CIS 5560

Cryptography
Lecture 22

Course website:
pratyushmishra.com/classes/cis-5560-s24/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

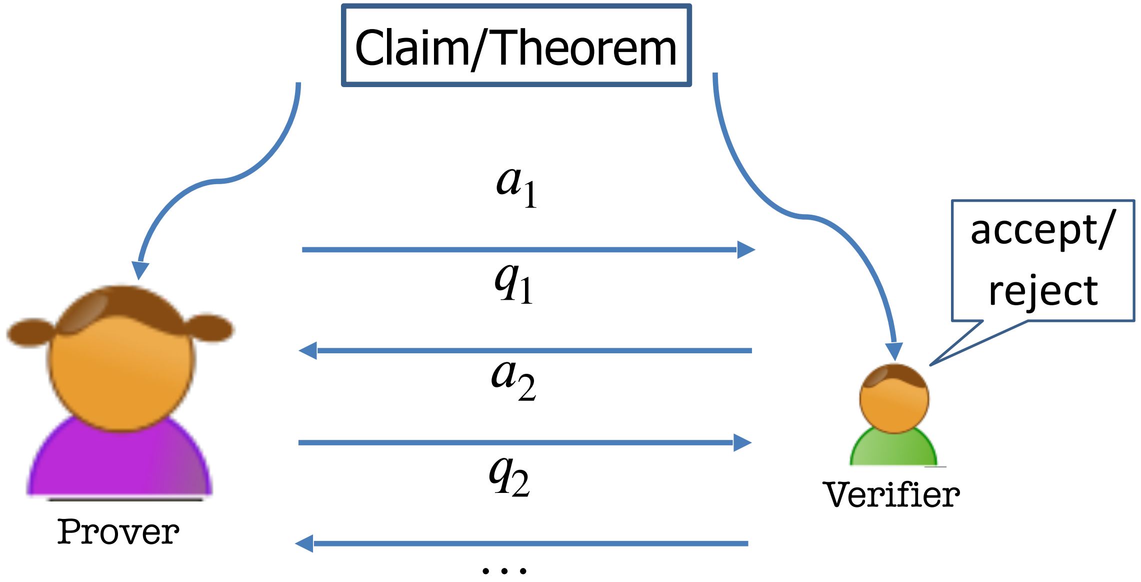
Announcements

- **HW 9** due **Wednesday Apr 17** at 11:59PM on Gradescope
- **HW10** will be released tomorrow evening/Thursday morning
  - Due **Wednesday Apr 24** at 11:59PM on Gradescope

Recap of last lecture

- What is a proof?
- Interactive Proofs
- *Zero-knowledge* interactive proofs
    - Definition
- ZKP for Graph Isomorphism
- ZKP for Quadratic Residuosity

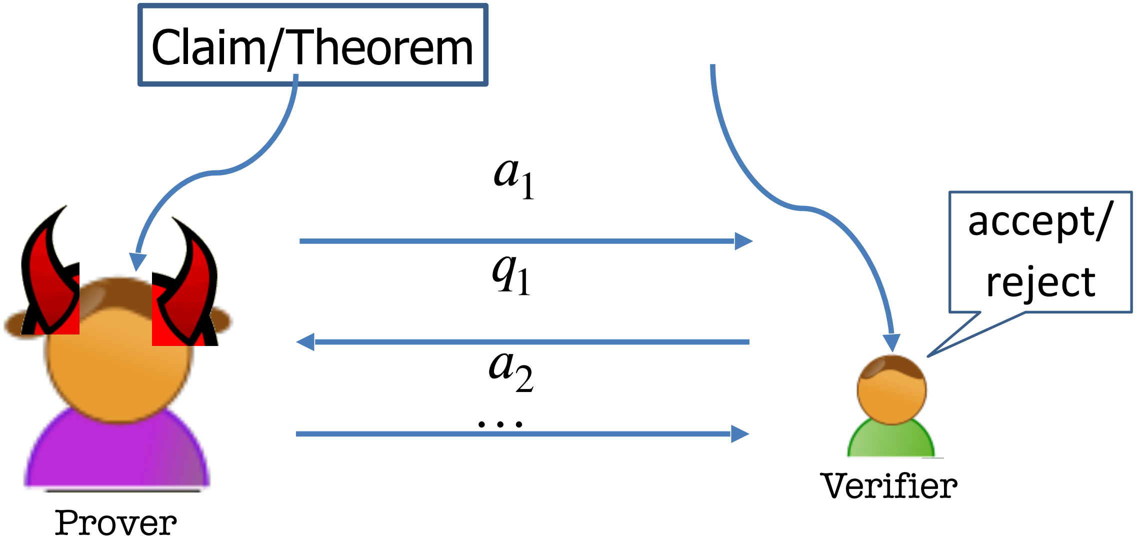
Interactive Proofs for a Language $\mathscr{L}$



Comp. Unbounded

Probabilistic
Polynomial-time

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



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

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

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**.

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.

