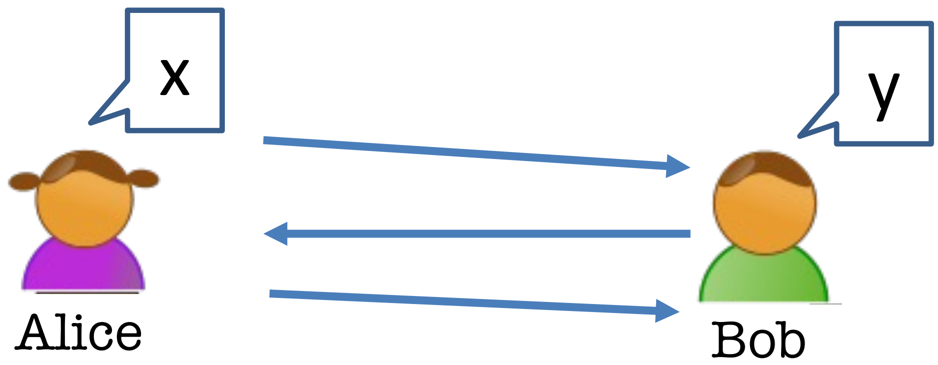
MIT 6.875

Foundations of Cryptography
Lecture 14

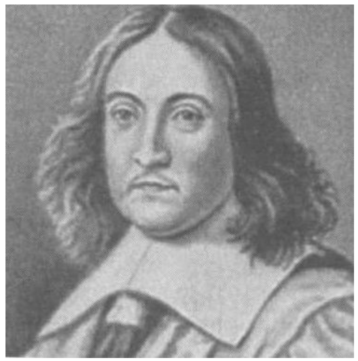
Beyond Secure Communication



Much more than communicating securely.

- Complex Interactions: proofs, computations, games.
- Complex Adversaries: Alice or Bob, adaptively chosen.
- Complex Properties: Correctness, Privacy, Fairness.
- Many Parties: this class, MIT, the internet.

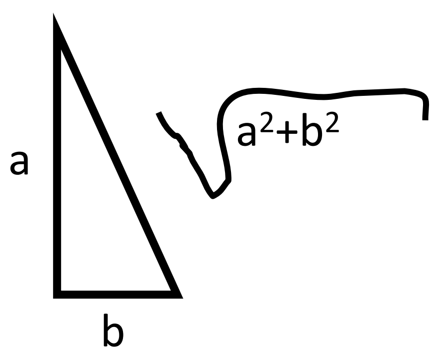
Classical Proofs

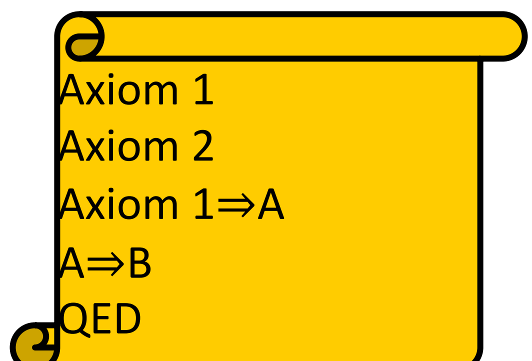


Steve Cook

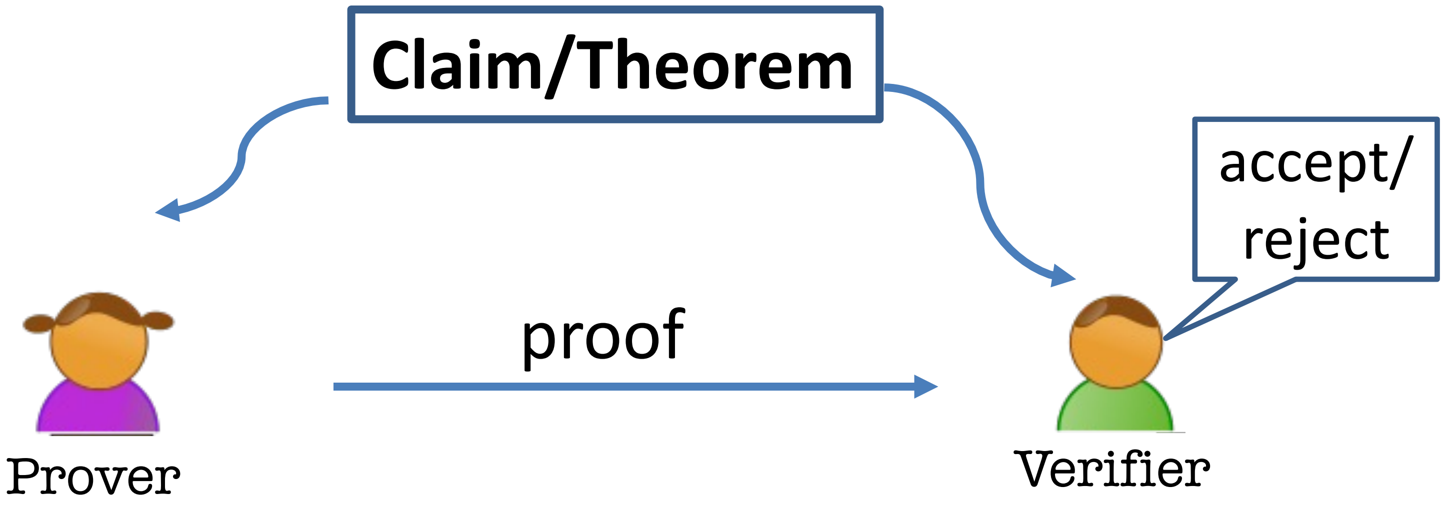
Leonid Levin

Prover writes down a string (proof); Verifier checks.

