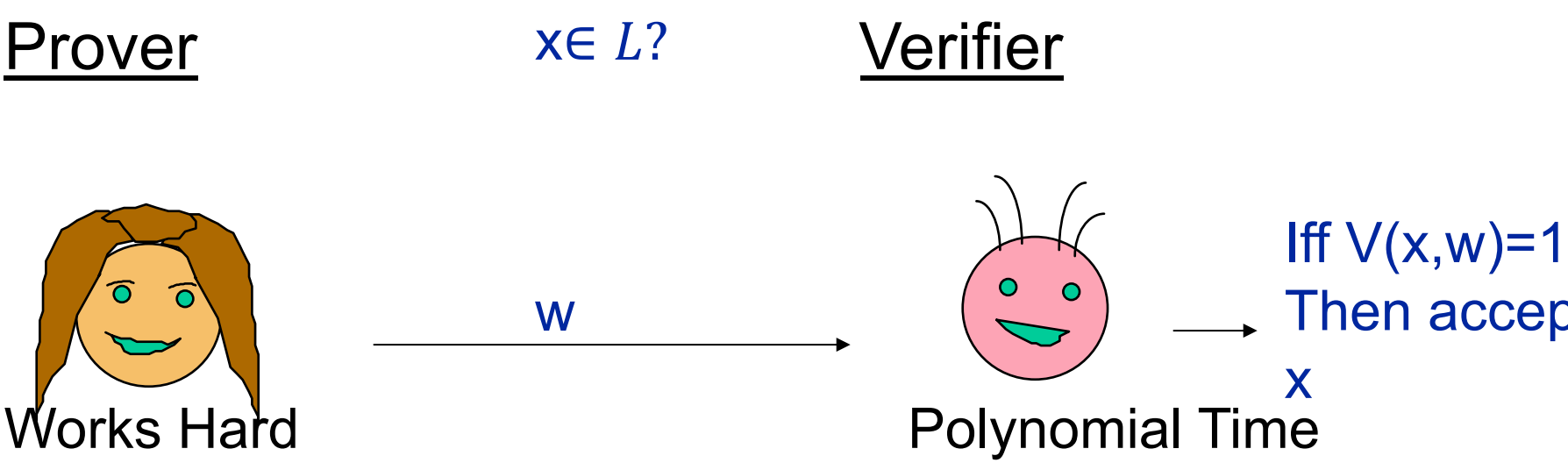
Lecture 15

Zero Knowledge Ii

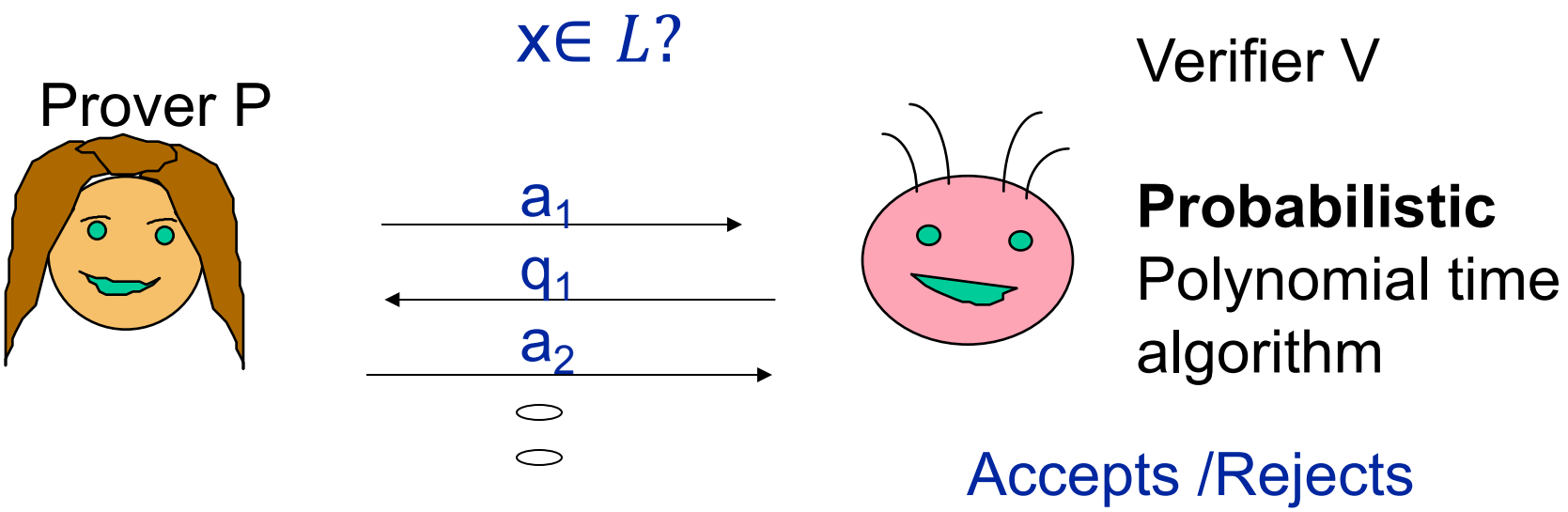
Class NP



NP={D s.t. ∃polynomial time V s.t
x∈ D iff ∃w of polynomial size s.t. V(x,w)=1}

Any statement which have
Efficiently Verifiable Classical Proofs

Class IP



IP = {L s.t. ∃ (P,V) s.t.
if x ∈ L, then prob((P,V)[x] =accepts] ≥2/3
if x ∉ L, then prob((P,V)[x]=accepts]<1/3

Perfect Zero-Knowledge

For a given P and V on input x, define probability space $View_{(P,V)}(x)= \{(q_1,a_1,q_2,a_2,\ldots,coins)\}$ (over coins of V and P)

(P,V) is **honest** verifier perfect zero-knowledge for L if:

∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V)}(x) = SIM(x)$

(P,V) is perfect zero-knowledge for L if : **∀PPT V***

∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V^*)}(x) = SIM(x)$

Statistical Zero-Knowledge

(P,V) is statistical zero-knowledge for L if : $\forall V^*$

$\exists$SIM expected polynomial time randomized algorithm s.t.$\forall x \in L$