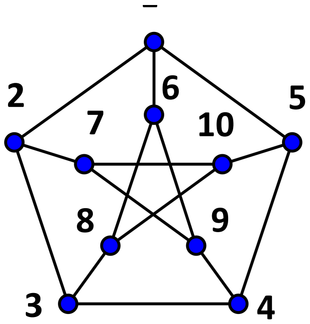
Computational Zero Knowledge: Definition

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**:

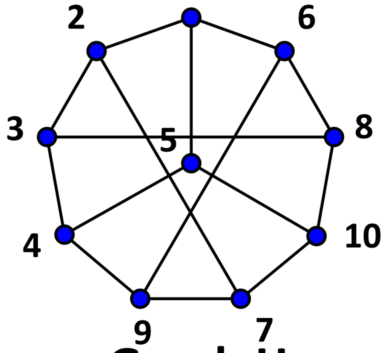
1. $view_V(P, V)$
2. $S(x, 1^\lambda)$

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

ZK Proof for Graph Isomorphism



Graph G



Graph H

$\mathbf{H} = \pi(G)$

$K = \rho(G)$

where ρ is a random permutation

random challenge bit b

Prover

Verifier

$b = 0$: send π_0 s.t. $\mathrm{K} = \pi_0(G)$

$b = 1$: send π_1 s.t. $\mathrm{H} = \pi_1(K)$

ZK Proof for QR

$$\mathscr{L} = \{(N, y) \mid \exists x \in \mathbb{Z}_N, y = x^2 \mod N\}.$$

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

(N, y)

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

(N, y)

If b=0: $z = r$ $\longrightarrow$

If b=1: $z = rx$

Check:

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

Today’s Lecture

- Malicious-verifier/“standard” ZK
  - ZKPs for GI and for QR achieve standard ZK
- ZKP for Quadratic Residuosity

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. $\text{view}_V(P, V)$ 2. $S(x, 1^\lambda)$

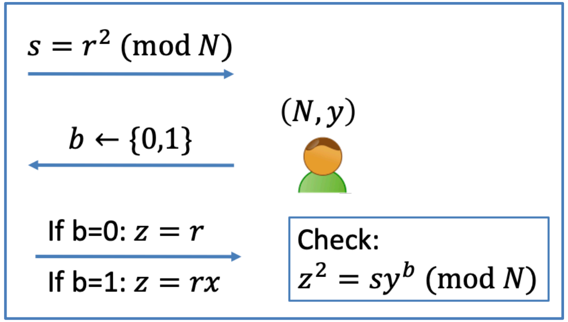
REAL DEF

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

Old: Honest-Verifier ZK

Claim: The QR protocol is honest-verifier zero knowledge.



$view_V(P, V)$:
(s, b, z)

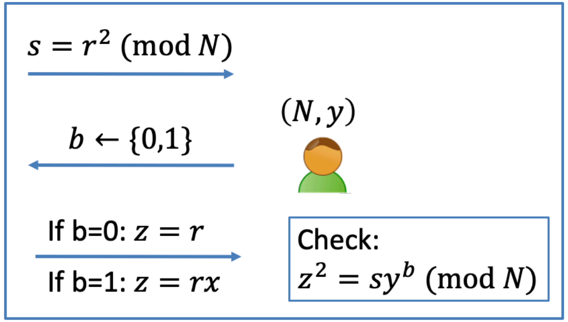
Simulator S works as follows:

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).

Now: Malicious Verifier ZK

Theorem: The QR protocol is (malicious verifier) zero knowledge.



$view_{V^*}\left(P, V^*\right)$:

(s, b, z)

Simulator S works as follows:

1. First pick a random s and "feed it to" V^*.

2. Let b $= V^*(s)$.

Now what???

Now: Malicious Verifier ZK

Theorem: The QR protocol is (malicious verifier) zero knowledge.

Simulator S works as follows:

1. First set $s = \dfrac{z^2}{y^b}$ for a random z and b and feed s to V^*.
2. Let $b' = V^*(s)$.
3. If $b' = b$, output (s, b, z) and stop.
4. Otherwise, go back to step 1 and repeat. (also called “rewinding”).

Simulator S works as follows:

1. First set $s = \dfrac{z^2}{y^b}$ for a random z and feed s to V^*.

2. Let $\mathrm{b}' = V^*(s)$.

3. If $b' = b$, output $(\mathrm{s}, \mathrm{b}, \mathrm{z})$ and stop.

4. Otherwise, go back to step 1 and repeat. (also called "rewinding").

Lemma:

(1) S runs in expected polynomial-time.

(2) When S outputs a view, it is identically distributed to the view of V^* in a real execution.

What Made it Possible?

1. **Each statement had multiple proofs** of which the prover chooses one at random.

2. **Each such proof is made of two parts**: seeing either one on its own gives the verifier no knowledge; seeing both imply 100% correctness.

3. **Verifier chooses to see either part, at random**. The prover's ability to provide either part on demand convinces the verifier.

Do all NP languages have Perfect ZK proofs?

We showed two NP languages with perfect ZK proofs. Can we show this for *all* NP languages?

Theorem [Fortnow'89, Aiello-Hastad'87] No, unless bizarre stuff happens in complexity theory (technically: the polynomial hierarchy collapses.)

Do all NP languages have

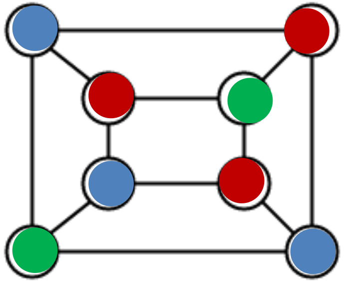
Winner of 2024 Turing Award!

Nevertheless, today, we will show:

<u>Theorem</u> [Goldreich-Micali-Wigderson'87] Assuming one-way functions exist, all of NP has computational zero-knowledge proofs.

This theorem is amazing: it tells us that everything that can be proved (in the sense of Euclid) can be proved in zero knowledge!