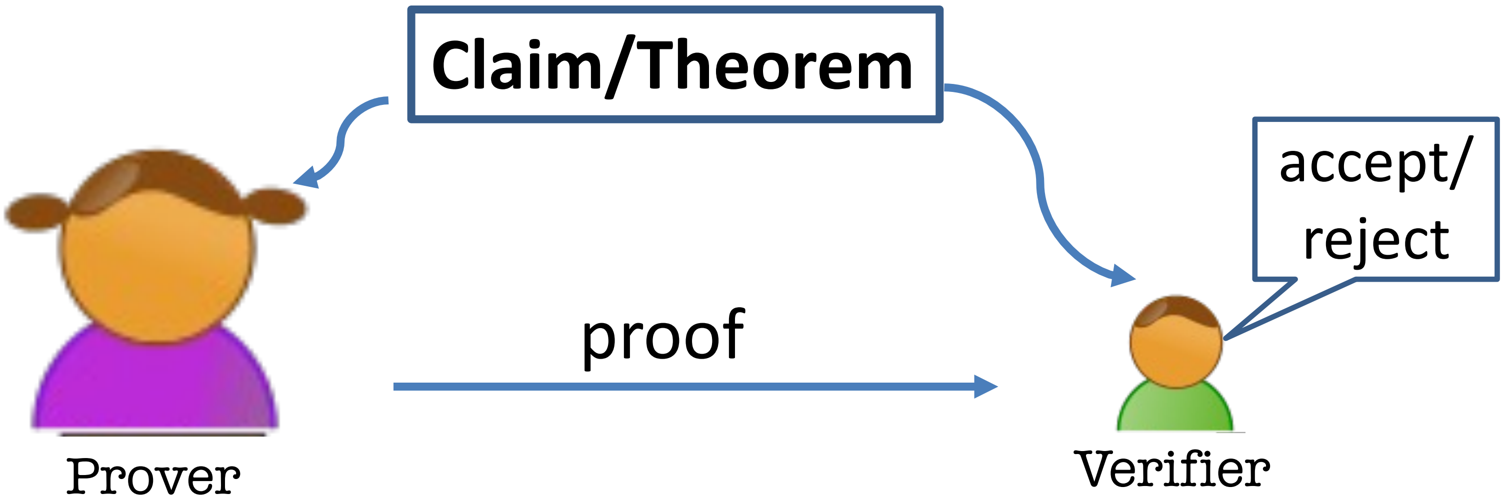




Proofs



Efficiently Verifiable Proofs: $\mathcal{NP}$



Works hard

Polynomial-time

Theorem: N is a product of two prime numbers

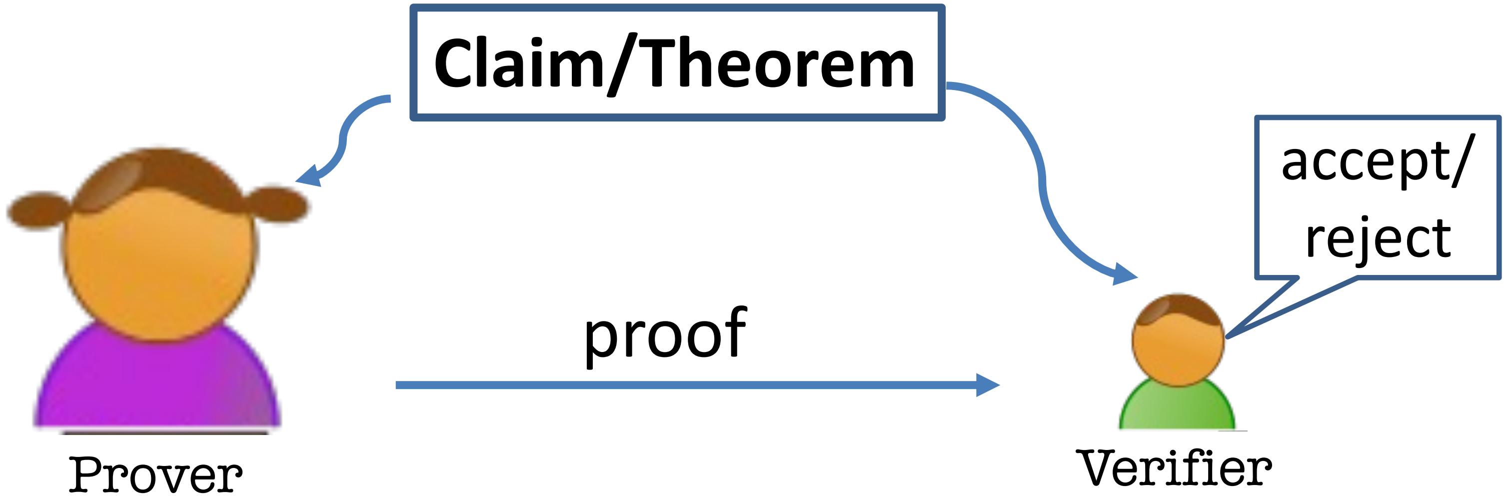
Prover

Proof $= (\boldsymbol{P}, \boldsymbol{Q})$

Verifier

Accept *iff* $\mathrm{N} = PQ\ and\ P, Q\ prime$

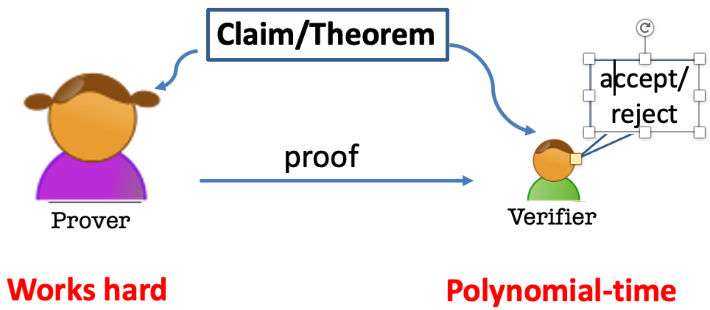
Efficiently Verifiable Proofs: $\mathcal{NP}$



Works hard **Polynomial-time**

<u>Def</u>: A language/decision procedure $\mathcal{L}$ is simply a set of strings. So, $\mathcal{L} \subseteq \{0,1\}^*$.

Efficiently Verifiable Proofs: $\mathcal{NP}$



Def: $\mathcal{L}$ is an $\mathcal{NP}$-language if there is a **poly-time** verifier V where

- **Completeness: True theorems have (short) proofs.**

  for all x $\in \mathcal{L}$, there is a **poly(|x|)-long** witness (proof) w $\in \{0,1\}^*$ s.t. $V(x, w) = 1$.

- **Soundness: False theorems have no short proofs.**

  for all x $\notin \mathcal{L}$, there is no witness. That is, for all polynomially long w $\in \{0,1\}^*$, $V(x, w) = 0$.