$| \sum_v |prob[v \in View_{(P,V^*)}(x) - prob[v \in SIM(x)]| < neg(|x|)$

Today

- Computational Zero Knowledge
- Every problem in NP has a Computational Zero Knowledge Interactive Proofs
- Is IP greater than NP?
  - Today: examples unknown to be in NP
  - Complexity class IP=PSPACE
- Applications

Computational Zero-Knowledge

(P,V) is **honest** verifier perfect zero-knowledge for L if:

∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V)}(x) \approx_c SIM(x)$

Relax to "indistinguishable" by any observer who runs In probabilistic polynomial time

(P,V) is computational zero-knowledge for L if : **∀PPT V***

∃SIM an expected polynomial time randomized algorithm s.t. ∀x in L, $View_{(P,V^*)}(x) \approx_c SIM(x)$

Notation: $View_{V^*}(x) \approx_c SIM(x)$

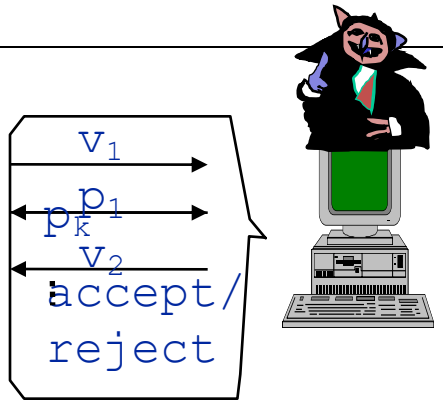
Prover Gives **Perfect** Zero Knowledge

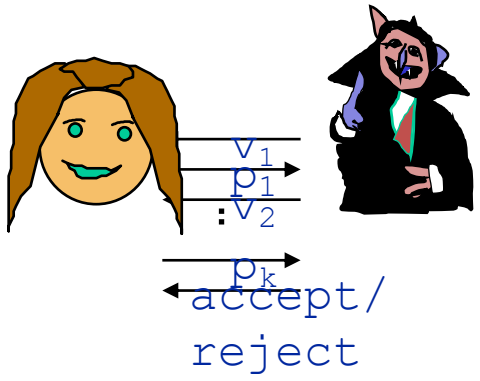
- If: we can efficiently simulate the view of any verifier s.t. `Simulated views' `real verifier" are indistinguishable by any PPT distinguisher

??