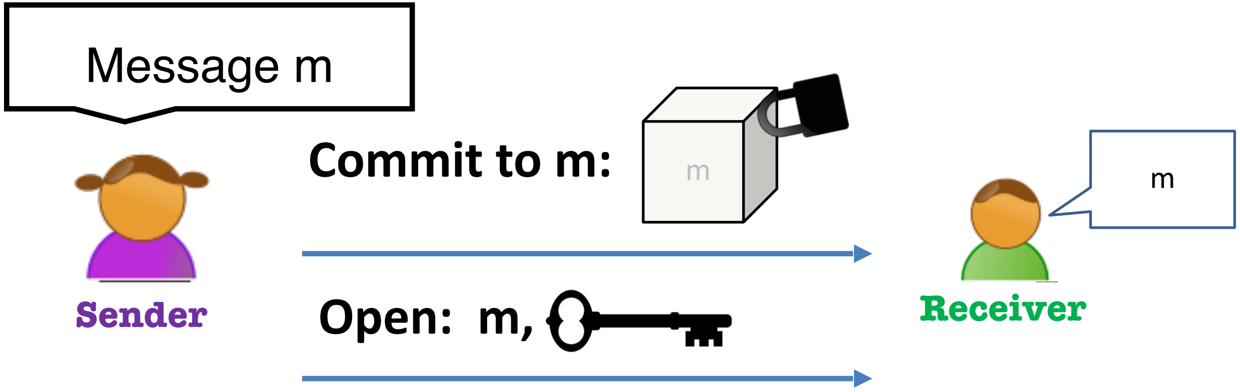
Zero Knowledge Proof for 3-Coloring



***NP-Complete* Problem:**

Every other problem in NP can be reduced to it.

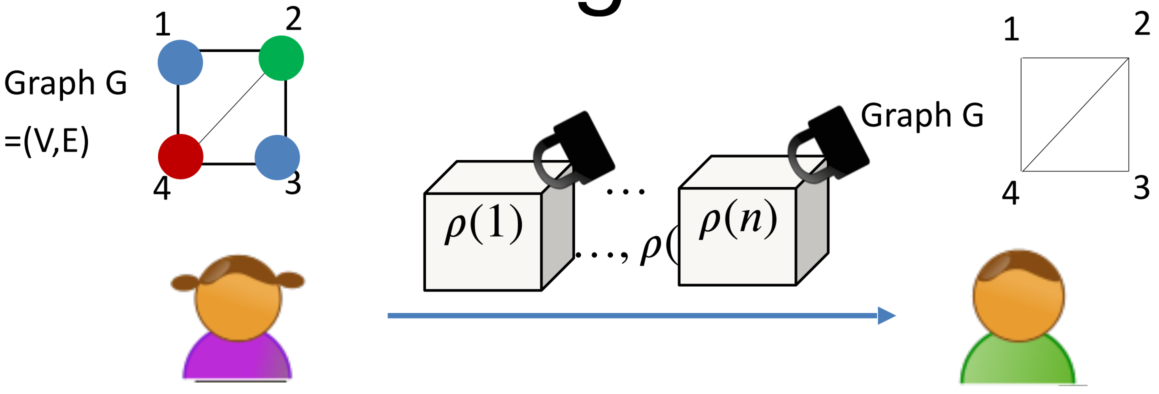
We need a *commitment scheme*



1. Hiding: The locked box should completely hide m.

2. Binding: Sender shouldn’t be able to open to different msg m’.

Zero Knowledge Proof for 3COL



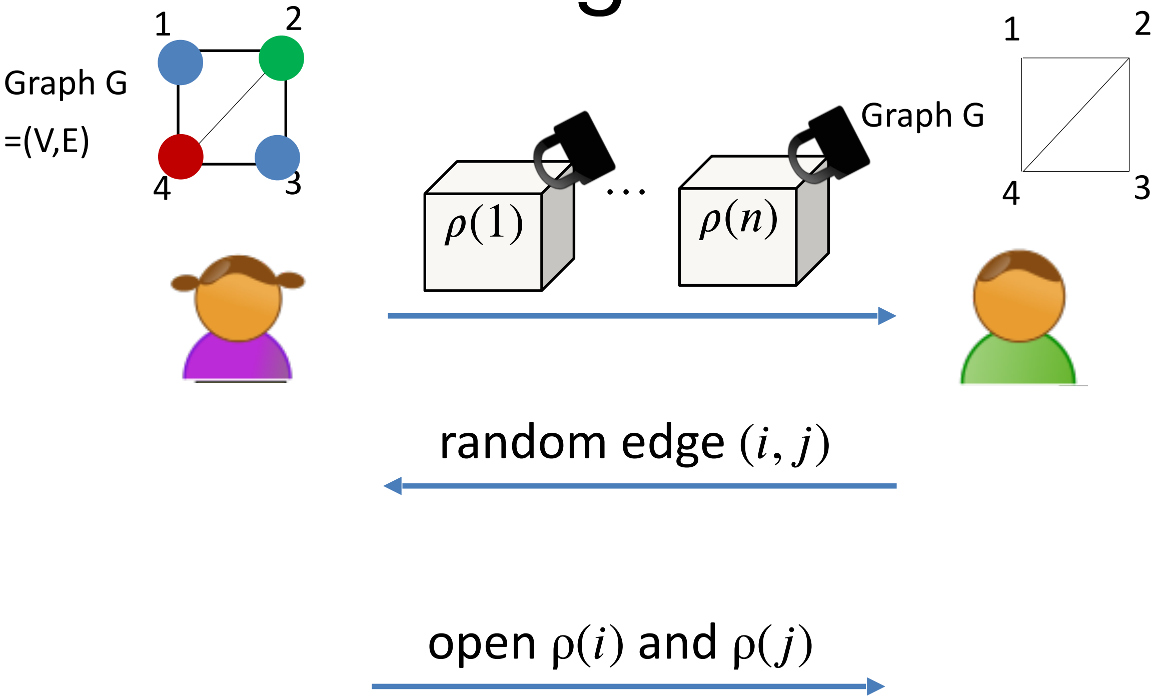
Come up with a random perm of the colors $\rho : V \to \{R, B, G\}$

