CSE 312

Foundations of Computing II

Lecture 27: Zero-Knowledge

- Please fill out the **class evaluation** by 12/10 Sunday !!!!!!!
  - Having your feedback is very important.
  - https://uw.iasystem.org/survey/279647

Classical Proofs

Euclid

The infinitude of primes

Gauss

Fundamental Theorem of algebra

Poussin, Hadamard

Prime Number Theorem

Efficiently Verifiable Proofs

Statement X: $x_1 + x_1^5 + x_2^7 = 0, x_1 - x_2^3 + 100x_1^6 = 0$ has a solution

Prover

P

Powerful

W: $x_1 = 100$ $x_2 = 50$

Verifier

V

Check,

Any other way?

V learns nothing?

V is convinced and <u>learns a solution</u>

NP Language: $x \in L$ iff $\exists w\ V_L(x, w) = 1$

Zero Knowledge Proofs [GMR89]

Prover $\quad\quad\quad\quad\quad\quad$ Verifier

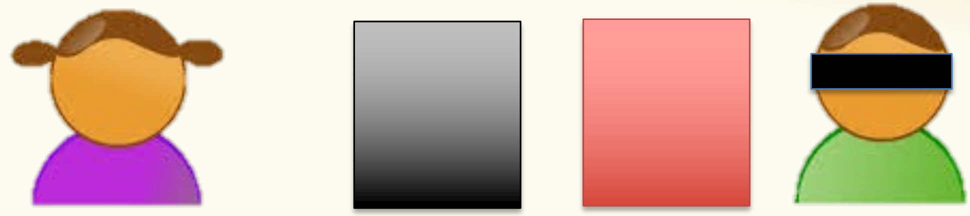
$$\mathsf{P}(x) \longrightarrow \mathsf{V}(x)$$

Goal: Prove that a statement x is true

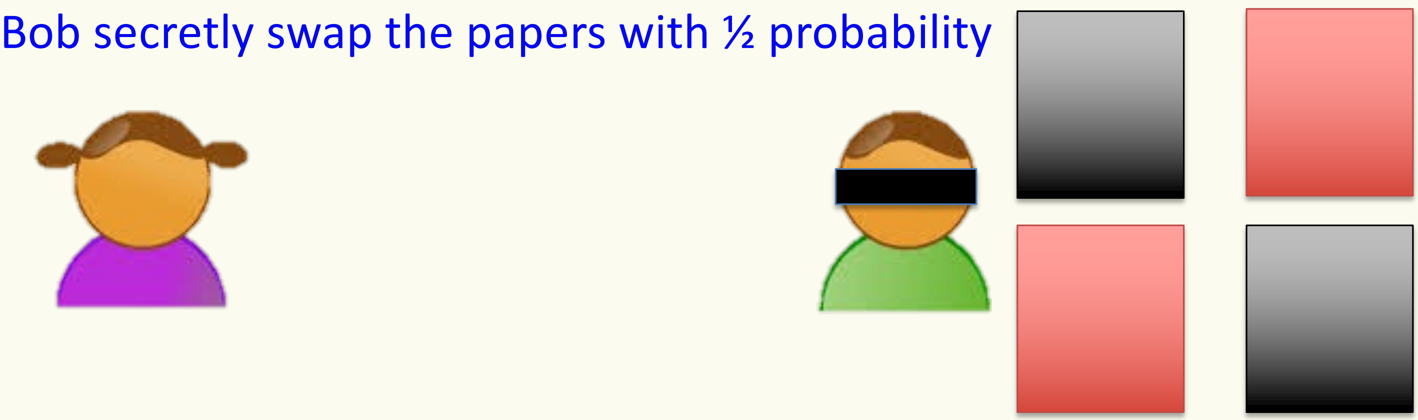
without revealing any information other than the validity

Paradoxical?

1. Bob holds the two papers out, Alice sees them



2. Bob secretly swap the papers with ½ probability



3. Bob holds the papers out. Alice guesses if Bob has swapped.

Interactive Proofs (IP)

Prover $\quad\quad\quad\quad$ Verifier

$$\mathsf{P}(x) \quad \mathsf{V}(x)$$

Fundamental Changes:

1. Use interaction
2. Use randomness / allow for error probability

Benefits:

1. ZK (today)
2. Verify way more statements efficiently e.g., verify an exponential time computation in polynomial time.

Interactive Proofs (IP)

A language is a set of true statement $L \subseteq \{0,1\}^*$.

An IP consists of a prover algorithm P and a verifier alg V.

