \in GNI$, unbounded $\mathcal{P}$ always answers correctly.
- If $(G_0, G_1) \notin GNI$, distributions are identical; any prover succeeds with probability at most $1/2$.

Sequential repetition gives negligible soundness error.

From Interactive Proofs to Zero Knowledge

Goal

Verifier should learn only statement validity, not the witness.

HVZK: Honest Verifier's View

Experiment (honest $\mathcal{V}$)

1. Common input $x \in L$; prover holds witness $w \in R(x)$.
2. Honest $\mathcal{V}(x)$ with random coins r interacts with $\mathcal{P}(x, w)$.
3. **View**: $\mathcal{V}$'s coins r together with all messages it sees (jointly distributed).

Honest-Verifier Zero Knowledge

Definition 2 (HVZK)

$(\mathcal{P}, \mathcal{V})$ is (computational) **HVZK** for language L w.r.t. witness relation R if there exists PPT simulator $\mathcal{S}$ such that for all $x \in L$, $w \in R(x)$:

$$\{View_{\mathcal{V}}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x))\} \approx_c \{\mathcal{S}(x)\}.$$

Honest-Verifier Zero Knowledge

Definition 2 (HVZK)

$(\mathcal{P}, \mathcal{V})$ is (computational) **HVZK** for language L w.r.t. witness relation R if there exists PPT simulator $\mathcal{S}$ such that for all $x \in L$, $w \in R(x)$:

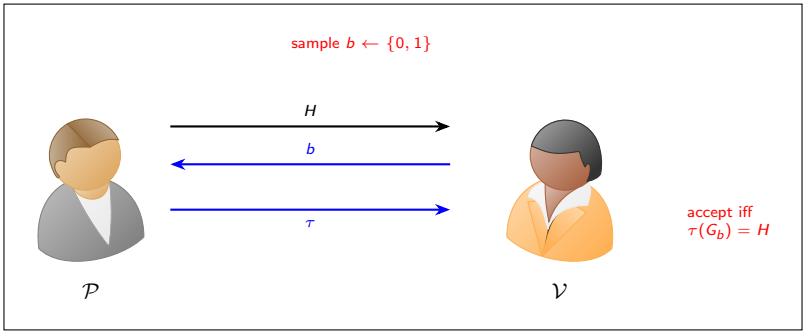
$$\{View_{\mathcal{V}}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x))\} \approx_c \{\mathcal{S}(x)\}.$$

Quantifiers

The simulator must be a *single* machine for all witnesses: we cannot swap to "$\forall x, w, \exists \mathcal{S}$" and let $\mathcal{S}$ hardcode w—that would trivialize the definition.

HVZK: Prover vs Honest Verifier (GI example)

$$View_{\mathcal{V}}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}(x))$$



GI Protocol: HVZK and ZK

GI protocol (single round)

Given $x = (G_0, G_1)$ and witness isomorphism w:

1. Prover picks random permutation σ, sends $H = \sigma(G_1)$.
2. Verifier sends random bit b.
3. Prover returns $\tau = \sigma$ if $b = 1$, else $\tau = \sigma \circ w$.
4. Verifier checks $\tau(G_b) = H$.

GI Protocol: HVZK and ZK

GI protocol (single round)

Given $x = (G_0, G_1)$ and witness isomorphism w:

1. Prover picks random permutation σ, sends $H = \sigma(G_1)$.
2. Verifier sends random bit b.
3. Prover returns $\tau = \sigma$ if $b = 1$, else $\tau = \sigma \circ w$.
4. Verifier checks $\tau(G_b) = H$.

HVZK simulator $\mathcal{S}(G_0, G_1)$

Sample $b \leftarrow \{0, 1\}$, random σ, set $H = \sigma(G_b)$; output (H, b, σ). Same distribution as the real transcript for honest $\mathcal{V}$.

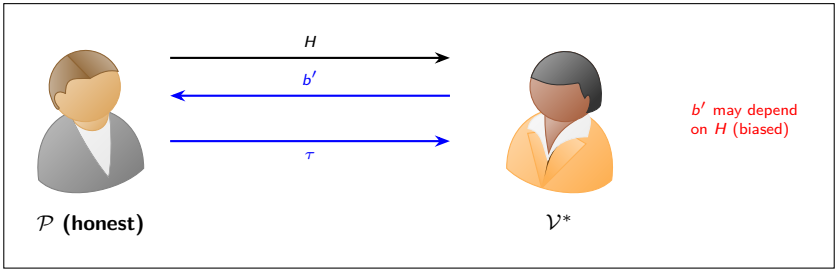
Malicious Verifier: Experiment

Real interaction with $\mathcal{V}^*$

1. Common input x; prover runs $\mathcal{P}(x, w)$; verifier runs arbitrary PPT $\mathcal{V}^*(x)$ (may deviate from protocol).
2. Messages are exchanged according to $\mathcal{V}^*$'s code and $\mathcal{P}$'s honest replies.
3. $\mathit{View}_{\mathcal{V}^*}$ = coins of $\mathcal{V}^*$ plus all messages it sees.

ZK: Prover vs Malicious Verifier

$$View_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x))$$



Zero Knowledge (Malicious Verifier)

Definition 3 (ZK)

$(\mathcal{P}, \mathcal{V})$ is **ZK** for L w.r.t. R if for every PPT $\mathcal{V}^*$ there exists PPT $\mathcal{S}$ such that for all $x \in L$, $w \in R(x)$:

$$\{View_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x))\} \approx_c \{\mathcal{S}(x)\}.$$

(Quantifiers: simulator may depend on $\mathcal{V}^*$; same $\mathcal{S}$ for all verifiers is stronger.)