random edge (i, j)

open $\rho(i)$ and $\rho(j)$

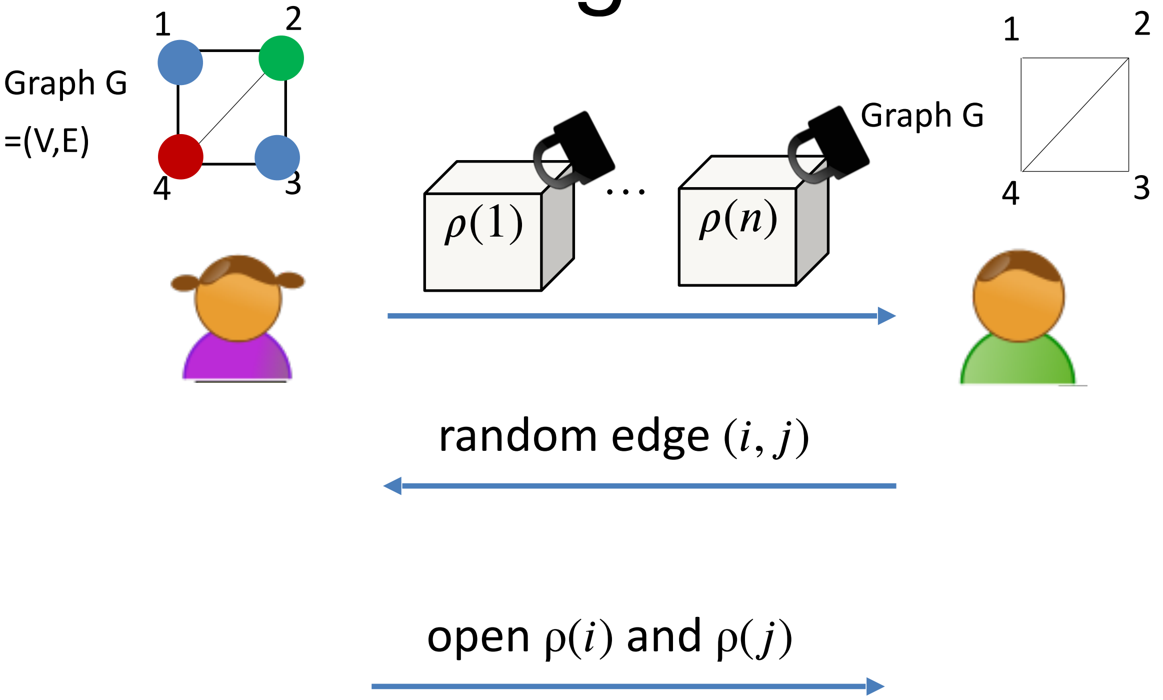
1. Check the openings
2. Check: $\rho(i),\ \rho(j) \in \{R, B, G\}$
3. Check: $\rho(i) \neq \rho(j)$.

Zero Knowledge Proof for 3COL



Completeness: Exercise.

