TOPIC: 6.857

DATE: 4/11/16

FILE UNDER:

PAGE: 18.1

Admin:

Quiz review tonight 7-9pm in

Quiz Wed in-class (notes only)

Pset #5 out later today

Today:

Zero-knowledge proofs (& proofs of knowledge)

- Interactive proofs & protocols: completeness, soundness, ZK
- statistics/commitments
- Sudoku
- Graph 3-colorability
- Graph isomorphism
- Hamiltonian cycle
- Discrete log
- Any problem in NP has a ZK proof

Reading:

Goldreich. Foundations of Cryptography: Basic Tools (2001) Chapter 4.

TOPIC:

DATE:

FILE UNDER:

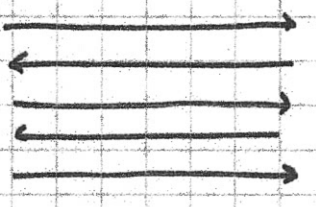
PAGE:

Interactive Protocol: (two-party) or Interactive Proof

Common input x (statement to be proved)

P (prover) V (verifier)

w



→ True/False
~ Accept/Reject

$(P, V)(x)$ = True/False

P may be powerful
V is poly-time

x typically NP statement: $(\exists w)\, P(y, w)$

↖ poly-time predicate
↑ poly-size witness

Example: $(\exists w)\; y = g^w \pmod p$

P doesn't want to reveal w ! wants to reveal zero about w.

P = prover
V = verifier

P ⟷ 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 ≥ constant > 0.
ε

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

May iterate protocol to reduce soundness error

t times ⇒ for false x, prover succeeds (verifier accepts)
with probability ≤ $(1-\varepsilon)^t$

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

Quality control

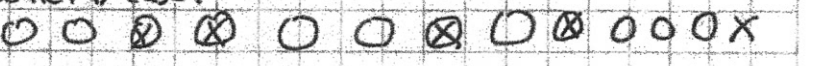
Suppose a widget-making machine either

(A) – works perfectly

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

on a given day. You can test widgets.

Can you tell which is case?



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:

Commitments

$c = \text{commit}(v, r)$ — commit to v using randomness r

$\text{open}(c) \rightarrow (v, r)$ — reveal or open commitment

hiding: seeing c gives no information about v

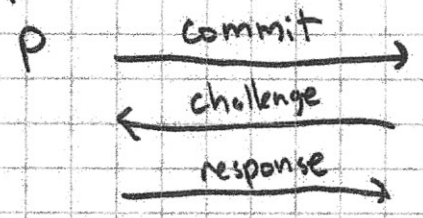
binding: c can only be opened one way (i.e. to one v)

e.g. Pedersen commitment $c = g^v h^r$ — g, h generators, r random

perfect hiding
computationally binding (DLP assumed hard)

- Allows prover to commit to everything he knows, but to only reveal some portion chosen (randomly) by verifier ("cut & choose")
- Verifier can check portion opened.

Typical ZK proof structure:



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"

Using cards

Using commitments

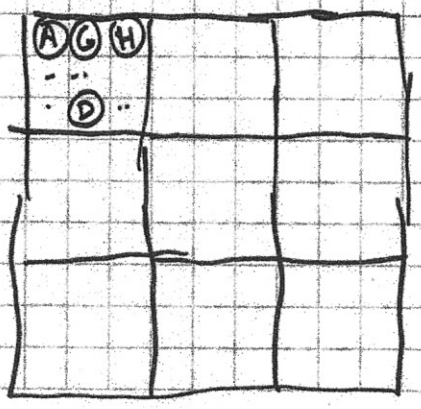
A B C D E F G H I
9 2 8 1 6 3 7 4 5

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

(A) = Commitment to A

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

Grid



Complete ✓
sound ✓
ZK ✓

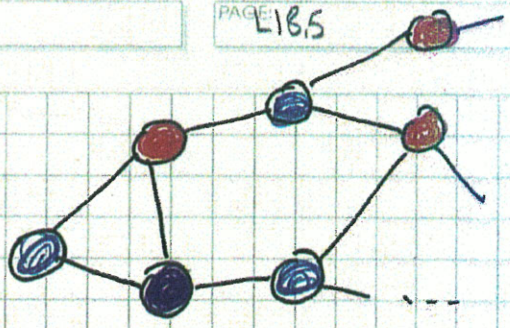
TOPIC:

DATE:

FILE UNDER:

PAGE: L18.5

Graph 3-colorability



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

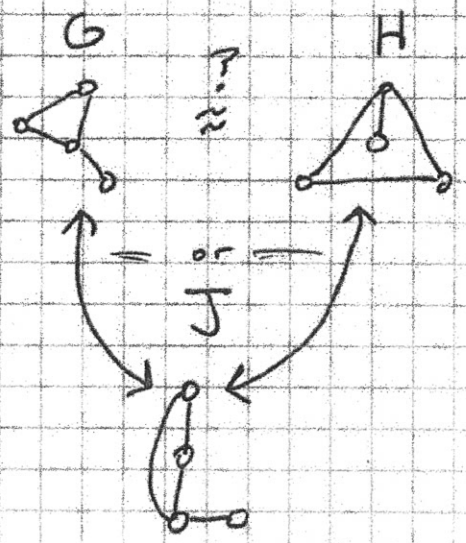
TOPIC:

DATE:

FILE UNDER:

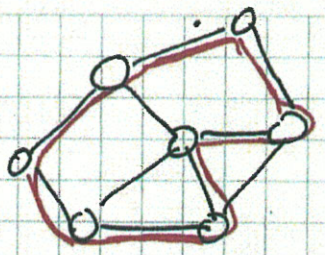
PAGE L18, 6

Graph isomorphism



How can I prove to you that G & H are isomorphic, without revealing isomorphism?

Hamiltonian graph

G = 

Hamiltonian path

commit to random isomorphic copy M
ham path in copy

verifier asks for: proof that G ≅ M
OR
ham path in M

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

$x = SK \quad x \in Z_q$
$y = g^x = PK \quad y \in G_q$

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

A		B
$k \in_R Z_q$		
$a = g^k$	$\xrightarrow[\text{"commitment"}]{a}$	
	$\xleftarrow[\text{"challenge"}]{c}$	$c \in_R Z_q$
$r = cx + k$	$\xrightarrow[\text{"response"}]{r}$	check $y^c \cdot a \stackrel{?}{=} g^r \pmod p$

Pf: $g^{cx} g^k \stackrel{?}{=} g^r$

$g^{cx+k} = g^r$ ✓

Thm: Protocol is complete.

(If Alice knows x, Bob always accepts.)

Thm. (Soundness & POK)

Alice can play game ⇒ Alice "knows" x] $\equiv$ [Alice doesn't know x ⇒ Alice can't play game

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 $c' \neq c$

$$r = cx + k$$
$$r' = c'x + k$$

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

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

(Note: Schnorr protocol can be turned into signature scheme by letting $c = \text{hash}(a, M)$, M ← 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 <u>on his own</u>!

With correct distribution!

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

$r \xleftarrow{R} Z_q$ (r uniform in Z_q since $k \in \mathbb{N}$)

$a = g^r / y^c$

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

$\therefore$ Bob learns nothing (except that Alice can play game)

$\therefore$ protocol is ZK. ▨

Thm: Any problem in NP has a ZK proof! (GMW)

NP problems have form:

$$f(x) \equiv (\exists w)\, P(x, w)$$

- $f(x)$: true/false predicate
- x: input instance
- w: witness
- $P(x, w)$: poly-time predicate

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

≡ Proof of knowledge of w

Pf: Use 3-colorability, which is NP-complete.

More examples:

- My modulus has exactly two prime factors
- All these ciphertexts encrypt the same 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:

challenge = hash (commitment || statement to be proved)

so Prover can derive challenge & write it all down