Theorem: N is a product of two prime numbers

Prover

Proof $= (\boldsymbol{P}, \boldsymbol{Q})$

Verifier

Accept *iff* $\mathrm{N} = PQ$.

After interaction, Bob the Verifier knows:

1) N is a product of two primes.

2) **Also**, the two factors of N.

Theorem: y is a quadratic residue mod N

Prover

Proof $= \boldsymbol{x} \in \boldsymbol{Z}_N^*$

Verifier

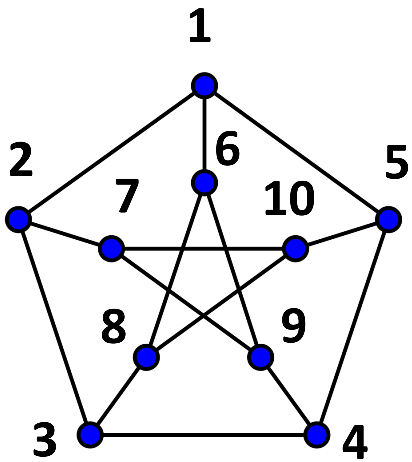
Accept *iff*
$y = x^2 \pmod{N}$.

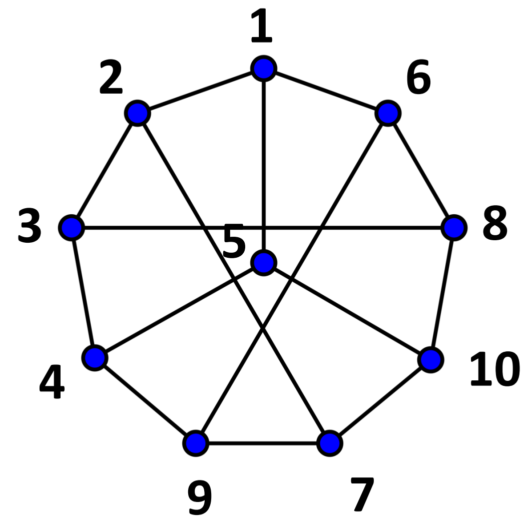
After interaction, Bob the Verifier knows:

1) y is a quadratic residue mod N.

2) **Also**, the square root of y.

Theorem: Graphs G_0 and G_1 are isomorphic.





Prover

Proof $= \pi$: $[N] \rightarrow [N]$,

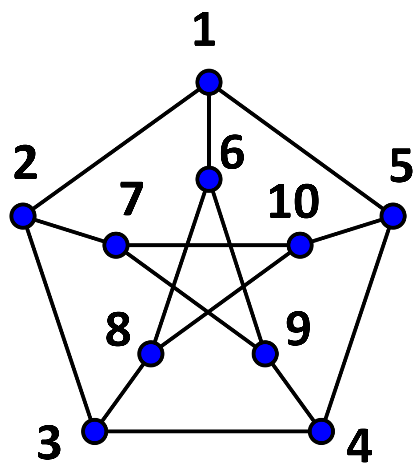
the isomorphism

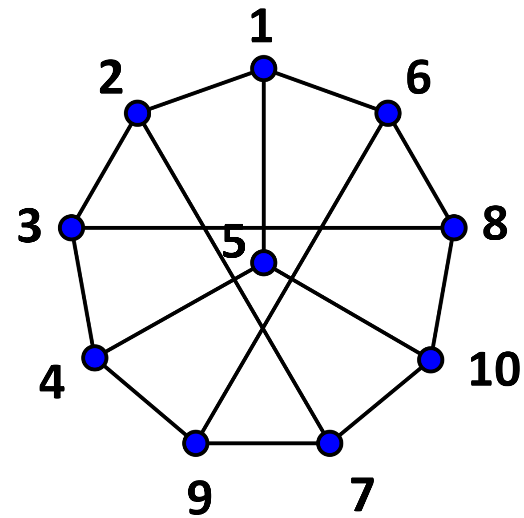
Verifier

Check $\forall i, j$:

$\left(\pi(i), \pi(j)\right) \in E_1$ iff $(i, j) \in E_0$.

Theorem: Graphs G_0 and G_1 are isomorphic.





Prover

Proof $= \pi: [N] \rightarrow [N]$,

the isomorphism

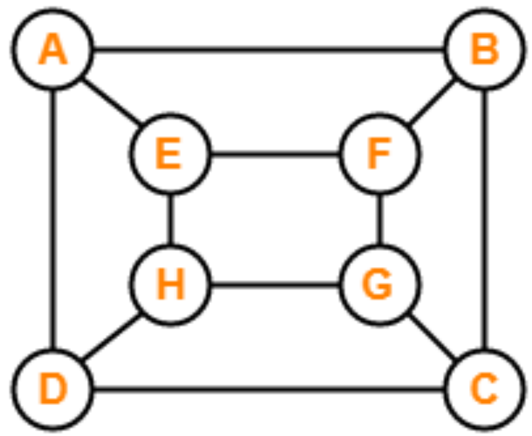
Verifier

After interaction, Bob the Verifier knows:

1) G_0 and G_1 are isomorphic.

2) **Also**, the isomorphism.

Theorem: Graphs G has a Hamiltonian cycle.



Prover

Proof = Hamiltonian cycle

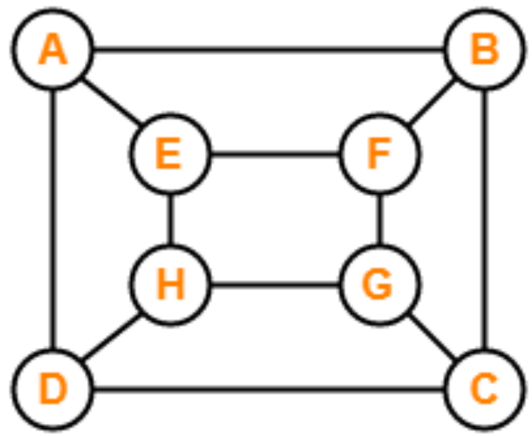
$$(\boldsymbol{v_0}, \ldots, \boldsymbol{v_{N-1}})$$

Verifier

Check $\forall i$:

$$(v_i, v_{i+1 \ mod \ N}) \in E$$

Theorem: Graphs G has a Hamiltonian cycle.