SIM

REAL





The observer

Any Probabilistic Poly Time Algorithm

Zero Knowledge for all of NP

Theorem: If one-way permutations exist, then every problem in NP has a computational zero knowledge interactive proofs

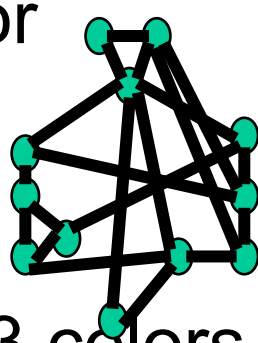
- The assumption can be relaxed to one-way **functions**

  Building Block: One Way Functions imply Commitments schemes

- To prove the theorem, should we construct ZK proof for every **NP** language? Not efficient!

How can you prove something so general?

Idea: Show a zero knowledge interactive proof for Complete Problem for NP.



3COLOR = all graphs which can be colored with 3 colors s.t for for all edges (u,v) color(u) ≠color(v)

NP Completeness [Cook-Levin-Karp]: Given L in NP.

Instances x is polynomial time reducible to G_x

$x \in L \Rightarrow G_x$ is 3 colorable

$x \notin L \Rightarrow G_x$ is not 3 colorable

Show a Zero-knowledge Proof for 3-coloring

Physical Intuition for Protocol

On common input graph G =(V,E) and

Provers private input coloring $\pi: V \longrightarrow \{0,1,2\}$

- P picks a random permutation σ of the coloring π & color the graph with coloring $\alpha=\sigma(\pi)$. It hides the color $\alpha(u)$ of each vertex inside a locked box

- V Select a random edge (u,v)
- P opens boxes corresponding to u and v

- V accepts if and only if $\alpha(u) \neq \alpha(v)$

  [colors are different]

Intuition for Completeness and Soundness

- Completeness: if prover uses a proper 3-coloring, the verifier will accept.

- Soundness: Let $k = |E|^2$

If G is not 3-colroable, then for all P*

Prob[(P*,V)(G) accepts]<1- 1/|E|

Repeat k times.

Soundness Prob[(P*,V)(G) accepts]<

$(1-1/|E|)^{k.} < 1/e^{|E|}$

From Intuition to a Proof

To “digitze” the above proof, we need to implement locked boxes

Need two properties from digital locked boxes:

- **Hiding**: *V* should not be able to see the content inside a locked box

- **Binding**: *P* should not be able to modify the content inside a box once its locked

Commitment Scheme
(Digital analogue of locked boxes)

- An efficient two-stage protocol between a sender S and receiver R on input (1^k) s.t.:

- commit stage: S has private input $b \in \{0, 1\}$;

  At the end of the commit stage
  - both parties hold output com (called the commitment)
  - S holds a private output dec (called the de-commitment)

- reveal stage: S sends the pair (dec , b) to R.

  R accepts or rejects

Properties of a Commitment Scheme

Completeness: R always accepts in an honest execution of S.

Hiding: $\forall R^*$, $b \neq b' \in \{0,1\}$, In commit stage

$\{View(S(b), R^*)(1^k)\} \approx_c \{View(S(b'), R^*)(1^k)\}$.

Binding: Let *com* be output of commit stage $\forall S^*$ Prob[S^* can reveal two pairs (dec,b) & (dec',b') s.t. R(com, dec, b) = R(com, dec′, b′) =Accept]<neg(k)

Ex: c $\in$Enc(r,b) for semantically secure PK enc.
Comm=c, Dec={r,b}

Commitment Schemes: Remarks

The previous definition only guarantees hiding for one bit and one commitment

Claim: One-bit commitment implies multiple string commitment (using hybrid argument as in encryption)

Commitment Schemes

Can be implemented using interactive protocols, but we will consider non-interactive case. Both commit and reveal phases will consist of single messages

One- Way function based commitments require 2 rounds of interaction in commit stage

Construction of Bit Commitments

Construction: Let f be a OWP, B be the hard core predicate for f

Commit phase(b): Sender chooses r, sends Comm = $f(r), b \oplus B(r)$

Reveal phase: Sender reveals (b, r). Receiver accepts
If Comm= $(f(r), b \oplus B(r))$, and rejects otherwise