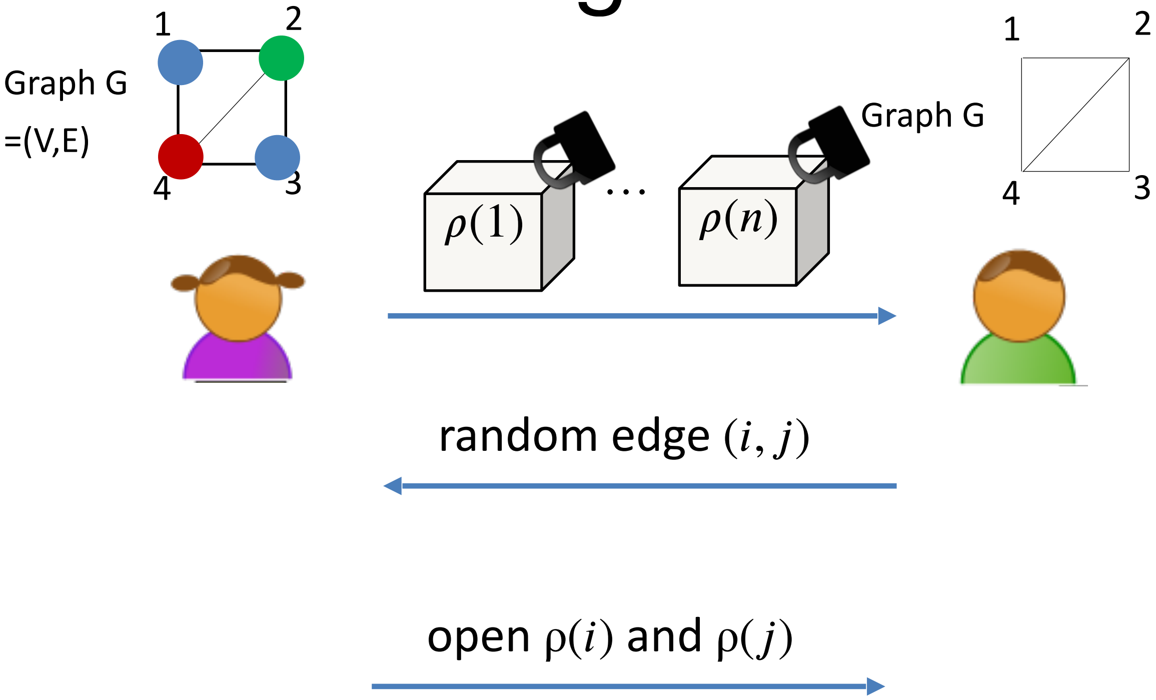
Zero Knowledge Proof for 3COL



Soundness: If the graph is not 3COL, in every 3-coloring (that P commits to), there is some edge whose end-points have the same color.

V will catch this edge and reject with probability $\geq 1/|E|$.

Zero Knowledge Proof for 3COL

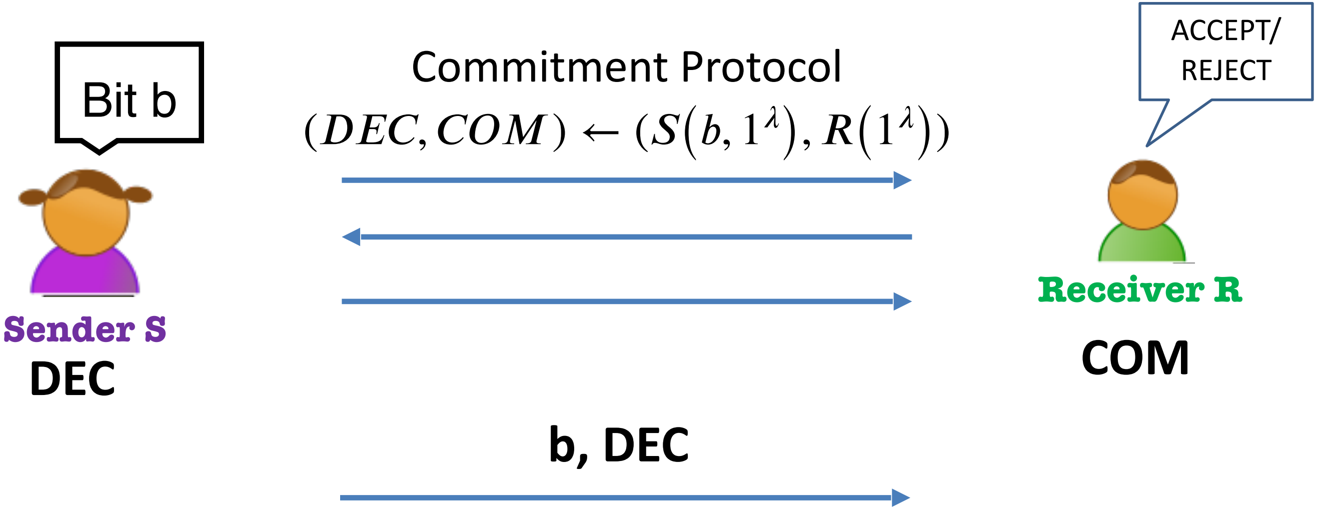


Repeat $|E| \bullet \lambda$ times to get the verifier to accept with probability