Proof = Hamiltonian cycle

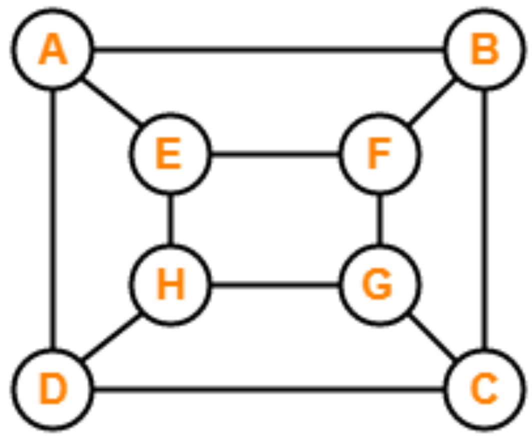
$$(v_0, \ldots, v_{N-1})$$

After interaction, Bob the Verifier knows:

1) G ***has*** a Hamiltonian cycle.

2) **Also**, the Hamiltonian cycle itself.

Theorem: Graphs G has a Hamiltonian cycle.



Prover

Proof = Hamiltonian cycle

$$(v_0, \ldots, v_{N-1})$$

Verifier

NP-Complete Problem:

Every one of the other problems can be reduced to it

Theorem: y is a quadratic residue mod N

Prover

Proof $= x \in Z_N^*$

Verifier

Accept *iff*
$y = x^2 \pmod N$.

After interaction, Bob the Verifier knows:

1) y is a quadratic residue mod N.

2) **Also**, the square root of y.

Is there any other way?

Zero Knowledge Proofs

“I will prove to you that I could’ve sent you a proof if I felt like it.”

Prover

Micali Goldwasser Rackoff

Zero Knowledge Proofs

Prover

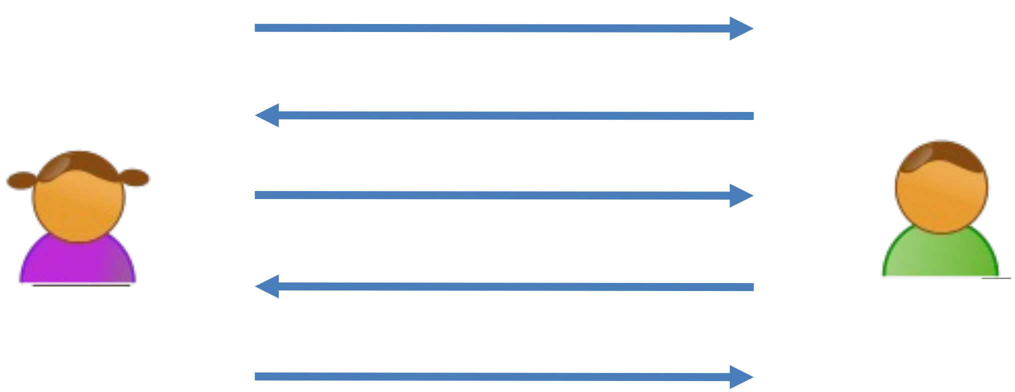
"I will not give you the square root, but I will prove to you that I could provide one if I wanted to."

Micali Goldwasser Rackoff

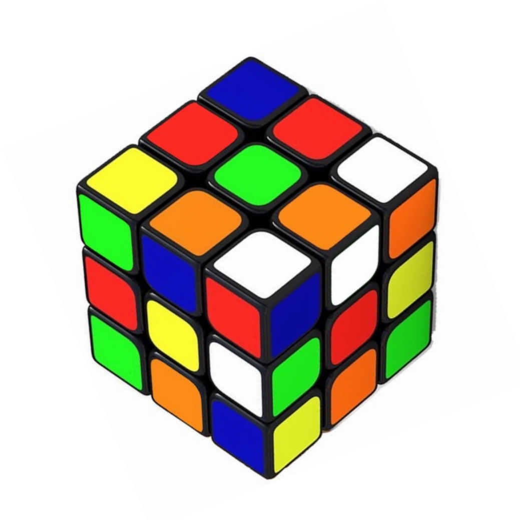
Two (Necessary) New Ingredients

1. Interaction: Rather than passively reading the proof, the verifier engages in a conversation with the prover.

2. Randomness: The verifier is randomized and can make a mistake with a (exponentially small) probability.



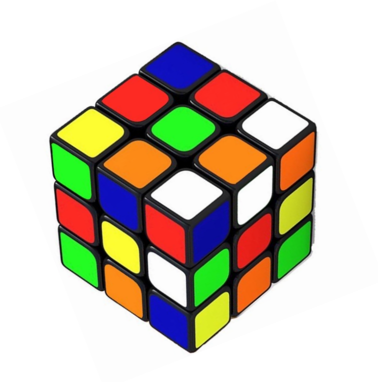
Here is the idea.



Prover

THEOREM: “there is an ≤ k move solution to this cube”

Here is the idea.



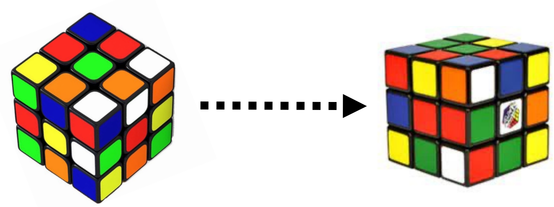
"Random" config

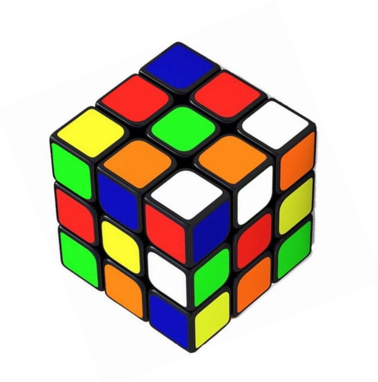
Challenge (0 or 1)

Prover

Verifier

0: Show $k/2$ moves





Here is the idea.

