18734: Foundations of Privacy

Zero Knowledge

Anupam Datta

CMU

Fall 2016

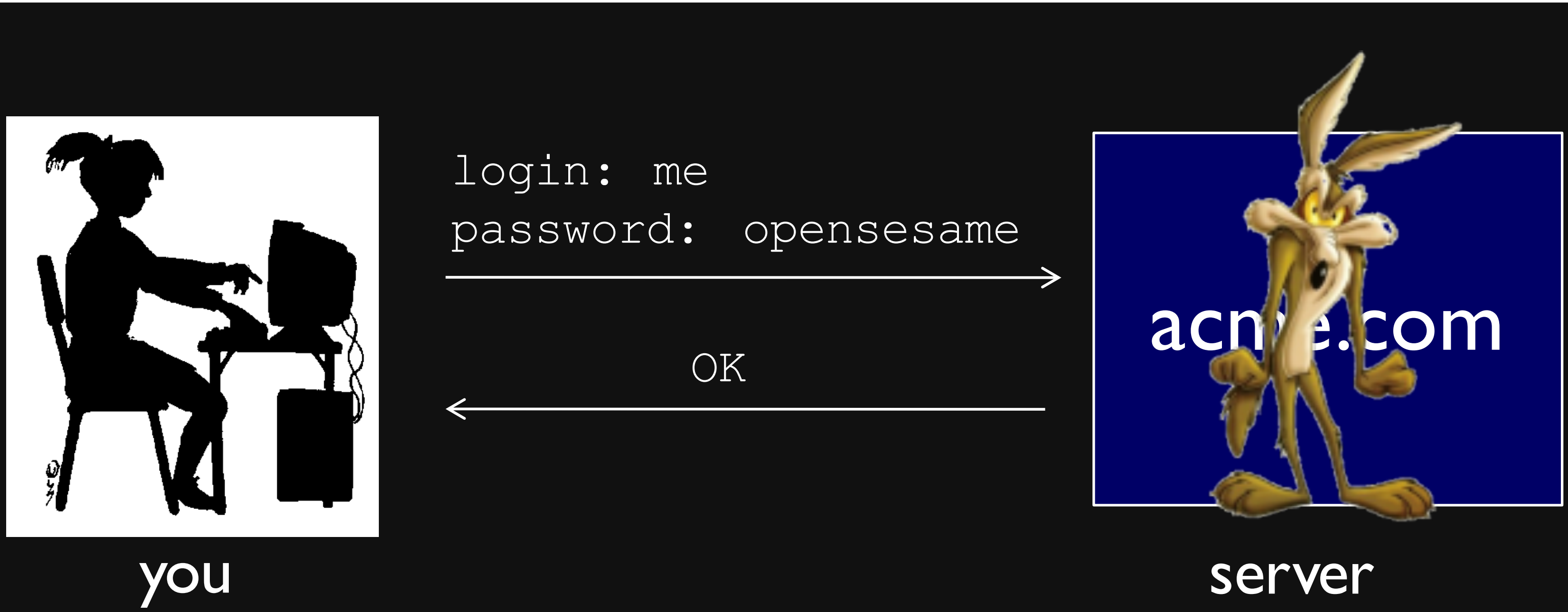
Authentication

Sign in to Gmail with your Google Account

☑ Remember me on this computer.

What happens when you type in your password?