Security:

Binding follows from construction since f is a permutation

Hiding follows in the same manner as IND-CPA security

ZK interactive proof for G3COL

On common input graph $G=(V,E)$ and private prover input coloring $\pi: V \rightarrow \{0,1,2\}$

- P $\rightarrow$ V: Pick a random permutation σ of the coloring & color the graph with coloring $\alpha(\pi)=\sigma(\pi(v))$. Send commitments $Enc(r_v, \alpha(v))$ $\forall$ vertex v.
- V $\rightarrow$ P: Select a random edge (u,v) and send it
- P $\rightarrow$ V: reveal colors of u and v committed in $Enc(r_u,\alpha(u))$ and $Enc(r_v,\alpha(v))$ by releasing r_u and r_v
- If $\alpha(v)) \neq \alpha(u)$ V rejects, otherwise repeat and V accepts after k iterations.

Honest Verifier Computational ZK

Simulator S in input G=(V,E) : guess in advance the challenge (a,b) of the honest verifier V.

- Choose random edge (a,b) in G
- Choose a_a, a_b in {0,1,2} s.t $a_a \neq a_b$ at random and for all v ≠a,b set $a_a = 2$.
- Output SIM =

$$(\mathrm{Enc}(r_v, a_v), (a, b), r_a, r_b)$$

Claim: SIM$\approx_c$ View$_{(P,V)}$(G)

Computational ZK: Simulation for any Verifier V*

Simulator SIM on input G and verifier V*:

Fix random tape ω for V*

For i = 1 to $|E|^2$:

- Choose random edge (a, b) and generate vector com = Enc(r_v, a_v) as in honest verifier simulation.
- Run V*(com; ω) to obtain challenge ($a*$, $b*$); if ($a*$, $b*$) = (a, b), then output transcript as in honest verifier case , transcript=Enc(r_v, a_v), (a, b), r_a, r_b)

If all iterations fail, output $\perp$.

Theorem: If Enc is semantically secure with respect to non-uniform adversaries, then

Claim 1: $\forall G,\pi$ (a true coloring) : prob[$\perp$ output]=neg(|E|)

Claim 2 :if $\perp$ is not output, then simulated-view $\approx_c$real-view

Simulation for any Verifier V*

Claim 1 : $\forall G,\pi$ (a true coloring) : prob[$\perp$ output]=neg(|E|)

Proof: By Hybrid argument.

Hybrid 1 (*G*):Fix random tape ω for V*

For i = 1 to $|E|^2$:

1. Choose random edge (a, b)
2. Let com=vector of encrypted colored vertices s.t all vertices v are colored by α(v) each with randomness ra(v) [as prover does in real protocol] .
3. Run V*(com)=(a*,b*). If (a*,b*)=(a,b),

   output transcript (com, (a*,b*), r_a, r_b)

   If all iterations fail, output $\perp$.

Lemma1: Hybrid 1 and *View* $_{(P,V^*)}$ (*G*) are statistically close (chance of $\perp$ is negligible)

Lemma2: Hybrid 1 and SIM(G,V*) are computationally indistinguishable if Enc is semantically secure

Examples of NP-assertions

- graph G is 3-colorable

- graph G has a traveling salesman tour of cost C,

…

- NP=Given encrypted inputs E(x) and program PROG, y=PROG(x)….

Many, Many Applications:

•Can prove properties about m without ever revealing m, only E(m)

• Can prove relationships between m1 and m2 never revealing either one, only E(m1) and E(m2).

For example: $L = \{(C_1, C_2)$: *there exists* r_1, r_2, M *s.t.* $C_1 = E_1(r_1, M)$ *and* $C_2 = E_2(r_2, M)\}$ is in NP

Generally: A tool to enforce honest behavior without forcing to reveal information.

