Interactive & Zero-Knowledge Proofs

CS 121 & CSCI E-207

Nov. 29 & Dec. 4, 2012

Salil Vadhan

Proofs & Complexity Theory

- NP = "languages L s.t. members of L have efficiently verifiable proofs"

- Def: A proof system for a language L is an algorithm V ("verifier") s.t.
  - Completeness ("true assertions have proofs"):
    $x \in L \Rightarrow \exists\, proof$ s.t. $V(x, proof) = \text{accept}$
  - Soundness ("false assertions have no proofs"):
    $x \notin L \Rightarrow \forall\, proof^* \;\; V(x, proof^*) = \text{reject}$
  - Efficiency
    V runs in time $\text{poly}(|x|)$

- NP = class of languages w/ proof systems.

Today: Two New Ingredients

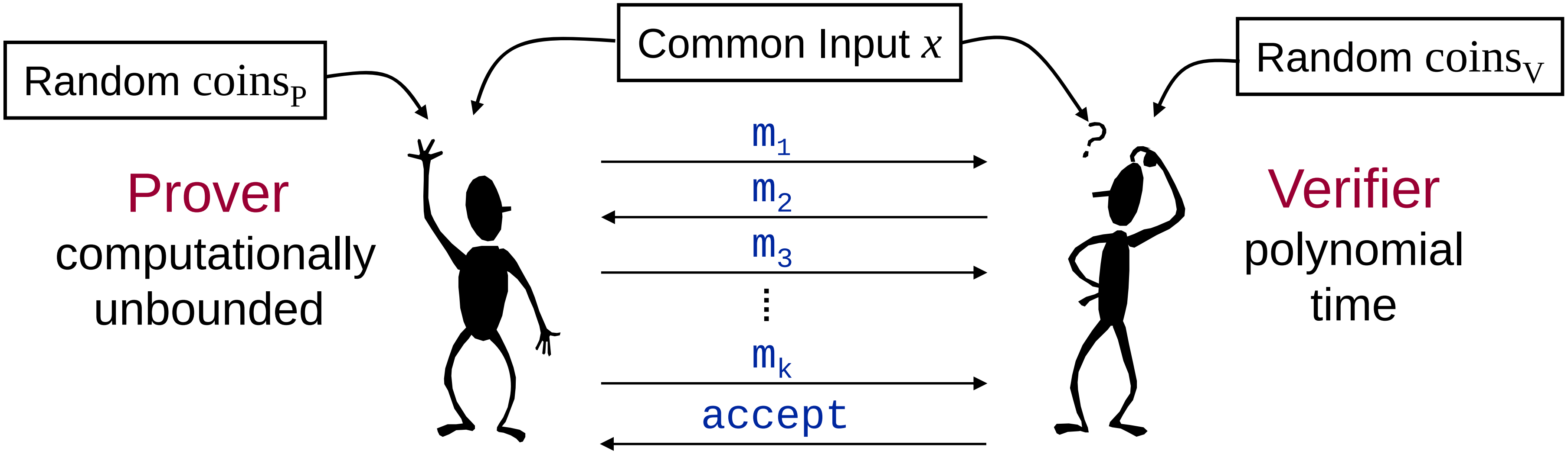
- Randomization: verifier can "toss coins"
  - Augment TM with extra tape of random bits
  - Allow verifier to err with small probability

- Interaction: replace static *proof* with dynamic, all-powerful *prover*
  - Will "interact" with verifier and try to "convince" it that assertion is true.

What can we gain?

- More general notion of "efficiently verifiable proofs"
- Greater efficiency in verification
  - verifier may not have to "read" entire proof
- Properties impossible in NP pfs ("zero knowledge")
- Cryptographic protocols.
- Connection to approximability of NP-complete problems.
  - E.g. Approximate size of largest clique in a graph within 1%.