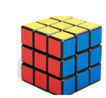
"Random" config

Challenge (0 or 1)

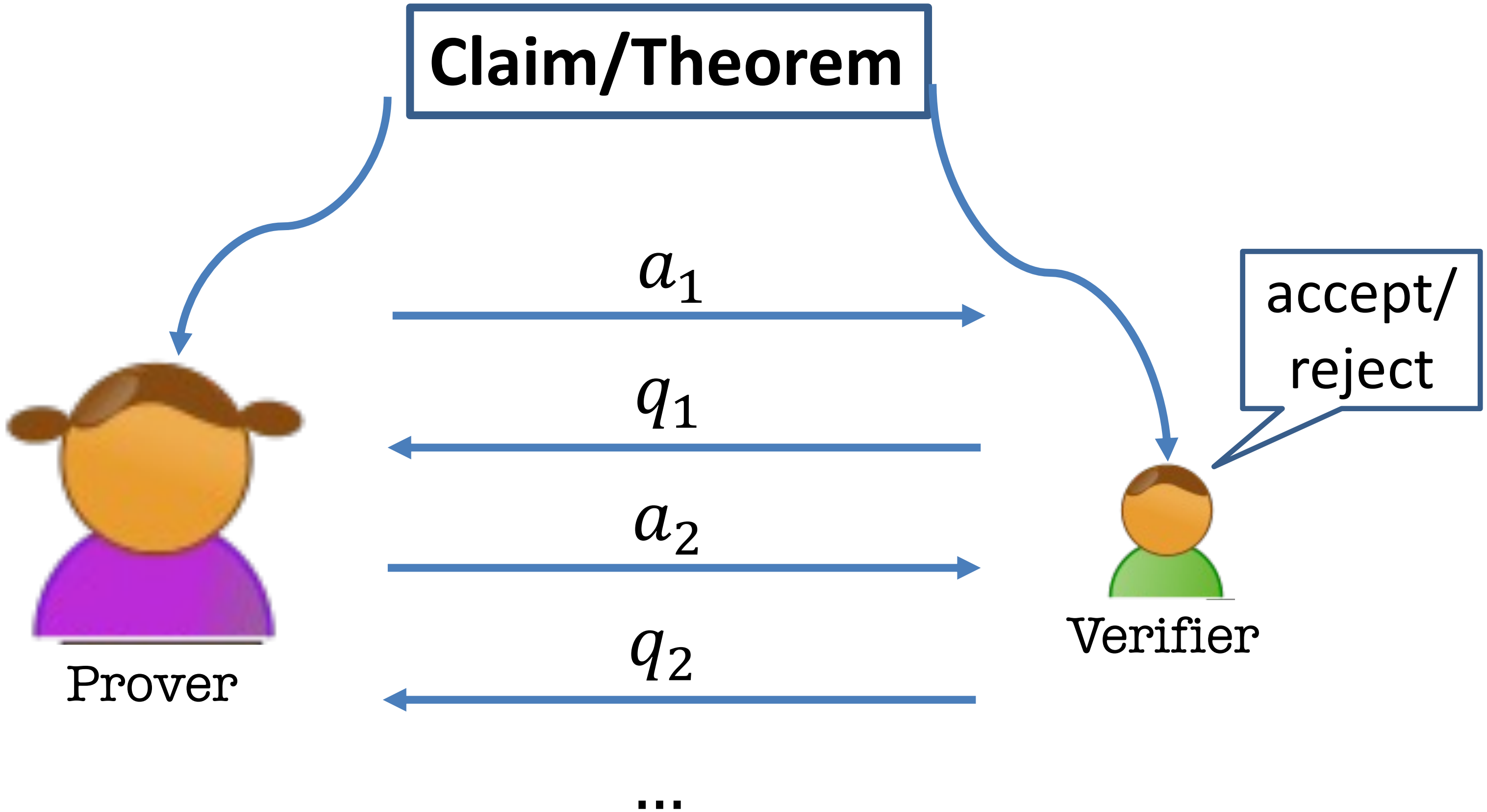
Prover

1: Show $k/2$ moves

Verifier

POINT IS THIS: If the prover can do both consistently, then there exist k moves that map  to 

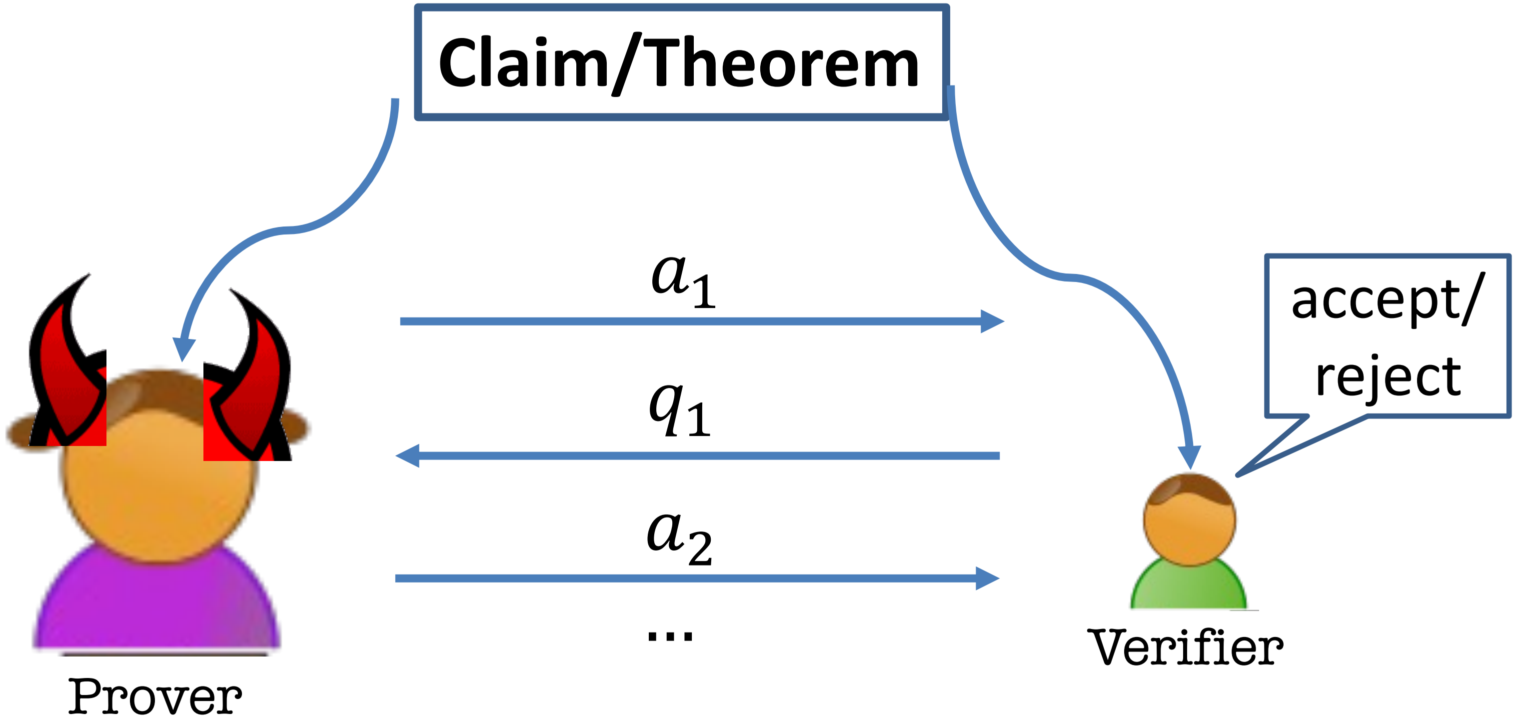
Interactive Proofs for a Language $\mathcal{L}$