$$\leq (1 - 1/|E|)^{|E| \bullet \lambda} \leq 2^{-\lambda}$$

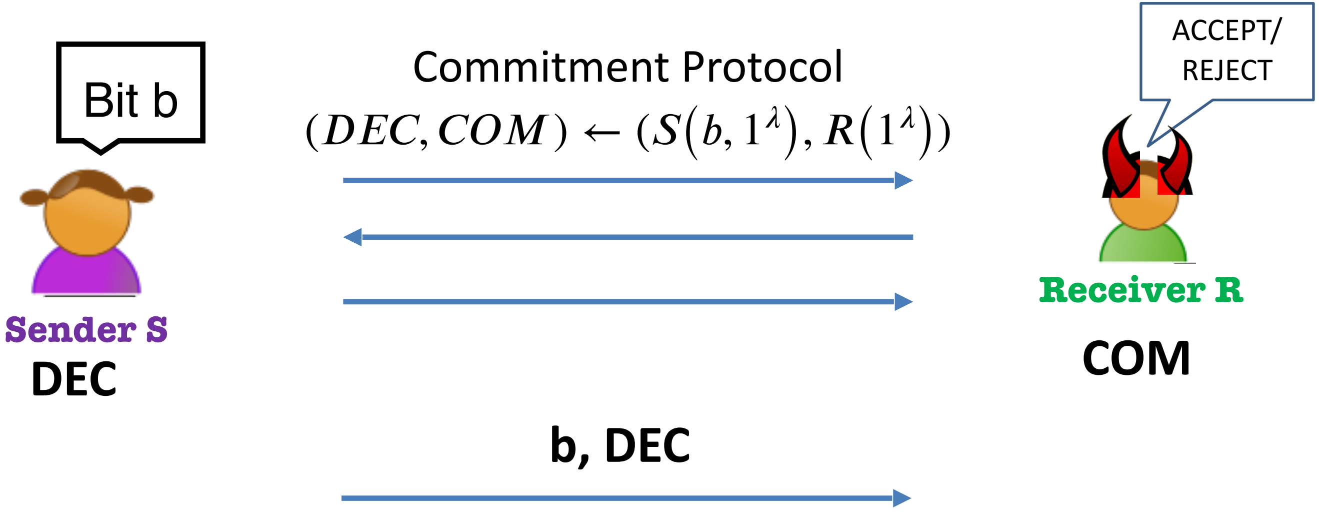
Constructing Commitment Schemes

Commitment Schemes



1. Completeness: R always accepts in an honest execution.

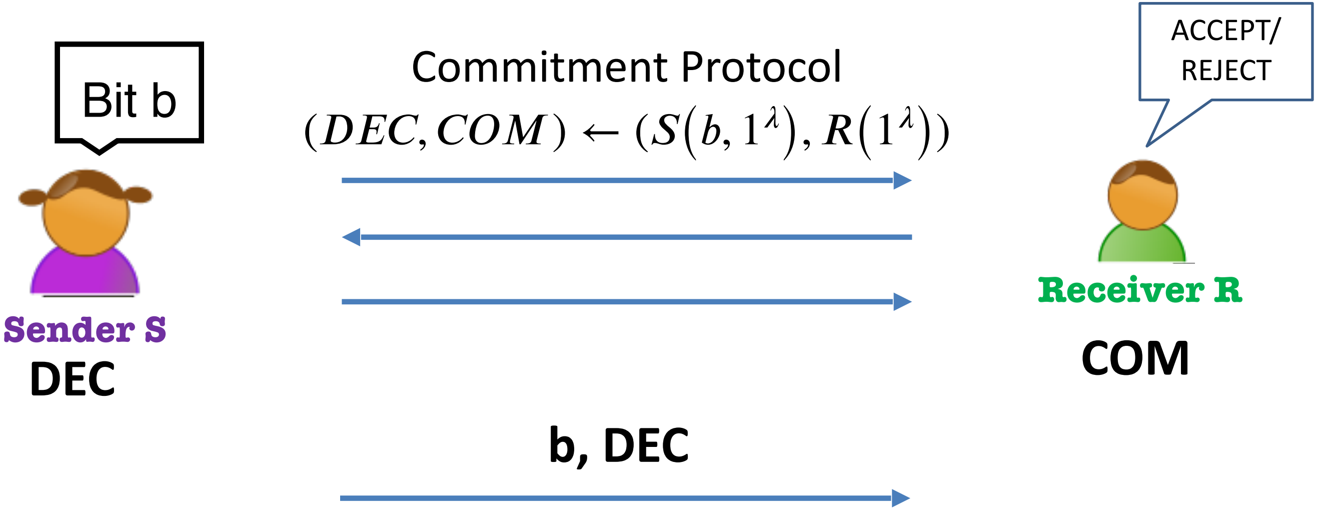
Commitment Schemes



2. Computational Hiding: For every possibly malicious (PPT) R^*,

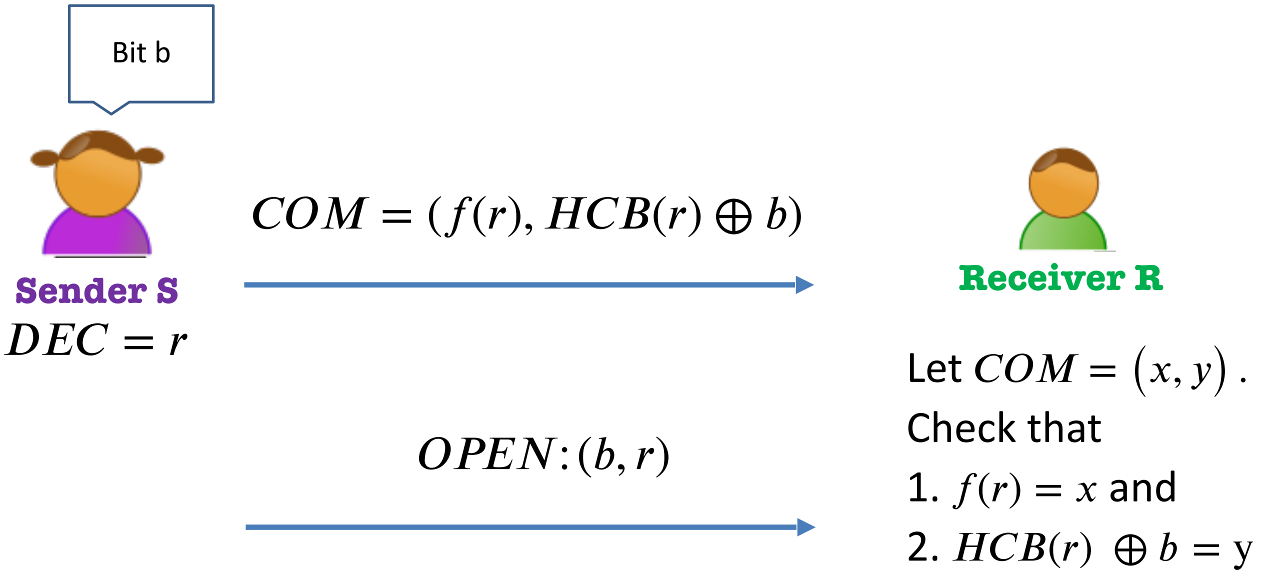
$$view_{R^*}(S(0), R^*) \approx_c view_{R^*}(S(1), R^*)$$

Commitment Schemes



3. Perfect Binding: For every possibly malicious S^*, let COM be the receiver's output in an execution of (S^*, R). There is no pair of decommitments (DEC_0, DEC_1) s.t. R accepts both $(\text{com}, 0, DEC_0)$ and $(\text{com}, 1, DEC_1)$.