Naïve authentication



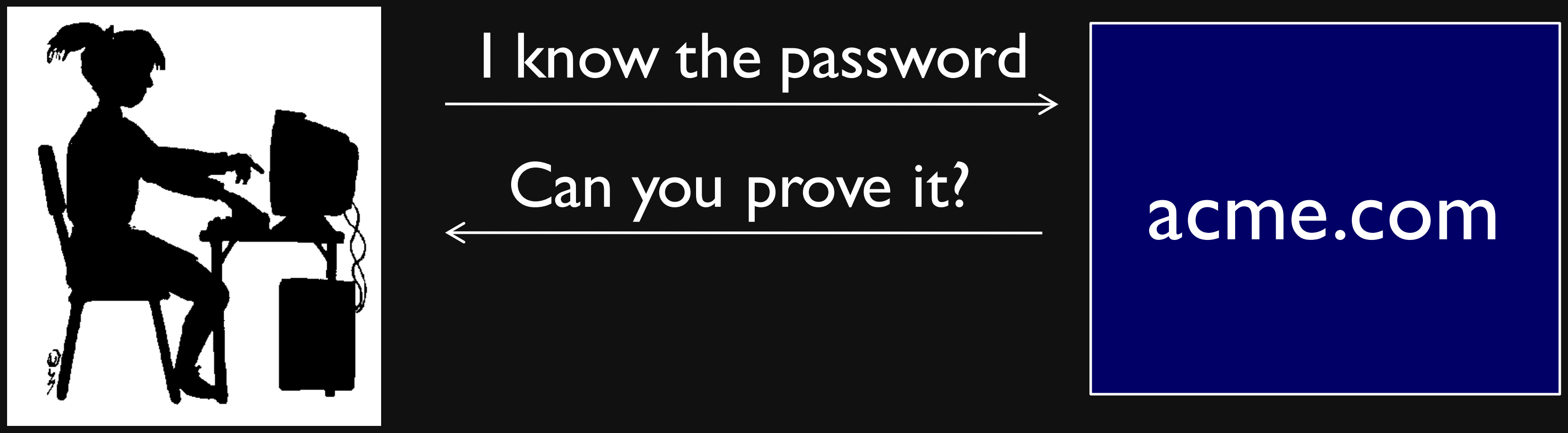
```
login:  me
password:  opensesame
```

OK

- The server knows your password

- So they can impersonate you at other web sites where you use the same password

“Zero-knowledge” authentication



I know the password

Can you prove it?

acme.com

Can you convince the server that you know your password, without revealing it (or any other information)?

What is knowledge?

What is ignorance?

(lack of knowledge)

- Example 1: Tomorrow’s lottery numbers

2 31 12 7 28 11

We are ignorant of them because they are random

What is ignorance?

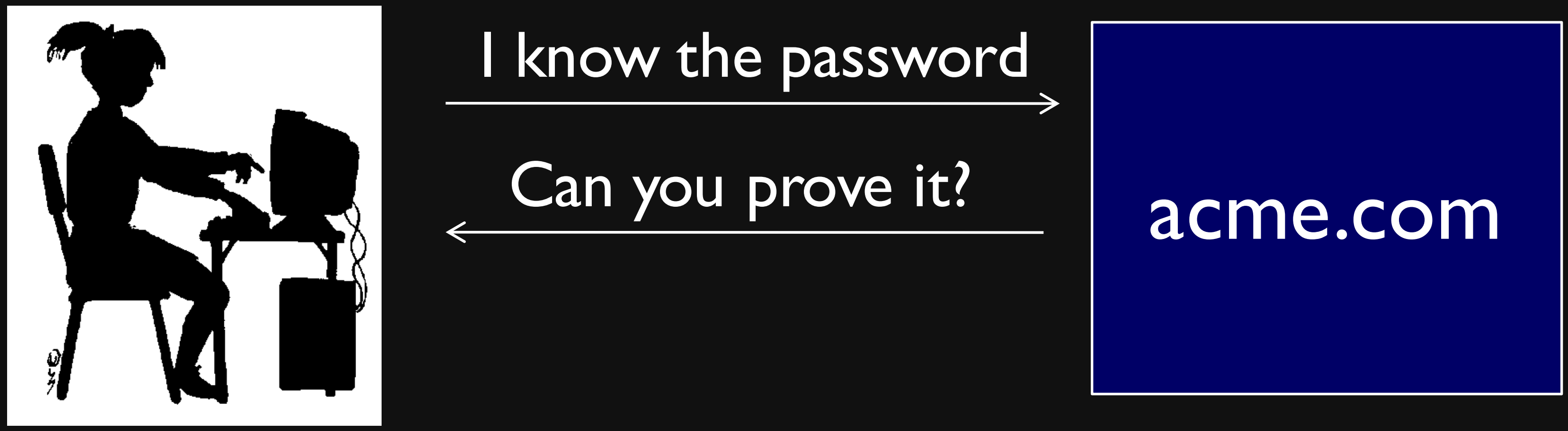
- Example 2: A difficult math problem

Prove that $\mathrm{P} \neq \mathrm{NP}$

We are ignorant because it takes a lot of work to figure out the answer

- Questions of this type include
  - Finding satisfying assignments to Boolean formulas
  - Finding cliques in graphs
  - All NP-hard problems

Using ignorance to our advantage



I know the password →

← Can you prove it?