General Cryptographic Importance

- Proving correctness of protocols is complex even if users are honest; If users deviate from protocol in arbitrary ways, almost impossible in a case-by-case manner, need tools and framework to prove correctness.
- Proof of proper behavior is fundamental tool for design of secure protocols

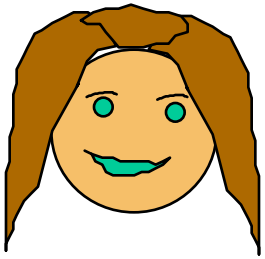
- Zero Knowledge Proofs enable automatic conversion of any protocol proven secure against honest-but-curious adversaries to protocol secure against deviating adversaries

Today

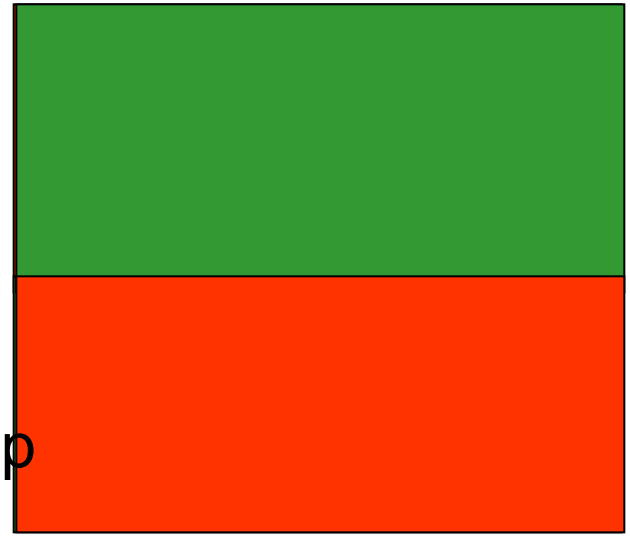
- Computational Zero Knowledge
- Every problem in NP has a Computational Zero Knowledge Interactive Proofs
- Is IP greater than NP?
  - Today: examples unknown to be in NP
  - Complexity class IP=PSPACE
- Applications

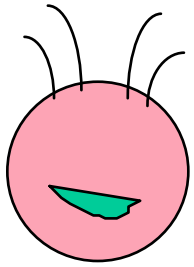
Zero Knowledge Proof

Prove to color blind bob that colors exist



Claim: there are 2 colors on this page, red top





Bob color blind

Bob tosses coin

If Heads keep red on top.

Tails flip to green on top.

Bob sends resulting page

Alice names color on top

Bob send resulting page

Alice names color on top

if Alice is wrong, Reject the claim

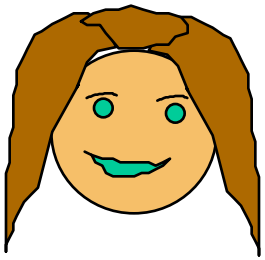
Do it Again

Chance that Alice lucky twice = ¼

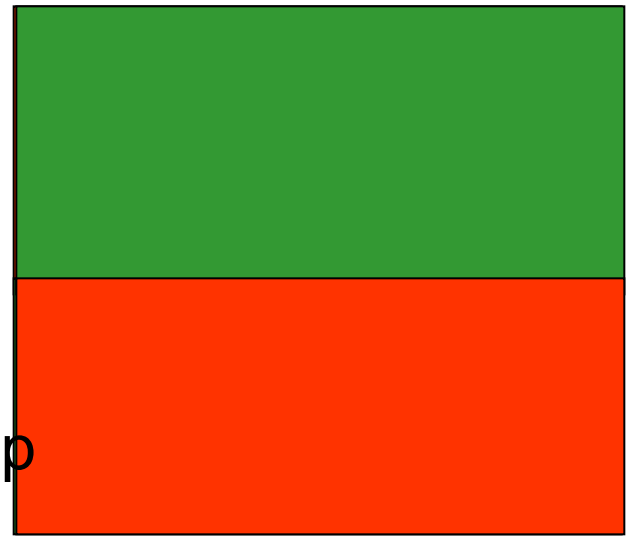
If Alice is correct ½ chance that its just luck

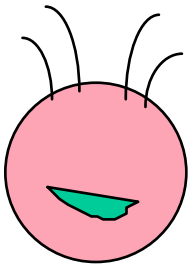
Zero Knowledge Proof

Prove to color blind bob that colors exist



Claim: there are 2 colors on this page, red top





Bob color blind

Bob tosses coin

If Heads keep red on top.