Interactive Proofs



- Parties are functions $(x, \text{coins}, m_1,\ldots,m_{i-1}) \mapsto m_i$
- $m_i \in \Sigma^* \cup \{\text{accept, reject, halt}\}$

Interactive Proofs

Def: An interactive proof system for a language L is an interactive protocol (P,V) where

- Completeness: If $x \in L$, then

  V accepts in $(P,V)(x)$ with probability 1

- Soundness: If $x \notin L$, then for every P^*,

  V accepts in $(P^*,V)(x)$ with probability $\leq ½$

- Efficiency: V runs in time $\mathrm{poly}(|x|)$.

Def: IP = { L : L has an interactive proof }

Comments on Definition

- Probabilities taken only over coin tosses, not over input.
- Can reduce error probability (in soundness) to 2^{-1000} with 1000 repetitions.
- Interactive proofs generalize classical proofs: NP$\subseteq$IP.
  - Is IP bigger?

GRAPH ISOMORPHISM

- When are two graphs the "same" upto relabelling?

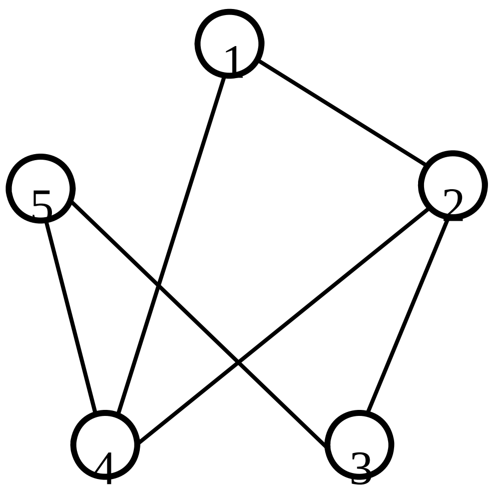
- Graph G with vertices $\{1,\ldots,n\}$ can be specified by (sorted) list of edges $E=\{(i_1, j_1), (i_2, j_2),\ldots,(i_m, j_m)\}$

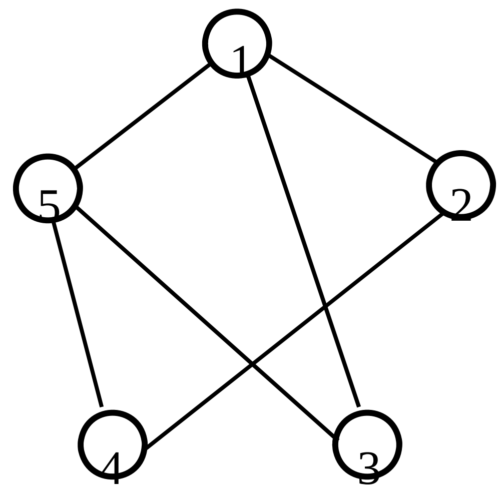
- Def: For $\pi : \{1,\ldots,n\}\to \{1,\ldots,n\}$ permutation (bijection), $\pi(G)$ = graph on $\{1,\ldots,n\}$ w/ edge set
$$E'=\{(\pi(i), \pi(j)) : (i, j)\in E\}$$

- Def: G is isomorphic to H (written $G\simeq H$) if $\exists\ \pi$ s.t. $\pi(G)= H$

Graph Isomorphism
Example 1

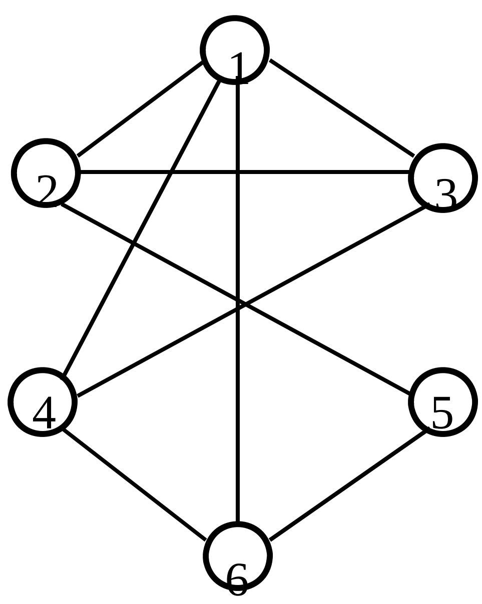
Are these graphs isomorphic?