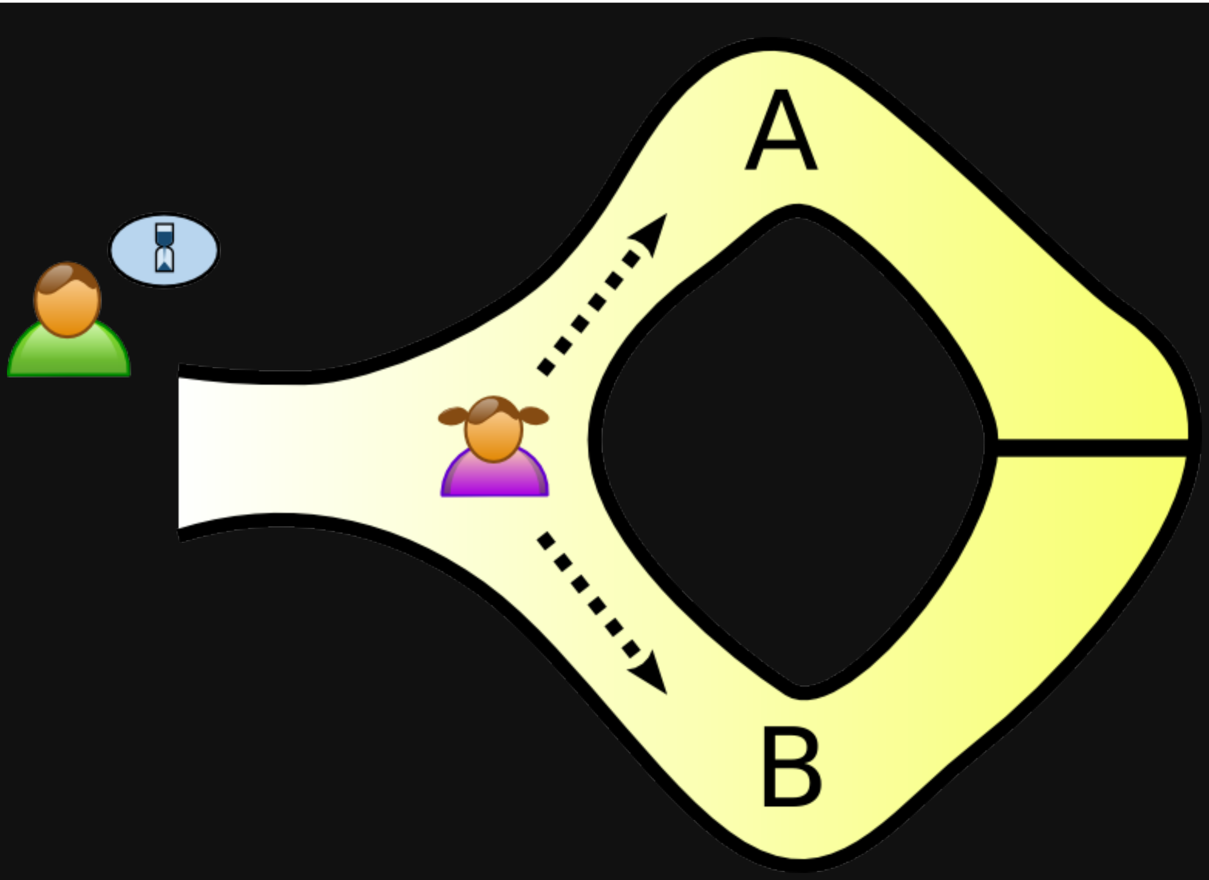
acme.com

We want to convince the server that we know the password, while keeping it ignorant of the password itself

The server is convinced, but gains zero-knowledge!