Tails flip to green on top

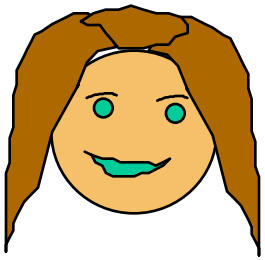
Completeness: if there are 2 colors, Bob will always accept

Soundness: if there is only 1 color, then

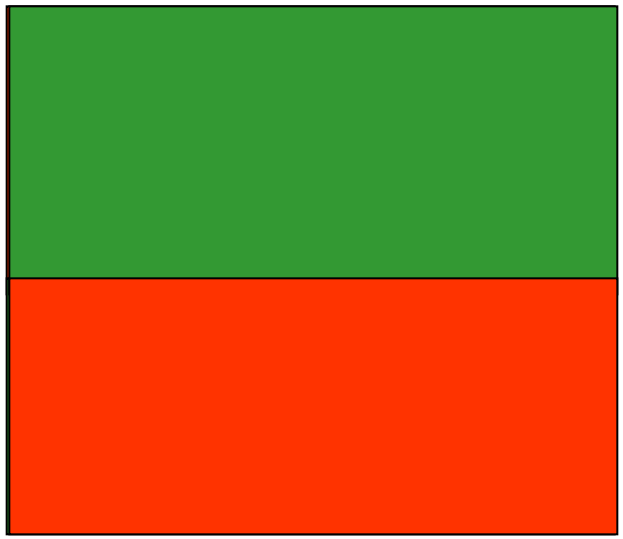
Probability that after 100 iterations Bob will reject > $1-1/2^{100}$

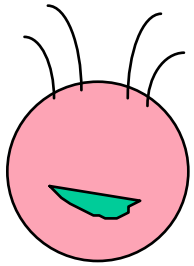
Zero Knowledge Proof

Prove to color blind bob that colors exist



Claims there are 2 colors on this page





Bob color blind

Bob tosses coin to decide if to flip page:
Heads keep red on top.
Tails flip to green on top.

