6.857 Computer and Network Security
Lecture 18

Admin:

- Quiz in-class Lecture 19
- Open notes
- Closed texts, phones, laptops
- Coverage through today's lecture

Today: ZK Proofs

- Quality control
- Sudoku
- 3-colorability
- isomorphism of graphs
- Hamiltonian path
- Discrete log

Quality control

Suppose a widget-making machine either

Ⓐ – works perfectly

Ⓑ – makes 1 out of k widgets defective (randomly) k known

on a given day. You can test widgets.

Can you tell which is case?

○ ○ ⊗ ⊗ ○ ○ ⊗ ○ ⊗ o o o X

Pick tk to test

$$\text{Prob}(\text{no defects found} \mid B) = (1 - 1/k)^{tk}$$

$$\approx (e^{-1/k})^{tk}$$

$$= e^{-t}$$

for sufficiently large t (e.g. $t = 20$) this is ≈ 0,

so you can conclude A holds. (Proper analysis needs

Bayes Rule & priors on A & B...)

TOPIC:

DATE:

FILE UNDER:

PAGE: L18,3

Sudoku

How can I
convince you
I know soln,
without
telling you
anything about soln?

						3	6	
	1			9	8		5	
		9	6		3	8		
		8	9		5	4		
		5	7		4	1		
	9		4	5			2	
	8	2						

"Zero-knowledge proof of knowledge" (ref next page)

Using cards

Using commitments

A	B	C	D	E	F	G	H	I
9	2	8	1	6	3	7	4	5

- Commit to letter for each position
- commit to table
- pick two in same row (column, or block) & test

or test table
or test known square

TOPIC:

DATE:

FILE UNDER:

PAGE: L13.4

P = prover
V = verifier

$P \longleftrightarrow V$

Interactive Proof: (of proposition x)

↖ e.g. "puzzle has soln"

Properties:

Completeness: if x true, V accepts

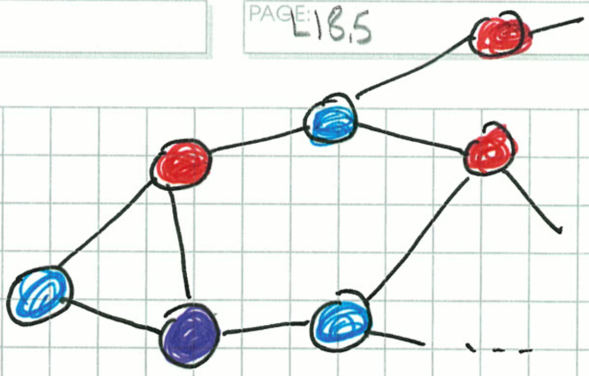
Soundness: if x false, V rejects with prob $\geq$ constant > 0.
ε

Zero-knowledge: verifier learns nothing else except whether x is true

May iterate protocol to reduce soundness error t times $\Rightarrow$ for false x, prover succeeds (verifier accepts) with probability $\leq (1-\varepsilon)^t$

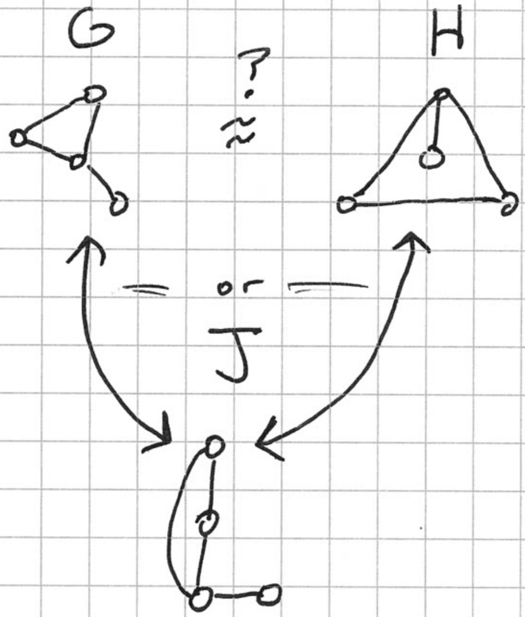
Proof of knowledge: Verifier becomes convinced that P actually knows solution

Graph 3-colorability



How can I convince you that I know 3-coloring of vertices, without telling you anything about the coloring I know?

Graph isomorphism



How can I prove to you that G & H are isomorphic,

without revealing isomorphism?