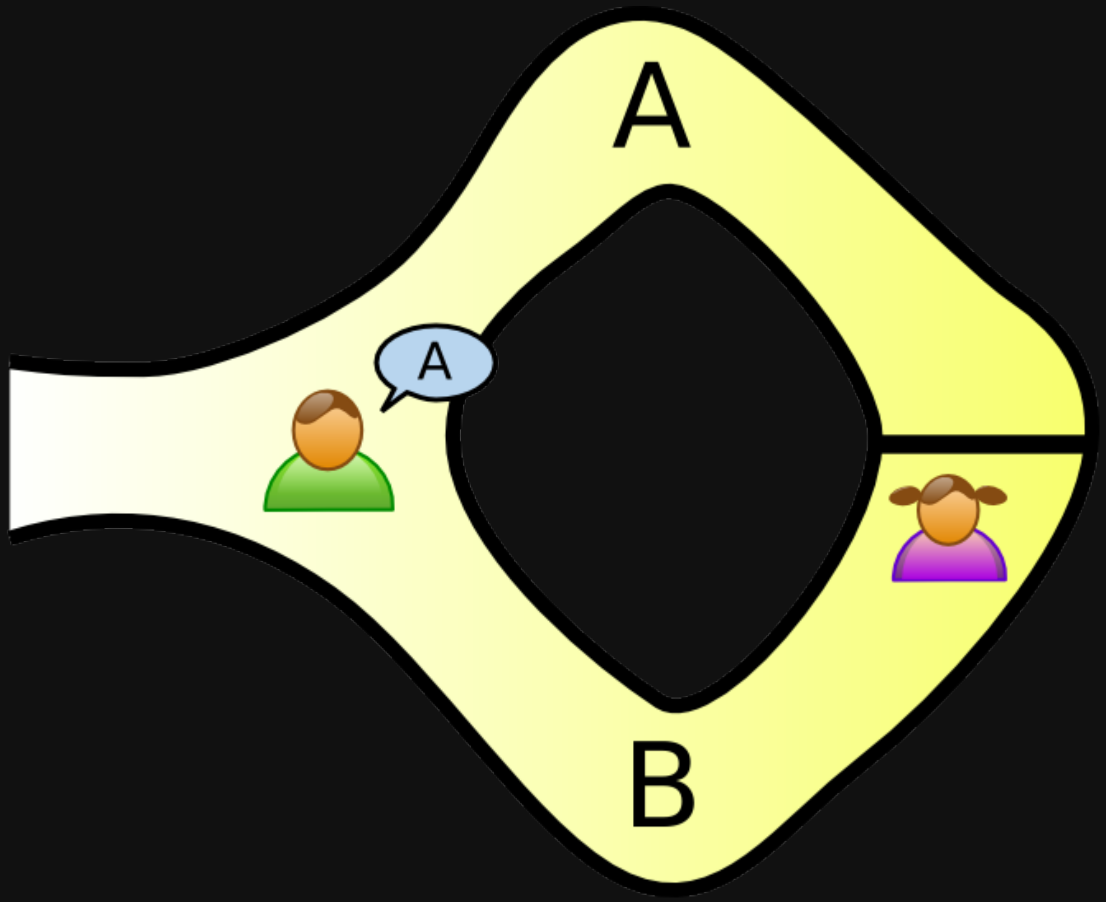
A Zero Knowledge Interactive Protocol

The “Cave” Problem



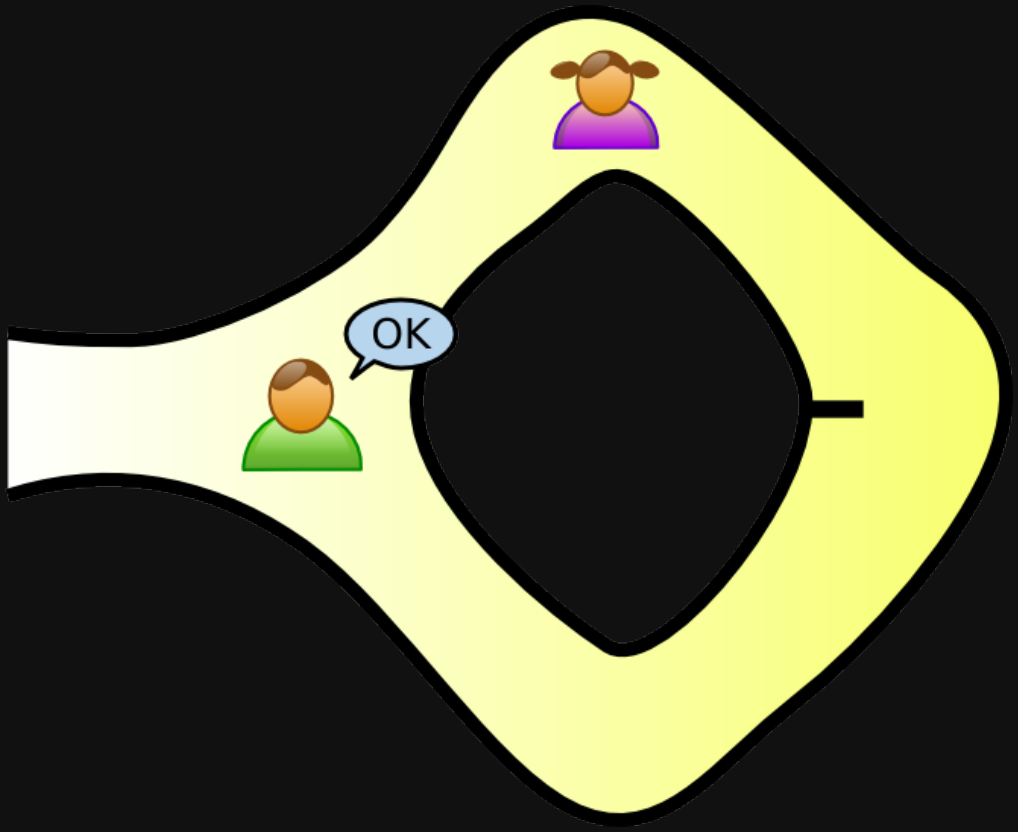
CC BY 2.5,
https://commons.wikimedia.org/w/index.php?curid=313643