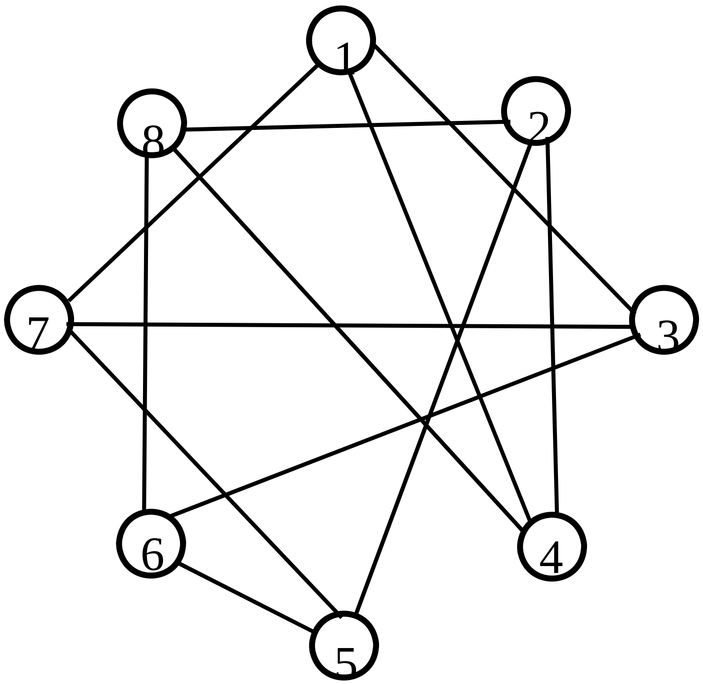
GRAPH ISOMORPHISM
Example 2

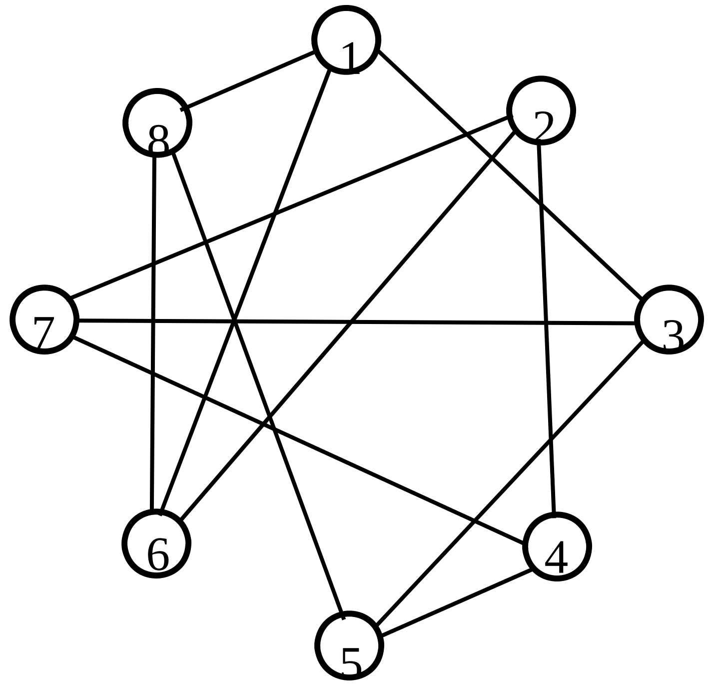
What about these graphs?



GRAPH ISOMORPHISM
Example 3

And these?





Graph Nonisomorphism

- Def: $\textsc{GraphIso} = \{(G_0, G_1) : G_0 \cong G_1\}$

  $\textsc{GraphNoniso} = \overline{\textsc{GraphIso}}$.