A Commitment Scheme from any OWP



1. Completeness: Exercise.

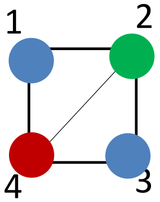
2. Comp. Hiding: by the hardcore bit property.

3. Perfect Binding: because f is a permutation.

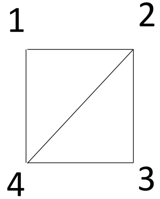
Back to ZK Proof for 3COL

Graph G

=(V,E)



Graph G



$\{Com\left(\rho(k); r_k\right)\}_{k=1}^{n}$

random edge (i, j)

send openings $\rho(i), r_i$ and $\rho(j), r_j$

Why is this zero-knowledge?

Simulator S works as follows:

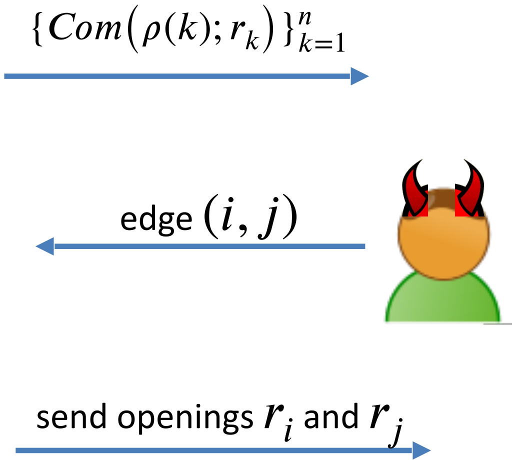
1. First pick a random edge (i^*, j^*)

   Color vertices i^* and j^* with random, different colors
   Color all other vertices red.

2. Feed the commitments of the colors to V^* and get edge (i, j)

3. If $(i, j) \neq (i^*, j^*)$, go back and repeat.

4. If $(i, j) = (i^*, j^*)$, output the commitments and openings r_i and r_j as the simulated transcript.

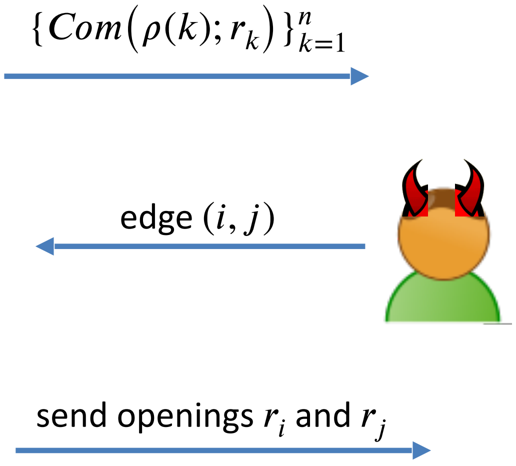


Why is this zero-knowledge?

Lemma:

(1) Assuming the commitment is hiding, S runs in expected polynomial-time.

(2) When S outputs a view, it is comp. indist. from the view of V^* in a real execution.



Why is this zero-knowledge?

Simulator S works as follows (call this Hybrid 0)

1. First pick a random edge (i^*, j^*)

   Color vertices i^* and j^* with random, different colors
   Color all other vertices red.

$\{Com(\rho(k); r_k)\}_{k=1}^{n}$

2. Feed the commitments of the colors to V^* and get edge (i, j)

edge (i, j)

3. If $(i, j) \neq (i^*, j^*)$, go back and repeat.

send openings r_i and r_j

4. If $(i, j) = (i^*, j^*)$, output the commitments and openings r_i and r_j as the simulated transcript.

Why is this zero-knowledge?

Not-a-Simulator S works as follows (call this Hybrid 1)

1. First pick a random edge (i^*, j^*)

   Permute a legal coloring and color all vertices correctly.

$\{Com\big(\rho(k); r_k\big)\}_{k=1}^{n}$

2. Feed the commitments of the colors to V^* and get edge (i, j)

edge (i, j)

3. If $\big(i, j\big) \neq (i^*, j^*)$, go back and repeat.

send openings r_i and r_j

4. If $\big(i, j\big) = (i^*, j^*)$, output the commitments and openings r_i and r_j as the simulated transcript.

Why is this zero-knowledge?

Claim: Hybrids 0 and 1 are computationally indistinguishable, assuming the commitment scheme is computationally hiding.

Proof: By contradiction. Show a reduction that breaks the hiding property of the commitment scheme, assuming there is a distinguisher between hybrids 0 and 1.

Why is this zero-knowledge?

Not-a-Simulator S works as follows (call this Hybrid 1)

1. First pick a random edge (i^*, j^*)

   Permute a legal coloring and color all vertices correctly.

$\{Com(\rho(k); r_k)\}_{k=1}^n$

edge (i, j)

2. Feed the commitments of the colors to V^* and get edge (i, j)

3. If $(i, j) \neq (i^*, j^*)$, go back and repeat.

send openings r_i and r_j

4. If $(i, j) = (i^*, j^*)$, output the commitments and openings r_i and r_j as the simulated transcript.

Why is this zero-knowledge?

Here is the real view of V* (Hybrid 2)

1. ~~First pick a random edge (i^*, j^*)~~

   Permute a legal coloring and color all edges correctly.