Zero Knowledge (Malicious Verifier)

Definition 3 (ZK)

$(\mathcal{P}, \mathcal{V})$ is **ZK** for L w.r.t. R if for every PPT $\mathcal{V}^*$ there exists PPT $\mathcal{S}$ such that for all $x \in L$, $w \in R(x)$:

$$\{View_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x))\} \approx_c \{\mathcal{S}(x)\}.$$

(Quantifiers: simulator may depend on $\mathcal{V}^*$; same $\mathcal{S}$ for all verifiers is stronger.)

Definition 4 (ZK security)

$\mathcal{V}^*$ cannot distinguish real interaction from $\mathcal{S}(x)$ except with advantage $\leq \text{negl}(|x|)$.

GI: Simulator for Malicious $\mathcal{V}^*$ (Rewinding)

Simulator $\mathcal{S}$ with black-box access to $\mathcal{V}^*$

1. For $i = 1, \ldots, T$ with $T = \text{poly}(|x|)$:
   1. Sample $b \leftarrow \{0,1\}$ and uniform permutation σ.
   2. Send $H = \sigma(G_b)$ to $\mathcal{V}^*$; receive reply b'.
   3. If $b' = b$, output transcript (H, b, σ) and **terminate**.
2. If no round succeeds, output $\perp$ (failure probability $\leq 2^{-T} = \text{negl}(|x|)$ for appropriate T).

GI ZK: Hybrid H_0 (real interaction)

H_0

Run the honest prover on $x = (G_0, G_1)$ with witness w against $\mathcal{V}^*(x)$ once. Output

$$\mathit{View}_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x)).$$

(This is the real distribution in the ZK definition.)

GI ZK: Hybrid H_1

Procedure for H_1

1. For $i = 1, \ldots, T$:
   1. Sample $b^* \leftarrow \{0, 1\}$ uniformly.
   2. Run a fresh copy of H_0: execute $\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x)$.
   3. If $b^* = 0$, output that round's $\mathit{View}_{\mathcal{V}^*}(\mathcal{P}(x, w) \leftrightarrow \mathcal{V}^*(x))$ and **stop**.
   4. If $b^* = 1$, continue the outer loop.
2. If no round outputs, output $\perp$.

GI ZK: Hybrid H_1

Procedure for H_1

1. For $i = 1, \ldots, T$:
   1. Sample $b^* \leftarrow \{0,1\}$ uniformly.
   2. Run a fresh copy of H_0: execute $\mathcal{P}(x,w) \leftrightarrow \mathcal{V}^*(x)$.
   3. If $b^* = 0$, output that round's $\mathit{View}_{\mathcal{V}^*}(\mathcal{P}(x,w) \leftrightarrow \mathcal{V}^*(x))$ and **stop**.
   4. If $b^* = 1$, continue the outer loop.
2. If no round outputs, output $\perp$.

Analysis (H_0 vs. H_1)

Each iteration runs the same experiment as H_0; b^* is independent of that interaction. The output of H_1 conditional on not outputting $\perp$ is therefore distributed exactly as in H_0. We have $\Pr[H_1 = \perp] \leq 2^{-T}$ (needed $b^* = 1$ for all T rounds). For $T = \mathrm{poly}(|x|)$ this is $\mathrm{negl}(|x|)$, so hybrids are statistically (hence computationally) indistinguishable from H_0.

GI ZK: Hybrid H_2

Procedure for H_2

Same as H_1, except in each iteration:

1. Sample $b^* \leftarrow \{0,1\}$.
2. Run $\mathcal{P}(x,w) \leftrightarrow \mathcal{V}^*(x)$; let b be the challenge bit b appearing in that transcript.
3. If $b^* = b$, output this round's view and **stop**; else continue.

If no round succeeds, output $\perp$.

GI ZK: Hybrid H_2

Procedure for H_2

Same as H_1, except in each iteration:

1. Sample $b^* \leftarrow \{0,1\}$.
2. Run $\mathcal{P}(x,w) \leftrightarrow \mathcal{V}^*(x)$; let b be the challenge bit b appearing in that transcript.
3. If $b^* = b$, output this round's view and **stop**; else continue.

If no round succeeds, output $\perp$.

Analysis (H_1 vs. H_2)

In each iteration the transcript is generated in the same way as in H_1; b^* is uniform and independent of the interaction until both are fixed. Replacing "output if $b^* = 0$" by "output if $b^* = b$" does not change the joint distribution of $(\text{transcript}, b^*)$ because, given transcript, b is fixed and b^* is still uniform—so $\Pr[b^* = 0] = \Pr[b^* = b] = \frac{1}{2}$. Hence $H_2 \equiv H_1$.

GI ZK: Hybrid H_3 (simulator)

H_3

Run $\mathcal{S}^{\mathcal{V}^*}(x)$: in each attempt sample b, σ, send $H = \sigma(G_b)$ to $\mathcal{V}^*$, receive b'; if $b' = b$ output the partial transcript (H, b, τ) with $\tau = \sigma$ (same rule as honest $\mathcal{P}$ whenever b matches the interaction); else **rewind** and retry. Output $\perp$ if all T attempts fail.

Analysis (H_2 vs. H_3)

- If $\mathcal{V}^*$ replies $b' = b$, the view equals the corresponding prefix of an honest interaction; if $b' \neq b$, $\mathcal{S}$ discards—same selective sampling as in H_2 with acceptance probability $\frac{1}{2}$ per try.