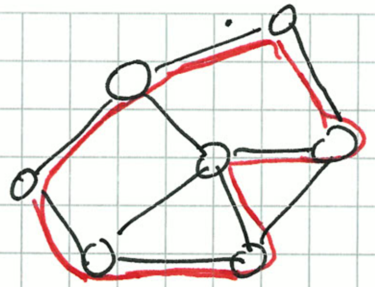
TOPIC:

DATE:

FILE UNDER:

PAGE: L18.7

Hamiltonian graph



Hamiltonian path

Discrete logarithm POK (Schnorr) (ZK)

p = large prime
q divides $p-1$, q prime

g generates subgroup $G_q = \langle g \rangle$ of order q

x = SK $\quad x \in \mathbb{Z}_q$
$y = g^x$ = PK $\quad y \in G_q$

How can Alice prove to Bob she knows x? in ZK?

A $\qquad\qquad\qquad\qquad$ B

$k \in_R \mathbb{Z}_q$
$a = g^k$ $\quad \xrightarrow[\text{"commitment"}]{a}$

$\xleftarrow[\text{"challenge"}]{c}$ $\quad c \in_R \mathbb{Z}_q$

$r = cx+k$

$\xrightarrow[\text{"response"}]{r}$ $\quad$ Check
$y^c \cdot a \overset{?}{=} g^r \pmod p$

Pf: $g^{cx} g^k \overset{?}{=} g^r$
$g^{cx+k} = g^r$ ✓

Thm: Protocol is complete.

(If Alice knows x, Bob always accepts.)

Thm. (Soundness & PoK)

$$\left[\begin{array}{l}\text{Alice can play game}\\ \Rightarrow \text{Alice ``knows'' } x\end{array}\right] \overset{\text{or}}{\equiv} \left[\begin{array}{l}\text{Alice doesn't know } x\\ \quad\Rightarrow \text{Alice can't play game}\end{array}\right]$$

Pf: Alice can play game $\triangleq$ for any a & almost all c she can produce r

Fix $a = g^k$

Suppose Alice can succeed for c & for $c' \neq c$

$$\begin{array}{l} r = cx + k \\ r' = c'x + k \\ \hline \end{array}$$

$$r - r' = (c - c') \cdot x$$

$$x = (r - r')/(c - c') \quad \therefore \text{Alice ``knows'' } x \quad ☒$$

(Note: Schnorr protocol can be turned into signature scheme by letting $c = \text{hash}(a, M)$
$\uparrow$ message

Thm: Protocol is ZK (for honest verifier)

Pf: Bob learns transcript (a, c, r). Nothing more.

Transcript is a random variable; Bob gets sample.

Bob can generate such samples on his own!

With correct distribution!

$$c \xleftarrow{R} Z_q \qquad \text{(assuming honest verifier)}$$

$$r \xleftarrow{R} Z_q \qquad \text{(r uniform in } Z_q \text{ since k is)}$$

$$a = g^r / y^c$$

$\Rightarrow$ (a, c, r) has exactly same distribution as in protocol.

∴ Bob learns nothing (except that Alice can play game)

∴ protocol is ZK. ▨

<u>Thm:</u> Any problem in NP has a ZK proof! (GMW)

NP problems have form:

$$\underset{\substack{\uparrow\\\text{true/false}\\\text{predicate}}}{f}(\underset{\substack{\uparrow\\\text{input}\\\text{instance}}}{x}) \equiv (\exists \underset{\substack{\uparrow\\\text{witness}}}{w})\, \underset{\substack{\uparrow\\\text{poly-time predicate}}}{P}(x, w)$$

I can convince you that $f(x) = \text{True}$
without showing w!

$\equiv$ Proof of knowledge of w

<u>Pf:</u> Use 3-colorability, which is NP-complete.

TOPIC:

DATE:

FILE UNDER:

PAGE: L18.13

More examples:

- My modulus has exactly <u>two</u> prime factors
- All these ciphertexts encrypt the <u>same</u> message.
- The plaintext for this message contains another message & signature on it by Bank

$$X = E(PK, (M, \sigma_B(M)))$$

- I know w s.t. $x = hash(w)$ (pre-image)

Extensions:

- Non-interactive ZK. (NIZK)

use Fiat-Shamir heuristic:

$$\text{challenge} = \text{hash}(\text{commitment} \,\|\, \text{statement to be proved})$$

so <u>Prover</u> can derive challenge & write it all down