The “Cave” Problem



CC BY 2.5,
https://commons.wikimedia.org/w/index.php?curid=313645

The “Cave” Problem



The “Cave” Problem

- Repeat n times (say n = 100)
- Verifier accepts only if Prover succeeds in all n iterations

Relevant property: Prover knows secret code to open door

Properties

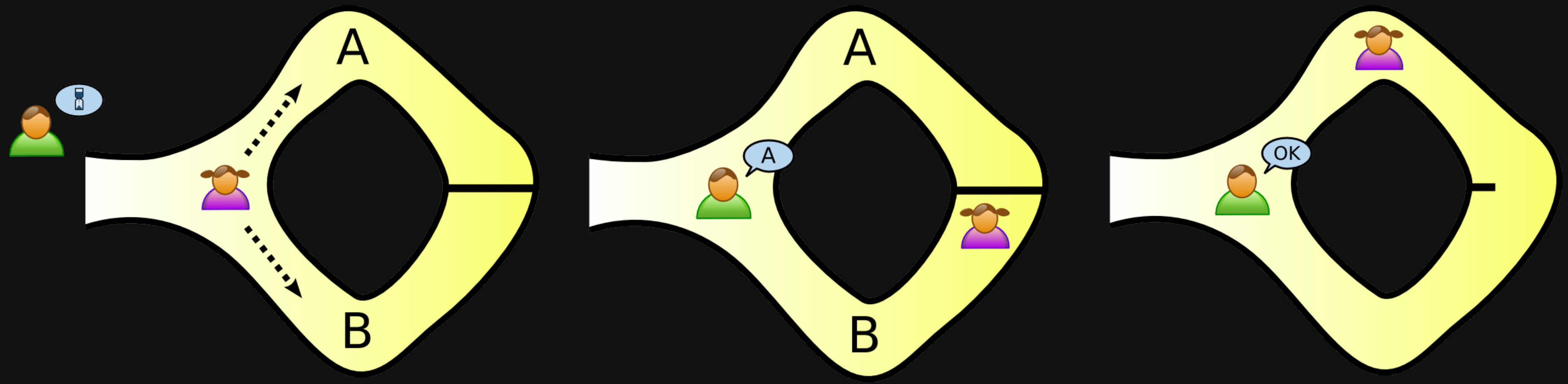
- Soundness
  - If Verifier accepts then the property holds with high probability even if Prover dishonest
- Completeness
  - If property holds then an honest Verifier always accepts proofs from an honest Prover

Zero Knowledge (Intuition)

- Verifier does not gain additional knowledge

- Verifier knows where prover will show up if she knows secret code without interacting with prover
- So Verifier does not gain knowledge when the prover shows up there