V is efficient – polynomial time in $|x|$ (P may be inefficient)

Prover — Verifier

$$\mathrm{P}(x, w) \rightleftarrows \cdots \rightarrow \mathrm{V}(x) \rightarrow b \in \{0,1\}$$

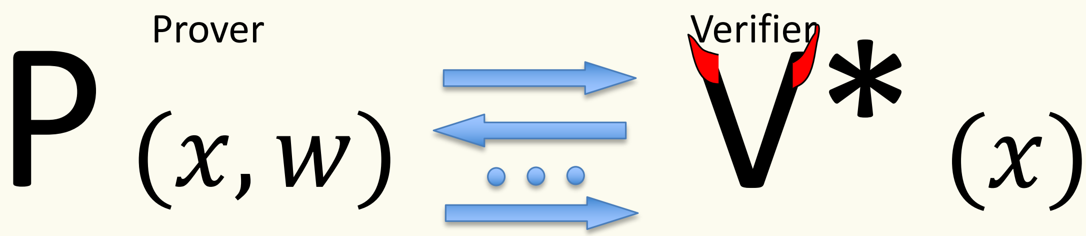
Interactive proof:

- **Correctness:** $\forall$ true statement x, P(x) convinces V(x) always
- **Soundness**: $\forall$ false statement x, $\forall$ *cheating prover P** (may not follow the honest prover algorithm) V(x) rejects with high probability $1 - \epsilon$ (e.g., $\epsilon = 0.01$)

Zero Knowledge (ZK) Proofs [GMR89]

Informally: An IP protocol for *L* is ZK if

- **Zero-knowledge:** ∀ true statement x, ∀ efficient cheating verifier *V** (may not follow the honest verifier algorithm), *V**(x) “learns nothing” about w from the interaction



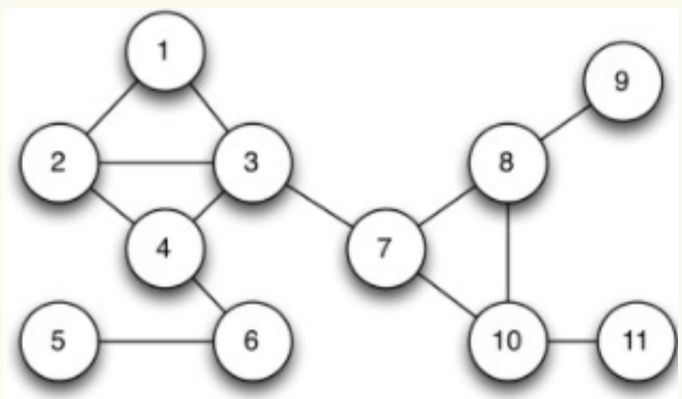
Interactive proof:

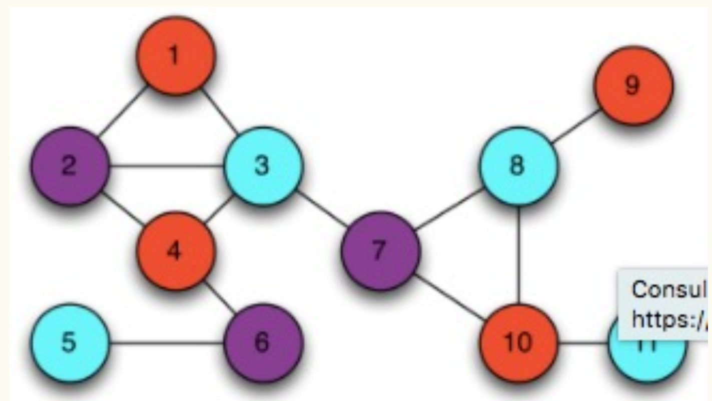
- **Correctness:** ∀ true statement x, P(x) convinces V(x) always
- **Soundness**: ∀ false statement x, ∀ *cheating prover P** (may not follow the honest prover algorithm) V(x) rejects with high probability $1 - \epsilon$ (e.g., $\epsilon = 0.01$)

Graph 3-Coloring

Problem. Given a graph $G = (V, E)$, Can the vertices be colored using one of three colors, so that, no two nodes connected by an edge have the same color?

Graph 3-coloring is NP-complete!