Bob sends resulting page

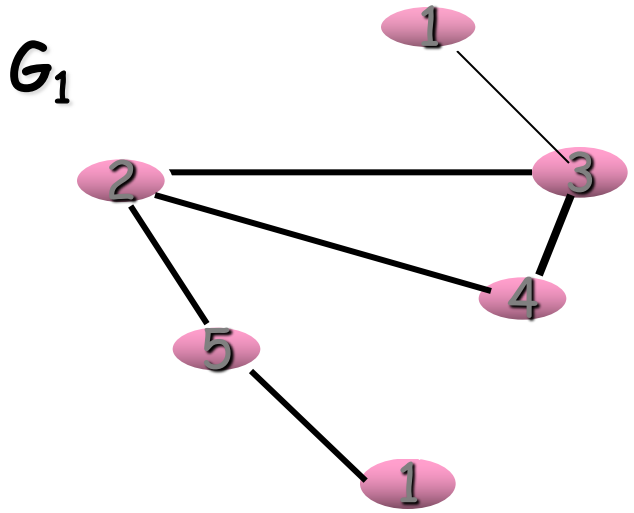
Alice names color on top=$coin_1$

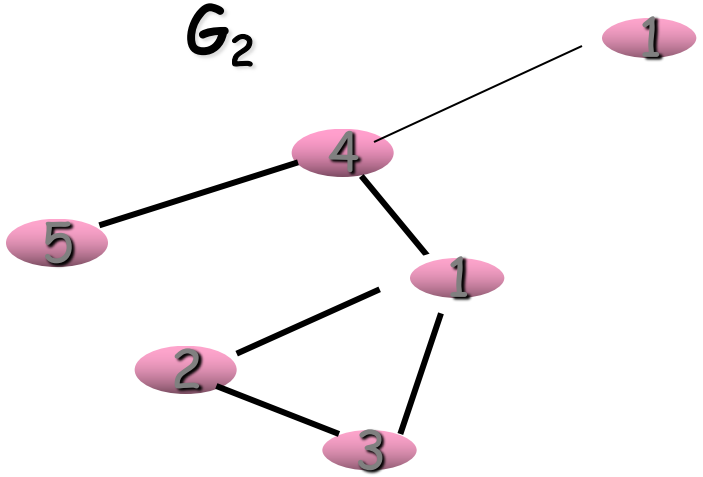
Bob send resulting page

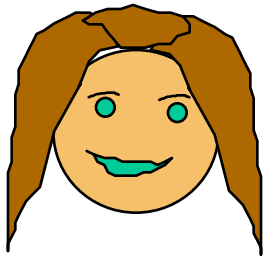
Alice names color on top=$coin_2$

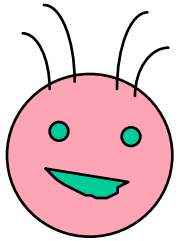
View of Bob= {(Page, coin)}

Example: G_1 is NOT isomorphic to G_2



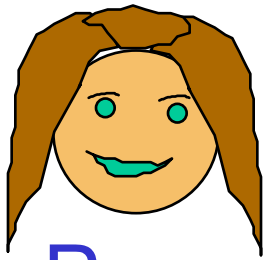






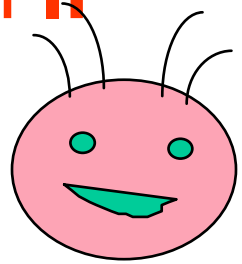
Shortest classical proof:
≈exponential n!
But can convince with an efficient interactive proof

Graph Non-Isomorphism (Non-ISO) in **IP**



Prover

input: (G_0, G_1)



Verifier

$H = \gamma(G_c)$

if H isomorphic

to G_0
then b = 0, else
b = 1

b

flip coin c $\in$ {0,1}; pick random γ

Output ACCEPT

iff b = c

Completeness: if $(G_0\ G_1) \in Non - ISO$, *then*
$Prob[(P,V)[(G_0\ G_1)]=accept]=1$

Soundness: if $(G_0\ G_1) \in ISO$,
$Prob[(P,V)[(G_0\ G_1)]=accept] \leqq 1/2$

GNI Interactive Proof

- completeness:
  - if G_0 not isomorphic to G_1 then H is isomorphic to exactly one of (G_0, G_1)
  - prover will choose correct b=c
- soundness:

  if G_0 is isomorphic to G_1 then prover sees same distribution on H for c = 0, c = 1 which no information on c $\Rightarrow$

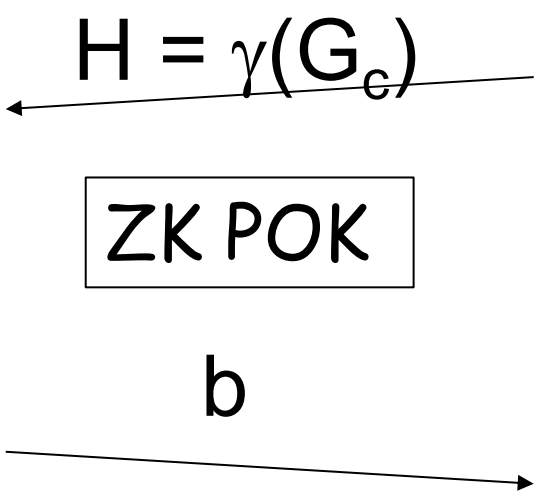
  Prob[prover P* outputs b=c]<=1/2

Honest Verifier Zero Knowledge

This is obviously honest verifier zero-knowledge (when the graphs are isomorphic):
--All the verifier gets is the coin c he tossed.

But, is it zero-knowledge for all verifiers?
-No. V can use P to find out if H is isomorphic to G_0 or isomorphic to G_1.

-Instead, the Verifier proves in ZK that he knows γ s.t either $H=\gamma(G_0)$ or $H=\gamma(G_1)$