- $\textsc{GraphIso} \in$ NP (relabelling is a proof), but not known to be in P or to be NP-complete.

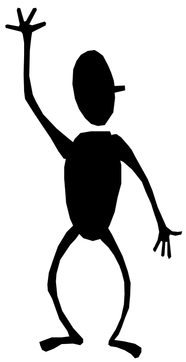
- $\textsc{GraphNoniso}$ not known to be in NP.

- Thm: $\textsc{GraphNoniso} \in$ IP

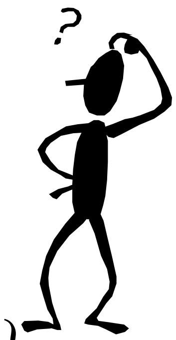
Proof System for GRAPHNONISO

(G_0, G_1)

Prover



Verifier



1. Flip $coin \in_R \{0,1\}$.

Choose random permutation π.

Let $H = \pi(G_{coin})$.

H

2. If $H \cong G_0$, let $guess = 0$, else let $guess = 1$.

$guess$

3. Accept if $guess = coin$.

Analysis of GRAPHNONISO Pf Sys.

Completeness: If $G_0 \ncong G_1$, then

- H is isomorphic to exactly one of G_0, G_1 (namely G_{coin}).

$\Rightarrow$ Prover always guesses correctly

$\Rightarrow$ Verifier accepts w.p.1

Soundness: If $G_0 \cong G_1$, then

- Every graph H isomorphic to G_0 is also isomorphic to G_1 & vice-versa. (+ distributions under random π are same)

$\Rightarrow$ H gives prover no information about coin.

$\Rightarrow$ Prover guesses correctly w.p. $\leq 1/2$ no matter what strategy it follows.

The Power of Interaction

- Have seen: an interactive proof for a language not known to have a classical proof system.
- Q: How much more powerful are interactive proofs?
- Thm: IP=PSPACE
  - Believed to be much larger than NP.
  - Contains all of co-NP.

What does one learn from a proof?

- The validity of the assertion being proven (by defn).
  Anything else?

- Classical (NP) proofs: Upon receiving a proof of statement x, one gains the ability to prove x to others.

- Interactive proofs: Can be "zero knowledge", i.e. reveal nothing other than the validity of the assertion being proven.

  $\Rightarrow$ verifier does not gain ability to prove same assertion to others!

Zero-Knowledge Proofs
GRAPHNONISO

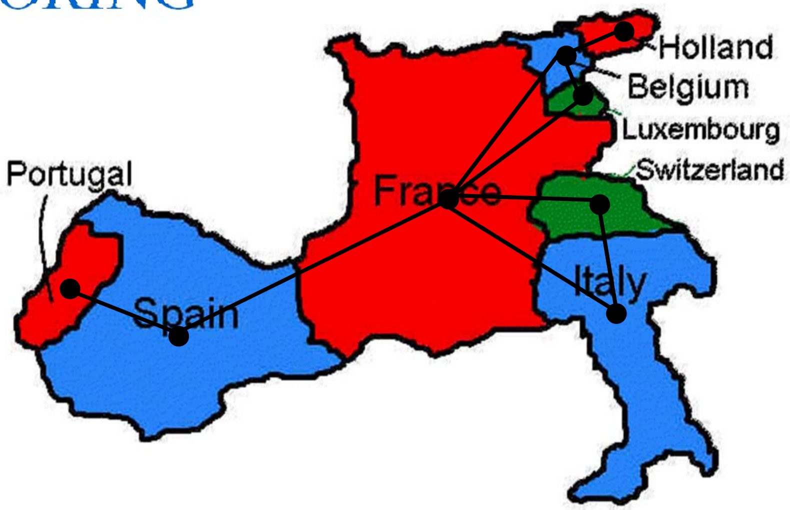
- In GRAPHNONISO proof system:
  - Only thing prover sends verifier is *guess*.
  - When $G_0 \not\cong G_1$, *guess* always equals verifier's *coin*.
  - Verifier "already knew" this value.
  - $\Rightarrow$ zero knowledge!