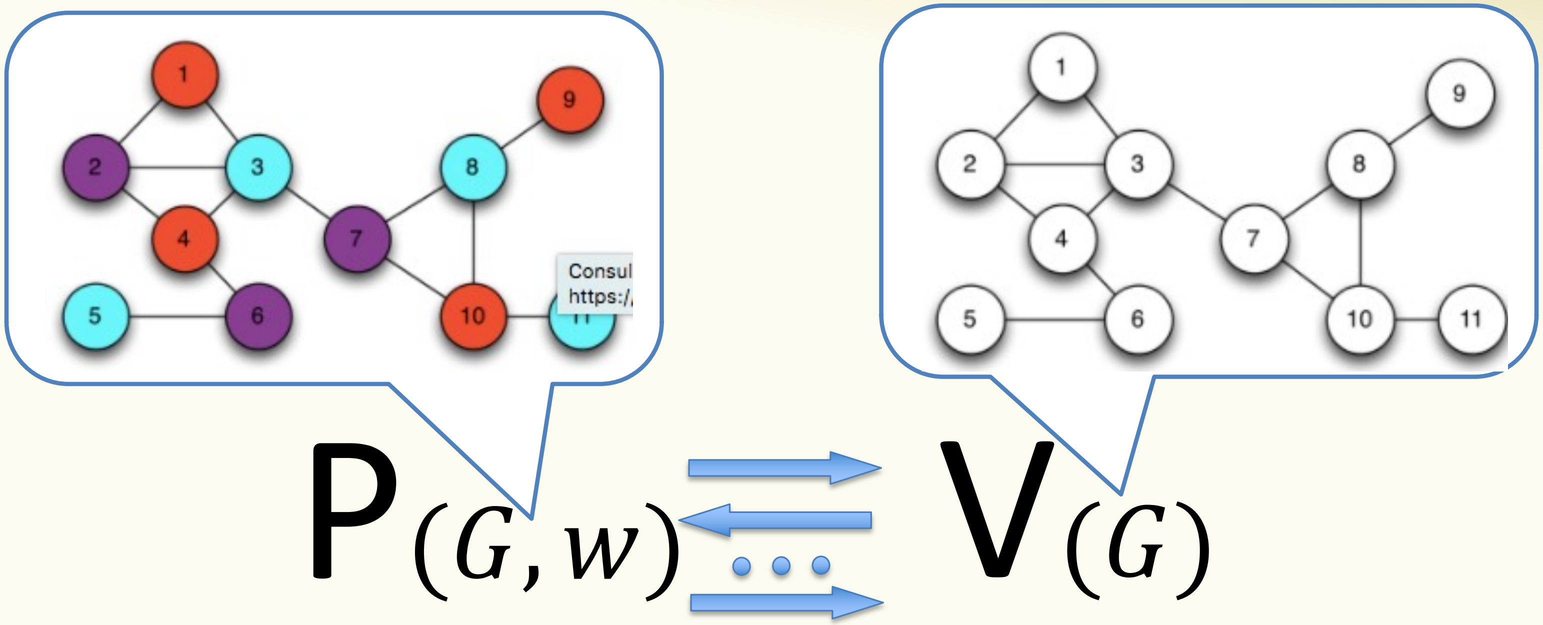




- Statement X: A graph G has a 3-coloring
- Solution W: A valid 3-coloring

Q: For the same graph, are there many valid 3-colorings?

A: Yes, in particular, permuting the colors gives valid coloring



Idea:

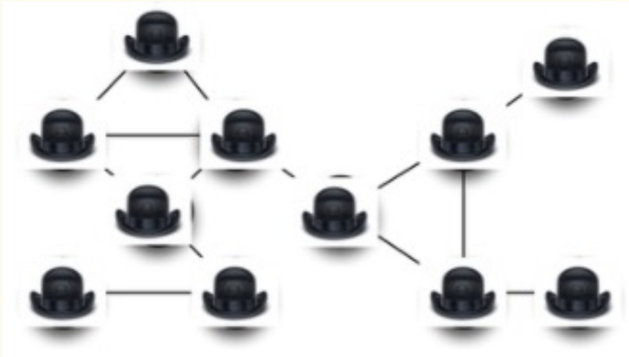
I am not going to give you the coloring.