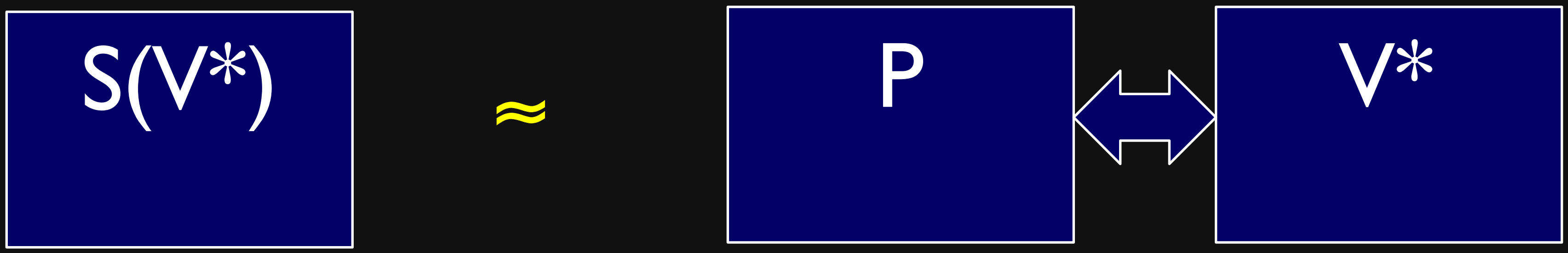
Zero Knowledge: Intuition



- Verifier’s view (or transcript): Imagine she is recording a video of the interaction
  - V flips coin and yells A or B based on that
  - Then P shows up on that side
- Probability distribution on transcript because P and V flip coins

Zero-knowledge

The verifier's view of the interaction with the prover can be efficiently simulated without interacting with the prover



S(V*)

≈

P

V*

Probability distributions on transcripts are indistinguishable

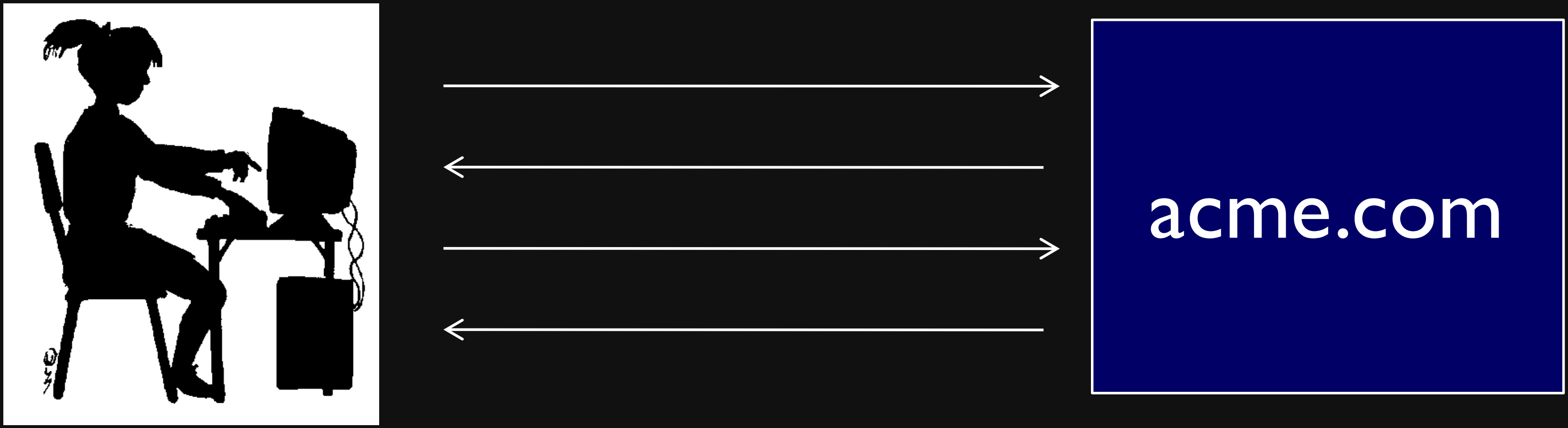
Comments

- Verifier is polynomial time

- Prover has unbounded computation power

- ZK property has to hold for all verifiers V* (not just the honest verifer V)

Zero-knowledge password authentication



acme.com

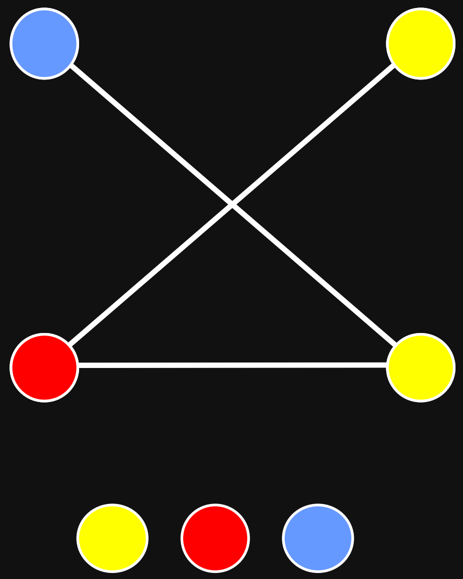
Oded Goldreich

Silvio Micali

Avi Wigderson

Graph coloring

Task: Assign one of 3 colors to the vertices so that no edge has both endpoints of same color