Comments:

- Only require zero-knowledge condition for inputs $x \in L$.

- Reasoning above relies on verifier following protocol.
  - Bad for cryptographic applications.
  - Can fix this by more complicated protocol.

MAP 3-COLORING



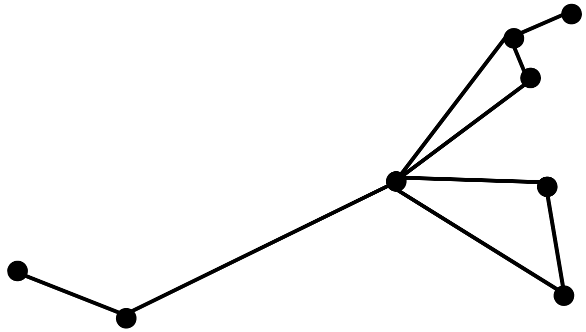
http://www.ctl.ua.edu/math103/

- Given: a map M
  Decide: can it be colored w/3 colors s.t. no two adjacent countries have the same color?

- Formally: 3-COL = { maps M : M is 3-colorable}

- Fact: 3-COL is NP-complete.

GRAPH 3-COLORING

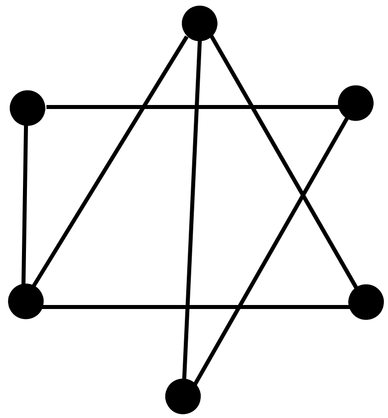


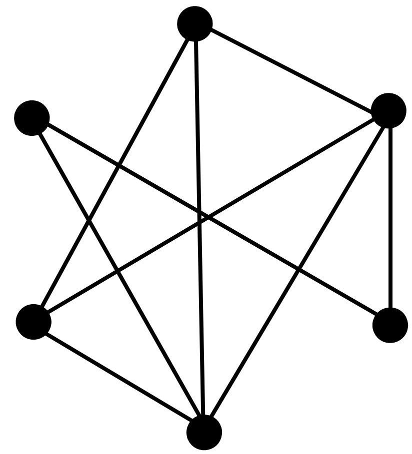
- Given: a graph G
  Decide: can it be colored w/3 colors s.t. no two adjacent vertices have the same color?