But I will prove to you that I could if I wanted to …

$\mathsf{P}(G, w)$ $\mathsf{V}(G)$

1

P permutes the colors and covers colored vertices with hats (without V watching)



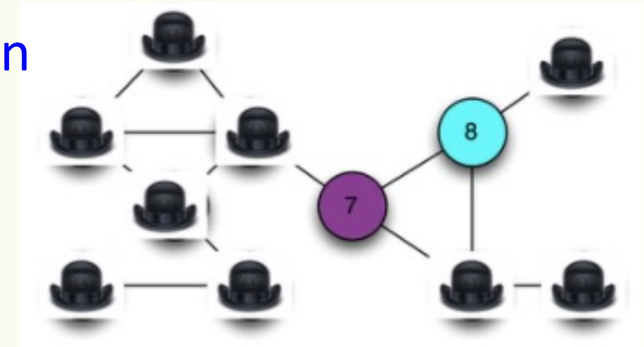
2

V chooses an edge at random

Edge $e = (7, 8)$

3

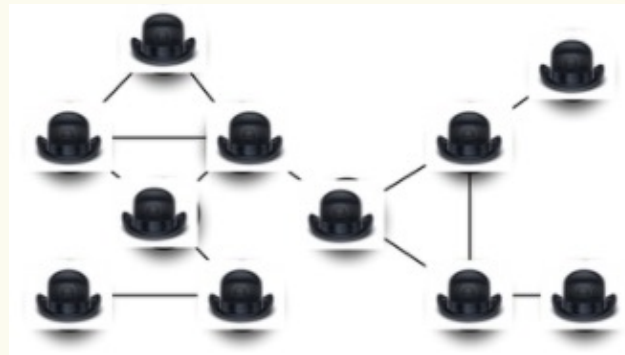
P removes hats on the two vertices connected by e (in front of V)



V rejects if they have same color

$\mathsf{P}(G, w)$ Zero-Knowledge $\mathsf{V}^*(G)$

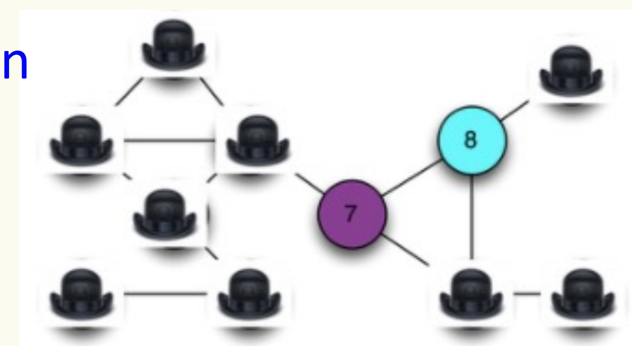
1 P permutes the colors and covers colored vertices with hats
(without V watching)



2 V* chooses an arbitrary edge,
not necessarily random

Edge $e = (7, 8)$

3 P removes hats on the two vertexes connected by e
(in front of V)



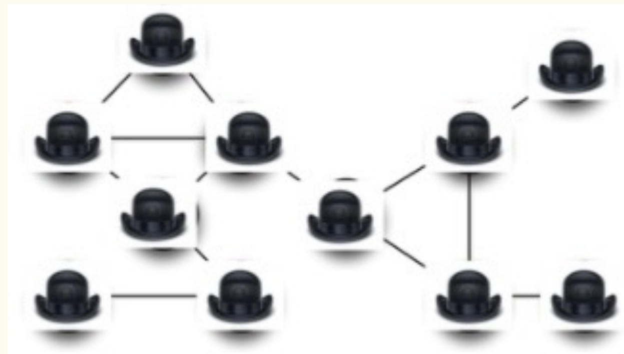
V rejects if they have same color

P* (G)

Soundness

V (G)

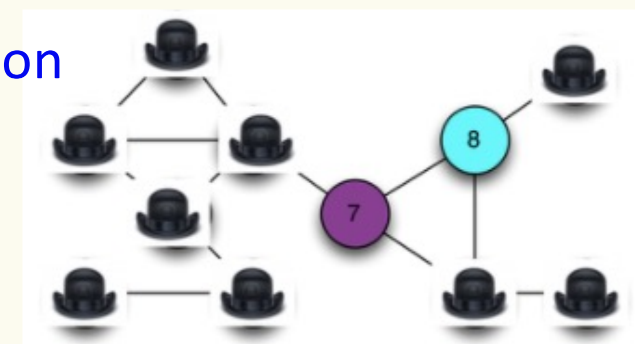
1 Color the vertices in arbitrary ways



2 V chooses an edge at random

Edge $e = (7, 8)$

3 P* removes hats on the two vertices connected by e (in front of V)



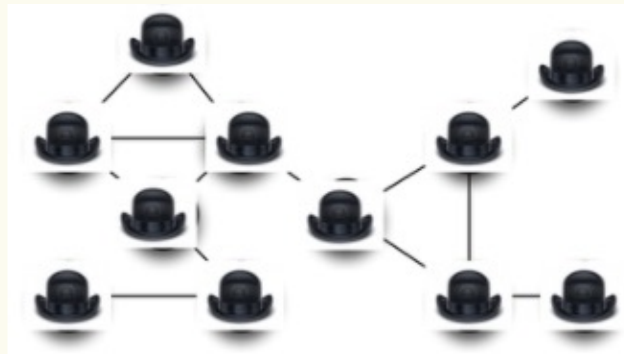
V rejects if they have same color

P* (G)