$\{Com(\rho(k); r_k)\}_{k=1}^{n}$

2. Feed the commitments of the colors to V^* and get edge (i, j)

edge (i, j)

3. ~~If $(i, j) \neq (i^*, j^*)$, go back and repeat.~~

send openings r_i and r_j

4. ~~If $(i, j) = (i^*, j^*)$,~~ output the commitments and openings r_i and r_j as the transcript.

Why is this zero-knowledge?

Claim: Hybrids 1 and 2 are identical.

Hybrid 1 merely samples from the same distribution as Hybrid 2 and, with probability $1 - 1/|E|$, decides to throw it away and resample.

Put together:

Theorem: The 3COL protocol is zero knowledge.

Examples of NP Assertions

- **My public key is well-formed** (e.g. in RSA, the public key is N, a product of two primes together with an e that is relatively prime to $\varphi(N)$.)
- **Encrypted bitcoin (or Zcash): "I have enough money to pay you."** (e.g. I will publish an encryption of my bank account and prove to you that my balance is $\geq \$X$.)
- **Running programs on encrypted inputs:** Given Enc(x) and y, prove that y = PROG(x).

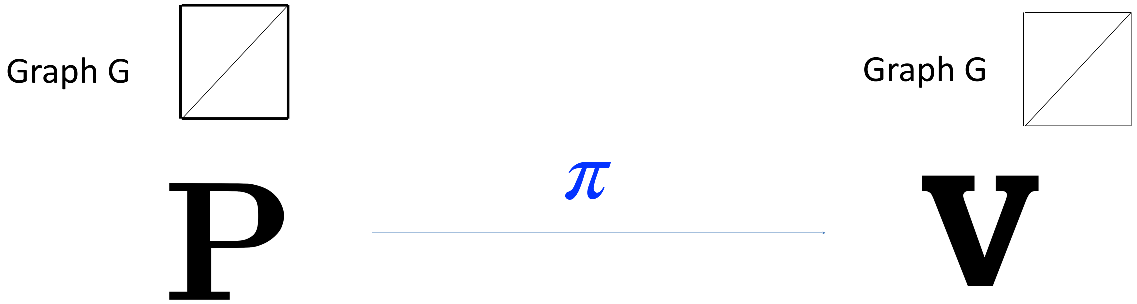
Examples of NP Assertions

- **Running programs on encrypted inputs:** Given Enc(x) and y, prove that y = PROG(x).

More generally: A tool to enforce honest behavior without revealing information.

Interaction is Necessary for ZK

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

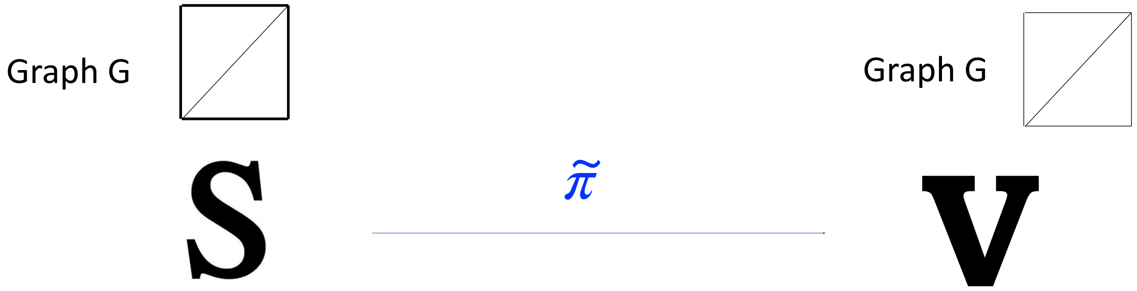


Step 1. When G is in 3COL, V accepts the proof π.

(Completeness)

Interaction is Necessary for ZK

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

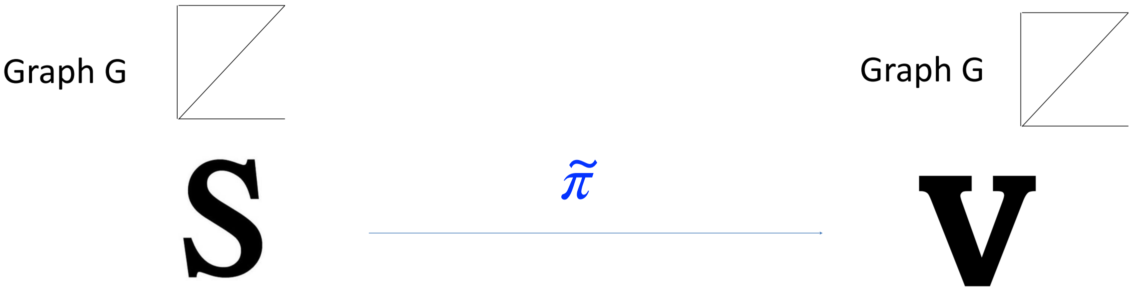


Step 2. **PPT** Simulator S, **given only G in 3COL**, produces an indistinguishable proof $\tilde{\pi}$ (Zero Knowledge).

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

Interaction is Necessary for ZK

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



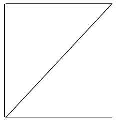
Step 3. Imagine running the Simulator S on a $G \notin$ 3COL.

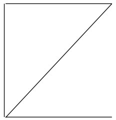
It produces a proof $\tilde{\pi}$ which the verifier still accepts!

(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.

Graph G 

Graph G 

$\mathbf{P} \xrightarrow{\pi} \mathbf{V}$

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!

$\blacksquare$