- Formally: 3-COL = { graphs G : G is 3-colorable}

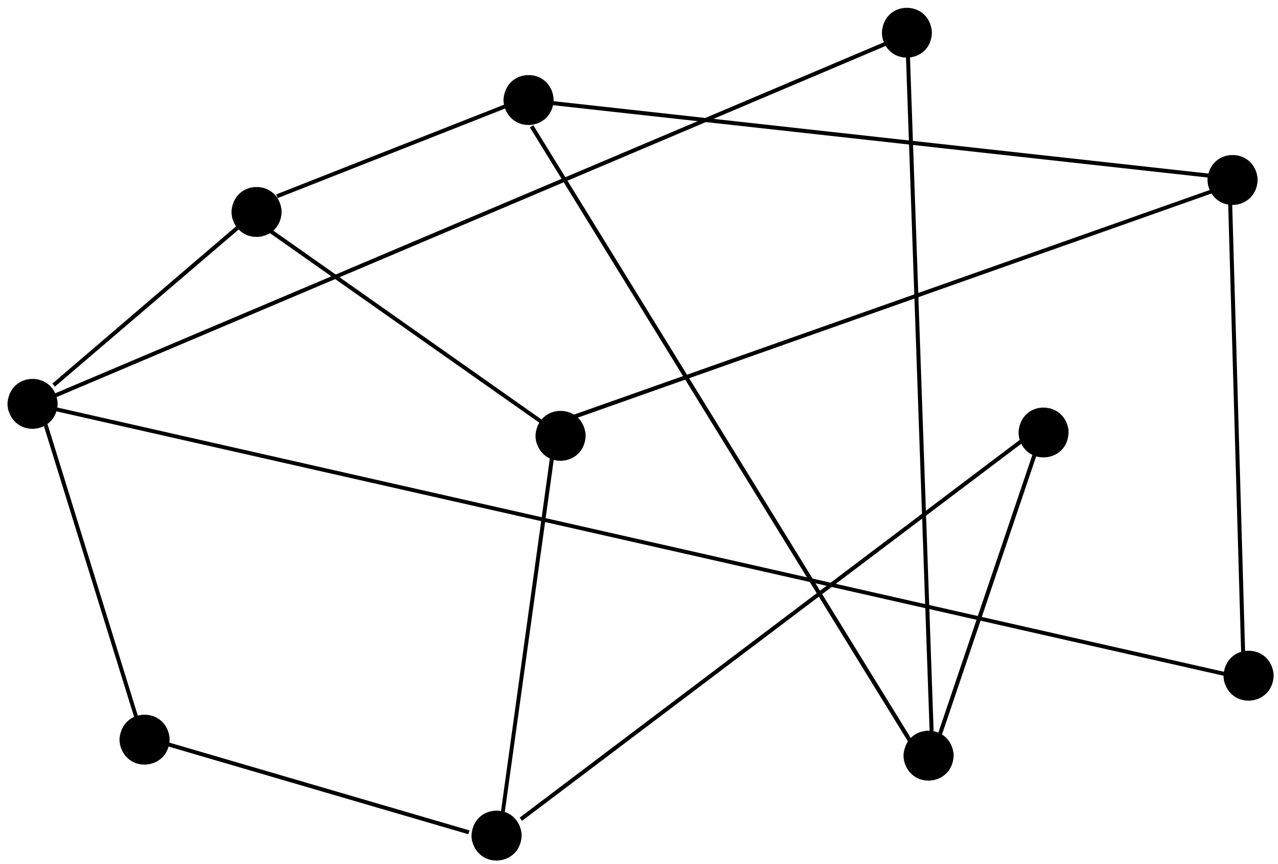
- Fact: 3-COL is NP-complete.

Graph 3-Coloring





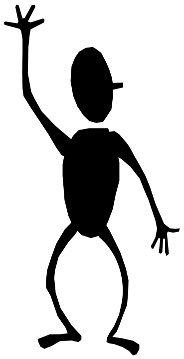
Claim: the following graph is 3-colorable



3-COL Proof System

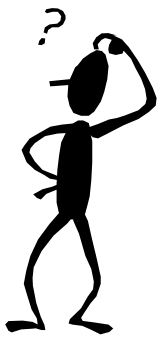
Prover

3-coloring
C of V



graph G

Verifier



1. Choose random permutation π of $\{R,G,B\}$. Let $C' = \pi(C)$.

"commit" to coloring C' →

2. Choose random edge (x,y).

← (x,y)

"reveal" $C'(x)$, $C'(y)$ →

3. Accept if $C'(x) \neq C'(y)$

Analysis of “Physical” 3-COL Proof Sys.

Completeness:

- If C is a proper 3-coloring, so is C'.

$\Rightarrow$ For every edge (x,y), $C'(x) \neq C'(y)$

$\Rightarrow$ Verifier accepts w.p. 1.

Soundness:

- Prover committed to some C' after step 1.

$\Rightarrow$ Since G is not 3-colorable, then for some edge (x,y), $C'(x) = C'(y)$

$\Rightarrow$ Verifier accepts w.p. $\leq 1\text{-}1/m$, where m = # edges

(repeat m times to get error prob. to $(1\text{-}1/m)^m < 1/2$.)