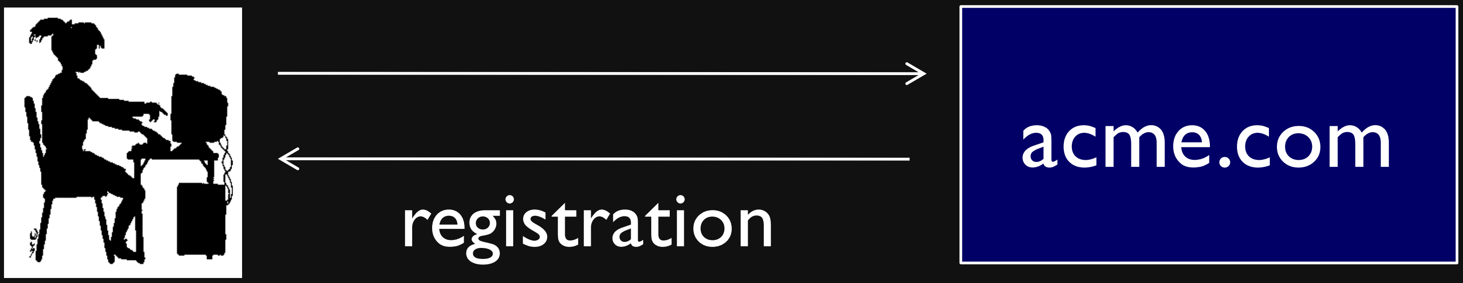
$\mathrm{3COL} = \{G\colon G$ has a valid 3-coloring$\}$

- Theorem

3COL is NP-complete

Password authentication via 3-coloring

- Step 0: When you register for the web service,

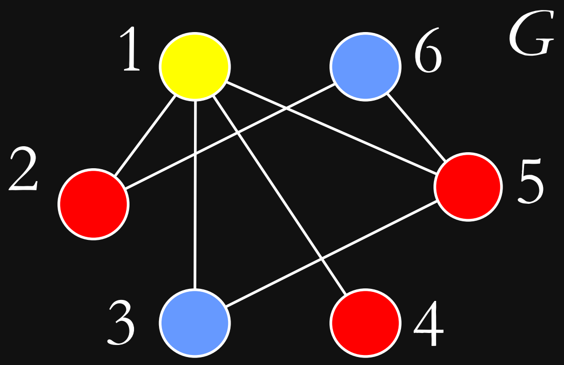


acme.com

registration

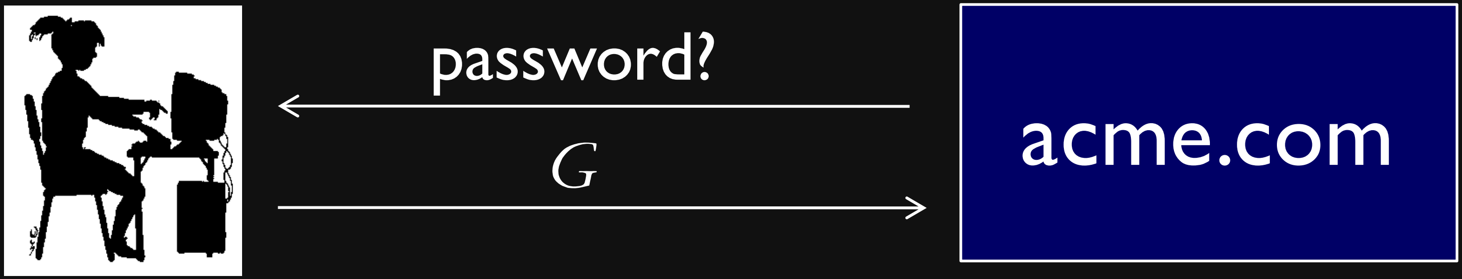
choose your password to be a valid 3-coloring of some (suitable) graph