Comp. Unbounded

Probabilistic
Polynomial-time

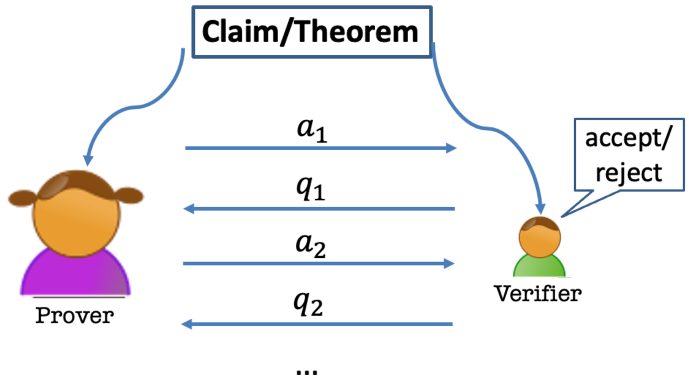
Interactive Proofs for a Language $\mathcal{L}$



Def: $\mathcal{L}$ is an $\mathcal{IP}$-language if there is a unbounded P and **probabilistic poly-time** verifier V where

- **Completeness**: If $x \in \mathcal{L}$, V always accepts.
- **Soundness:** If $x \notin \mathcal{L}$, regardless of the cheating prover strategy, V accepts with negligible probability.

Interactive Proofs for a Language $\mathcal{L}$



Def: $\mathcal{L}$ is an $\mathcal{IP}$-language if there is a **probabilistic poly-time** verifier V where

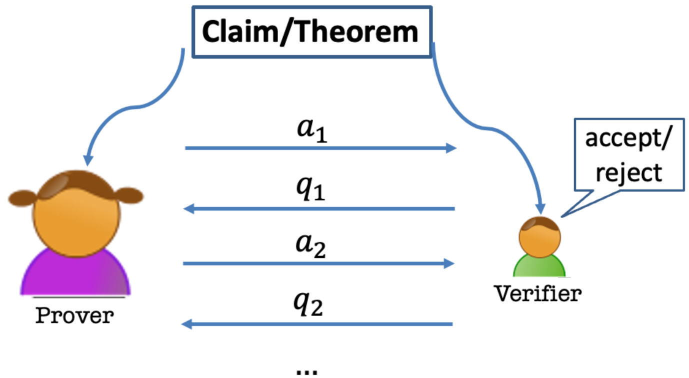
- **Completeness**: **If** $\mathrm{x} \in \mathcal{L}$**,**

$$\Pr[(\mathrm{P}, V)(x) = accept] = 1.$$

- **Soundness: If** $\mathrm{x} \notin \mathcal{L}$, **there is a negligible function** negl **s.t. for every** $\boldsymbol{P}^*$,

$$\Pr[(P^*, V)(x) = accept] = \mathrm{negl}(\lambda).$$

Interactive Proofs for a Language $\mathcal{L}$



<u>Def</u>: $\mathcal{L}$ is an $\mathcal{IP}$-language if there is a **probabilistic poly-time** verifier V where

- **Completeness**: **If** $\mathrm{x} \in \mathcal{L}$**,**

$$\Pr[(\mathrm{P}, V)(x) = accept] \geq \boldsymbol{c}.$$

- **Soundness: If** $\mathrm{x} \notin \mathcal{L}$, **there is a negligible function** negl **s.t. for every** $\boldsymbol{P}^*$**,**

$$\Pr[(P^*, V)(x) = accept] \leq \mathbf{s}.$$

Equivalent as long as $\boldsymbol{c} - \mathbf{s} \geq 1/\mathrm{poly}(\lambda)$

Interactive Proof for QR

$\mathcal{L} = \{(N, y) : y \text{ is a quadratic residue mod } N\}.$

(N, y)

$s = r^2 \pmod N$

$\xrightarrow{\hspace{3em}}$

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

$\xleftarrow{\hspace{3em}}$

If b=0: $z = r$

$\xrightarrow{\hspace{3em}}$

If b=1: $z = rx$

(N, y)

Check:

$z^2 = sy^b \pmod N$

Completeness

Claim: If $(N, y) \in L$, then the verifier accepts the proof with probability 1.

Proof:

$$z^2 = (rx^b)^2 = r^2(x^2)^b = sy^b \pmod{N}$$

So, the verifier's check passes and he accepts.

Soundness

Claim: If $(N, y) \notin L$, then for every cheating prover P^*, the verifier accepts with probability at most 1/2.

Proof: Suppose the verifier accepts with probability > 1/2.

Then, there is some $s \in Z_N^*$ s.t. the prover produces

$$z_0 : z_0^2 = s \pmod N$$

$$z_1 : z_1^2 = sy \pmod N$$

This means $(z_1/z_0)^2 = y \pmod N$, which tells us that $(N, y) \in L$.

Interactive Proof for QR

$\mathcal{L} = \{(N, y): y \text{ is a quadratic residue mod } N\}.$

$s_i = r_i^2 \pmod N$

(N, y)

$b_i \leftarrow \{0,1\}$

(N, y)

If b_i=0: $z_i = r_i$

If b_i=1:$z_i = xr_i$

Check for all i:

$z_i^2 = s_i y^b \pmod N$

REPEAT sequentially λ times.

Soundness

Claim: If $(N, y) \notin L$, then for every cheating prover P^*, the verifier accepts with probability at most $(\frac{1}{2})^{\lambda}$.

Proof: Exercise.

This is Zero-Knowledge.