Analysis of "Physical" 3-COL Proof Sys. (cont.)

Zero Knowledge:

- All verifier sees are commitments & colors on one edge.
- Commitments reveal nothing (in physical implementation).
- Colors on one edge = random pair of distinct colors.
- Verifier can generate random pair of distinct colors on its own, without prover.

$\Rightarrow$ zero knowledge!

“Digital” Implementation?

- Need way to “commit” to coloring C' s.t.
  – Binds prover to C', i.e. cannot later change its mind about colors of any vertices.
  – Yet reveals nothing to verifier.

- Impossible? NO
  – Key observation: only need it to “reveal nothing” to a polynomial-time algorithm.
  – Cryptography provides such commitments.

- Thm: Every language in NP has a zero-knowledge pf (assuming $\exists$ commitments).
  – Pf: 3-COL is NP-complete $\Rightarrow$ can reduce any NP problem to it.

Defining Zero Knowledge

- How to formalize "Verifier learns nothing"?

Simulation Paradigm (informally):

- Require: anything that can computed in poly-time by interacting with prover can also be computed in poly-time without interacting with prover.

- That is, for every poly-time verifier V^*, there exists a poly-time simulator S s.t.

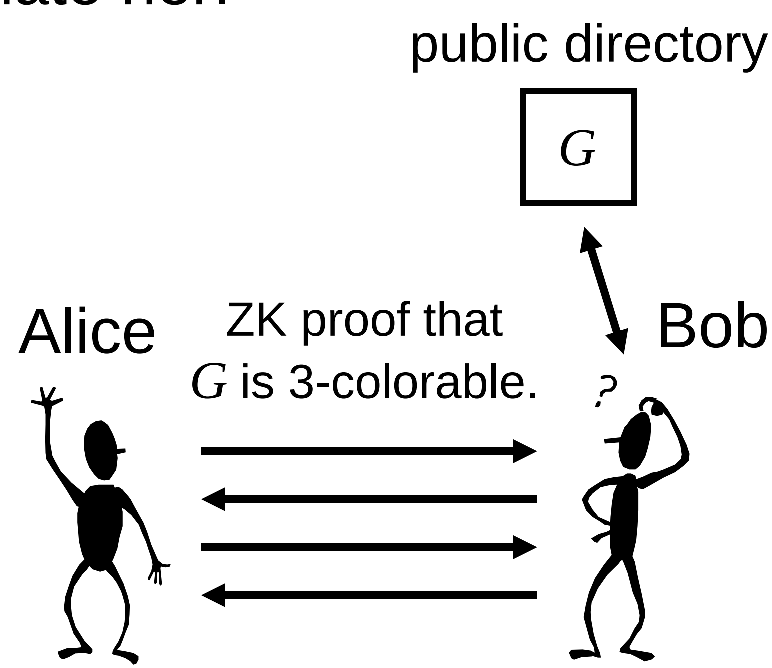
[output of $S(x)$] $\approx$ [output of V^* after interacting w/ P on x].

An Application: Identification

- Alice wants to securely identity herself to Bob.
- Traditional “password” solutions: Bob learns Alice’s password & can later impersonate her.

Using zero-knowledge (ZK) proofs:

- Alice publishes a graph G s.t. only she knows a 3-coloring.
- ZK property $\Rightarrow$ Bob (or an eavesdropper) cannot later impersonate Alice.



For more on these topics

- Sipser Sec. 10.4
- "Interactive and Zero-Knowledge Proofs." Lecture Notes from Park City Math Institute Graduate Summer School 2000. http://www.eecs.harvard.edu/~salil/research.html
- Oded Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness.* Springer-Verlag, 1998.
- Oded Goldreich. *Foundations of Cryptography — Volume I (Basic Tools).* Cambridge University Press, 2001.

  See http://www.wisdom.weizmann.ac.il/~oded/.
- Courses: CS 220r, 221, MIT 18.405J, 18.425J