`password:` 🟡 🔴 🔵 🔴 🔴 🔵



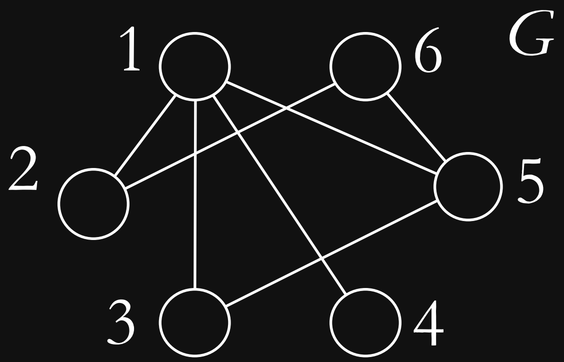
Password authentication via 3-coloring

- When the server asks for your password



do not send the password, but send the graph G instead (without the colors)

`password:` 🟡 🔴 🔵 🔴 🔴 🔵



Intuition about registration phase

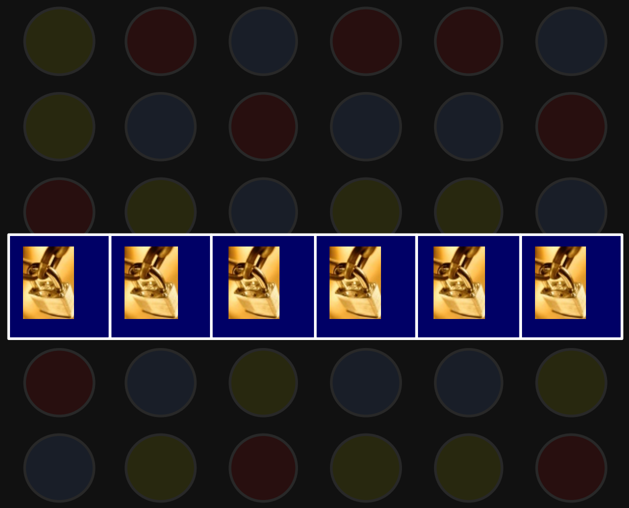
- Because 3-coloring is hard, the server will not be able to figure out your password (coloring) from G

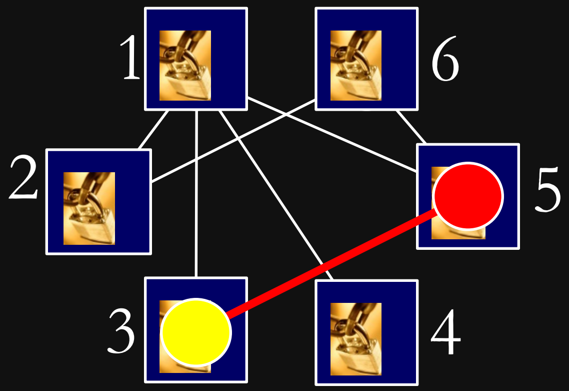
- Later, when you try to log in, you will convince the server that you know how to color G, without revealing the coloring itself

- The server will be convinced you know your password but remain ignorant about what it is

The login phase

```
password:
```





You randomly permute the colors

You lock each of the colors in a box and send the boxes to the server

The server chooses an edge at random and asks for the keys to the boxes at the endpoints

You send the requested keys

The server unlocks the two boxes and checks the colors are different

Repeat this 1000 times. Login succeeds if colors always different

Analysis in the login phase

Completeness

If you know the coloring then you will always successfully convince the server

Analysis in the login phase

Soundness

If you are an impostor, you won’t know how to color the graph, so at least one of the edges will have endpoints of the same color.

After n repetitions, the server will fail to catch this with probability

$$(1 - \frac{1}{|E|})^n$$

Analysis in the login phase

Zero Knowledge

If you are honest, the server remains ignorant about your password because all he sees are two random different colors

ZK Proof Outline for 3-COL

- Simulator S
  - Internally select random edge (i, j) and random permutation

(P,V*) interaction transcript ≈ S(V*) transcript

are not equal)

Note: If V* is not honest, use V* as a blackbox to output edge e in step 2; rewind if e not equal to (i,j)

Acknowledgment

- Some slides are (adapted) from Andrej Bogdanov

Seminal Results

- IP and ZK defined [GMR’85]
- ZK for all NP languages [GMW’86]
  - Assuming one way functions exist
- ZK for all of IP [BGGHKMR’88]
  - Everything that can be proven can be proven in ZK assuming one way functions exist