But what does that mean?

$$s = r^2 \pmod N$$

(N, y)

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

(N, y)

If b=0: $z = r$

If b=1: $z = rx$

Check:
$z^2 = sy^b \pmod N$

How to Define Zero-Knowledge?

After the interaction, V knows:

- The theorem is true; and
- A **view** of the interaction
  (= transcript + coins of V)

P gives zero knowledge to V:

When the theorem is true, the view gives V nothing that he couldn't have obtained on his own without interacting with P.

How to Define Zero-Knowledge?

(P, V) is zero-knowledge if V can generate his **VIEW** of the interaction **all by himself** in **probabilistic polynomial time**.

How to Define Zero-Knowledge?

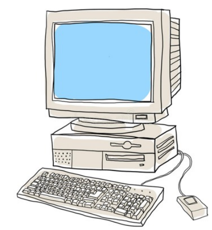
(P, V) is zero-knowledge if V can "simulate" his **VIEW** of the interaction **all by himself** in **probabilistic polynomial time**.

The Simulation Paradigm

sim_S:
(s, b, z)

$\boldsymbol{view_V(P,V)}$:
Transcript $= (s, b, z)$,
Coins = b

PPT "simulator" S



(N, y)

$s = r^2 \pmod{N}$ $\rightarrow$

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

(N, y)

If b=0: $z = r$ $\rightarrow$
If b=1: $z = rx$

Check:
$z^2 = sy^b \pmod{N}$

Zero Knowledge: Definition

An Interactive Protocol (P,V) is zero-knowledge for a language L if there exists a **PPT** algorithm S (a simulator) such that **for every $\boldsymbol{x \in L}$**, the following two distributions are indistinguishable:

1. $view_V(P, V)$
2. $S(x, 1^\lambda)$

(P,V) is a zero-knowledge interactive protocol if it is complete, sound and zero-knowledge.

Perfect Zero Knowledge: Definition

An Interactive Protocol (P,V) is **perfect zero-knowledge** for a language L if there exists a PPT algorithm S (a simulator) such that for every $x \in L$, the following two distributions are **identical**:

1. $view_V(P, V)$
2. $S(x, 1^\lambda)$

(P,V) is a zero-knowledge interactive protocol if it is complete, sound and zero-knowledge.

Statistical Zero Knowledge: Definition

An Interactive Protocol (P,V) is **statistical zero-knowledge** for a language L if there exists a PPT algorithm S (a simulator) such that for every $x \in L$, the following two distributions are **statistically indistinguishable**:

1. $view_V(P, V)$

2. $S(x, 1^\lambda)$

(P,V) is a zero-knowledge interactive protocol if it is complete, sound and zero-knowledge.

Computational Zero Knowledge: Definition

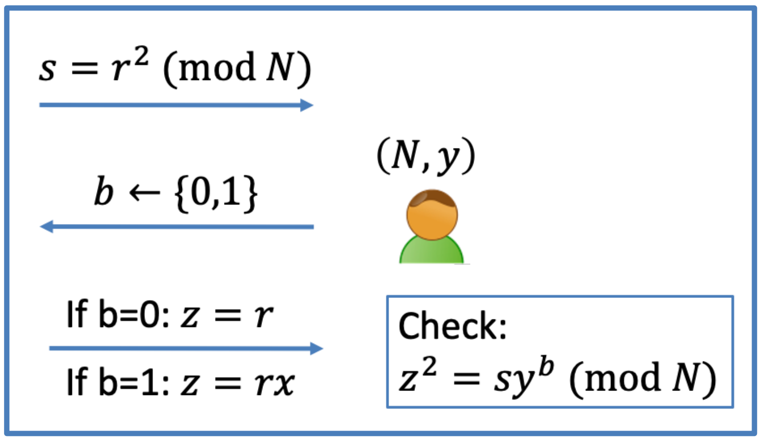
An Interactive Protocol (P,V) is **computational zero-knowledge** for a language L if there exists a PPT algorithm S (a simulator) such that for every $x \in L$, the following two distributions are **computationally indistinguishable**:

1. $view_V(P, V)$
2. $S(x, 1^\lambda)$

(P,V) is a zero-knowledge interactive protocol if it is complete, sound and zero-knowledge.

Zero Knowledge

Claim: The QR protocol is zero knowledge.



$view_V(P, V)$:

(s, b, z)

Simulator S works as follows:

1. First pick a random bit b.
2. pick a random $z \in Z_N^*$.
3. compute $\mathrm{s} = z^2/y^b$.
4. output $(\mathrm{s}, \mathrm{b}, \mathrm{z})$.

Exercise: The simulated transcript is identically distributed as the real transcript in the interaction (P,V).

What if V is NOT HONEST.

OLD DEF

An Interactive Protocol (P,V) is **honest-verifier** perfect zero-knowledge for a language L if there exists a PPT simulator S such that for every $x \in L$, the following two distributions are identical:

1. $view_V(P, V)$ 2. $S(x, 1^\lambda)$

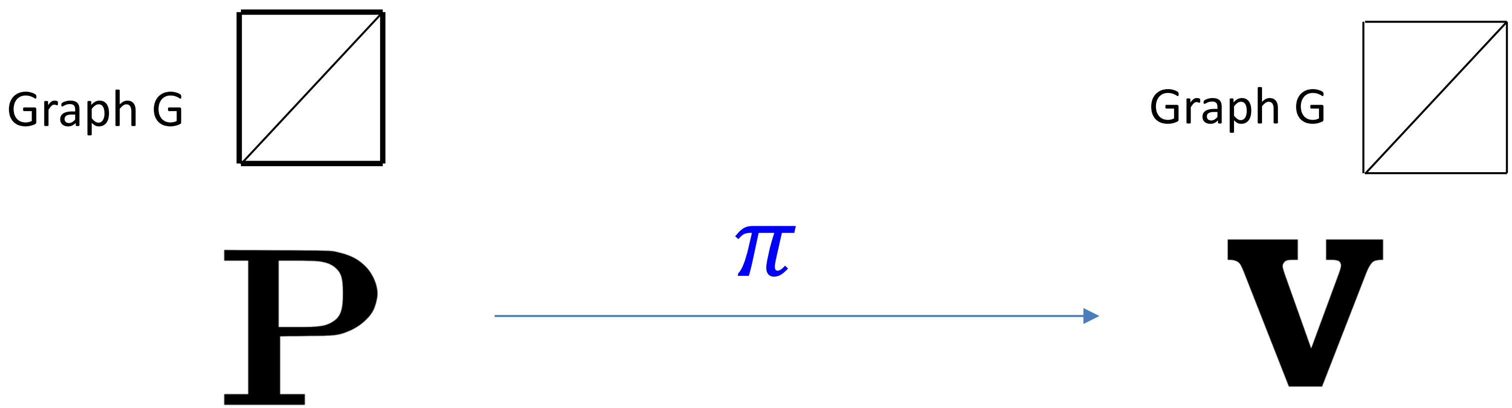
REAL DEF

An Interactive Protocol (P,V) is **perfect zero-knowledge** for a language L if **for every PPT V^***, there exists a (expected) poly time simulator S s.t. for every $x \in L$, the following two distributions are identical:

1. $view_{V^*}(P, V^*)$ 2. $S(x, 1^\lambda)$

Interaction is Necessary for ZK

Suppose there *were* a non-interactive ZK proof system for 3COL.

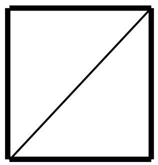


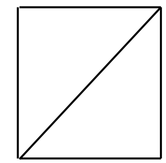
Step 1. When G is in 3COL, V accepts the proof π.

(Completeness)

Interaction is Necessary for ZK

Suppose there *were* a non-interactive ZK proof system for 3COL.

Graph G 

Graph G 

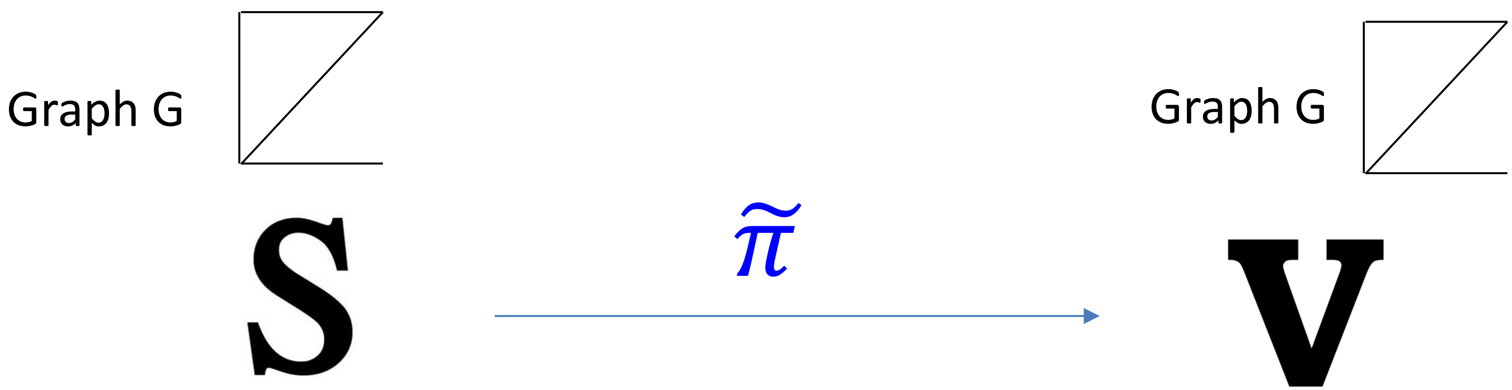
S $\xrightarrow{\tilde{\pi}}$ **V**

Step 2. **PPT** Simulator S, **given only G in 3COL**, produces an indistinguishable proof $\tilde{\pi}$ (Zero Knowledge).

In particular, V accepts $\tilde{\pi}$.

Interaction is Necessary for ZK

Suppose there *were* a non-interactive ZK proof system for 3COL.

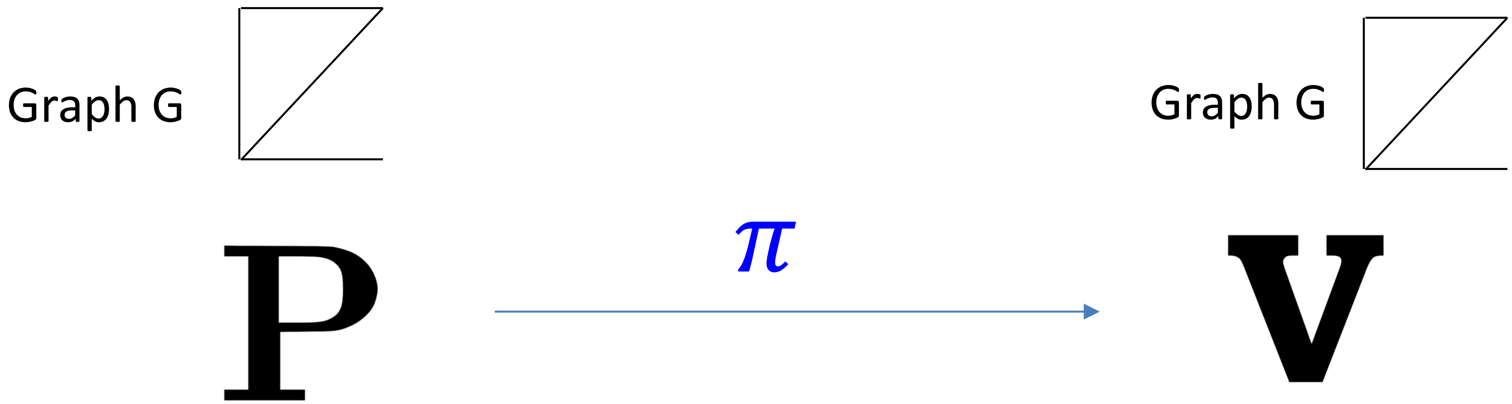


Step 3. Imagine running the Simulator S on a $G \notin$ 3COL. It produces a proof $\tilde{\pi}$ which the verifier still accepts!

(WHY?! Because S and V are PPT. They together cannot tell if the input graph is 3COL or not)

Interaction is Necessary for ZK

Suppose there *were* a non-interactive ZK proof system for 3COL.



Step 4. **Therefore, S is a cheating prover!**

Produces a proof for a $G \notin$ 3COL that the verifier nevertheless accepts.

Ergo, the proof system is NOT SOUND! $\blacksquare$