Applications

✓ Preventing Identity Theft

- Secure Protocols

- Proving properties of secrets:
  - Commit + prove

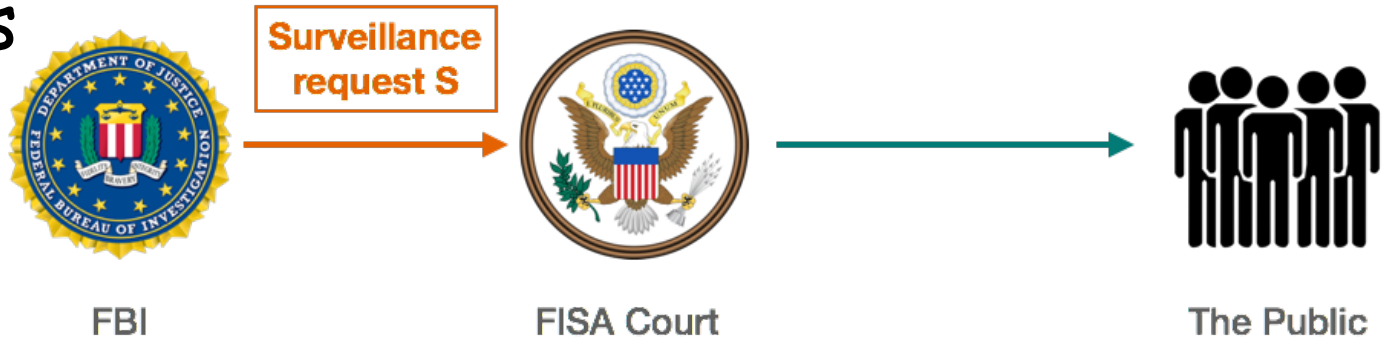
Recent Uses of Zero Knowledge Proofs

2014 Zero Knowledge and Nuclear Disarmament: projects at Princeton and MIT [Barak etal]

2015 Zero Knowledge and Forensics [Naor etal]

2016 Zero Cash, crypto currency which protects the privacy of transactions [BenSasson,Chiesa, Tromer etal]

2017 Proof of "compliance" of FISA with secret laws



Recent Uses of Zero Knowledge Proofs

2014 Ze…ment: projects

2015 Ze…]

Zero Co… 2016 the priv… Tromer eta…

2017 Pr… ecret laws

ZERO KNOWLEDGE PROOF STANDARDIZATION

An Open Industry / Academic Initiative

HOME | INTRODUCTION | 1ST WORKSHOP | STANDARDS DOCUMENTS

The 1st ZKProof Standards Workshop
10-11th May, 2018

ZKProof

Zero Knowledge Proofs are a cutting edge cryptographic tool that is starting to see adoption. This breakthrough technology forms the basis of several cryptographic applications, improving the trade-offs between data privacy and integrity. Zero Knowledge Proofs allow a prover to convince a verifier that some computational statement is correct without revealing any information except the veracity of the statement.

ZKProof.org is an open initiative of industry and academia to standardize the use of zero knowledge proofs. We are planning several workshops to standardize the security, implementation, applications and all other related aspects of this technology. The first workshop will take place in Boston in mid May and will bring together for the first time academic and industry experts in the field.

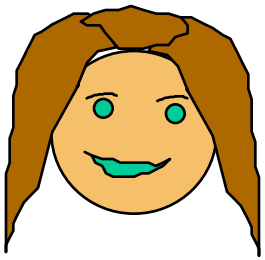
…Members:

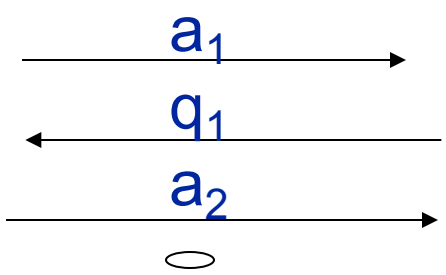
ZK Arguments

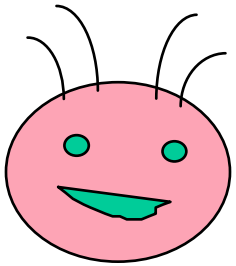
Prover P
PPT

$x \in L?$

Verifier V







Probabilistic Polynomial time algorithm

Completeness: there exists P Accepts /Rejects
if $x \in L$, then prob((P,V)[x] =accepts] $\geq 2/3$

Soundness': if $x \notin L$, then for all probabilistic polynomial time provers P* Prob((P,V)[x]=accepts]<1/3

Theorem: Perfect ZK Arguments exist for all NP if one way functions exist [OWF used for soundness']

Basic Questions about Zero Knowledge(I)

- Q1: Sequential Compositions
- Q2: Parallel Compositions?
  - Not always (artificial counter example)
  - Known natural examples cannot be proved using black box simulation
  - A: Weaken definition of ZK to Witness Hiding [FeSh87]