Soundness

V (G)

1 Color the vertices in arbitrary ways



Edge $e = (7,8)$

2 V chooses an edge at random

3

Let Col be the random variable describing the hidden coloring of G

$\forall$ coloring C of G, $\quad \Pr[\, V \text{ accepts} \mid Col = C] \leq 1 - 1/m$

By LTP, $\Pr[\, V\ accepts \mid \leq 1 - 1/m$

Reduce Soundness Error

Issue: The 3-move protocol has soundness error $\epsilon = 1 - \frac{1}{m}$, where m is the number of edges

Solution: Reduce the soundness error by repeating k times

$P^*_{(G)}$ **Soundness** $V_{(G)}$

accept

accept

Repeat k times

Let Col_i be the random variable describing i'th hidden coloring of G

$\forall i, \forall$ coloring $C_1 \cdots C_i$ of G,

$\Pr[\, V \text{ accept in } i'\text{th run} \mid Col_1 = C_1, \cdots, Col_i = C_i] \leq 1 - 1/m$

By chain rule, $\Pr[V \text{ accepts in all runs } |\mathrm{Col}_1 = C_1, \cdots Col_k = C_k] \leq \left(1 - \frac{1}{m}\right)^k$

By LTP, $\Pr[V\ accepts\ in\ all\ runs] \leq \left(1 - \frac{1}{m}\right)^k$

Reduce Soundness Error

Issue: The 3-move protocol has soundness error $\epsilon = 1 - \frac{1}{m}$, where m is the number of edges

Solution: Reduce the soundness error by repeating $k = m \cdot \lambda$

- The verifier rejects, as soon as any of these k runs results in rejection
- Soundness error is now $\left(1 - \frac{1}{m}\right)^k \leq e^{-\frac{1}{m}k} = e^{-\lambda}$

$\mathsf{P}_{(G,w)}$ **Zero-Knowledge** $\mathsf{V^*}_{(G)}$

Q: can V* ask about every edge, and learn the whole coloring?

A: No!!!! In every run P permutes the colors!

Edge $(3,5)$ assigned different colors

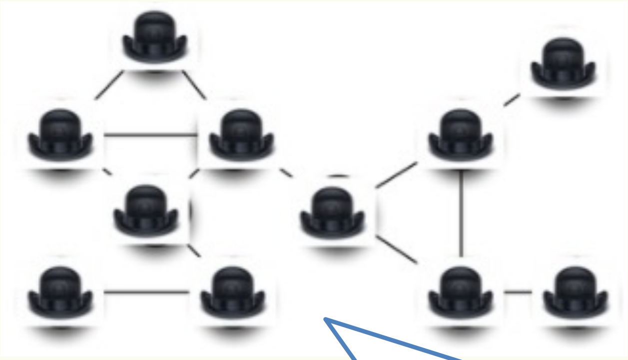
Edge $(7,20)$ assigned different colors

Edge $(3,5)$ assigned different colors

In each run V* learns that the endpoints of one edge can be colored with two different colors.

This is implied by the fact that G can be 3-colored and hence V* learns nothing.

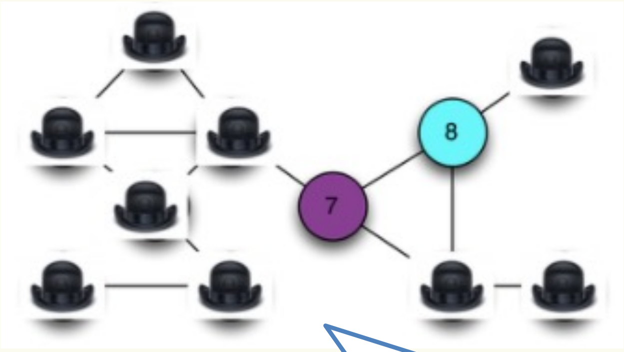
Cryptographic hats



Need a crypto tool s.t.

- Can “commit” to a color, while hiding it

Cryptographic hats



Need a crypto tool s.t.

- Can "commit" to a color, while hiding it
- Later, can "open" to a color, and there is only one color can be opened to