Zero Knowledge - May 3

Logistics

- PS2 Out now, due next week in class
- Project milestone due in 2 weeks (May 17)
  - * 1-2 pages USE TEMPLATE
  - * First half of Project
- Return PS1.

Today:

- So far we have covered crypto primitives
  - * PRP, PRF, DH, RSA
  - ↳ Essentially just simple algorithms
- In real world, lots of interaction
  - * What does it mean for an interactive protocol to be secure?
  - * How do we prove this?
- Zero Knowledge
  - * One of my favorite ideas of all time: { prove something to you w/o leaking extra info }
  - [* rejected three times before published (!)]
  - * Idea is counter-intuitive
    - ↳ But powerful! So useful in many crypto protocols
  - * Shows the importance of definitions
    - ↳ The original ZK paper is not important b/c of construction but b/c of defn of IP, ZK!
    - ↳ Defn is > ½ the battle.
  - * Foundational paper in "theoretical" crypto

→ What is possible in theory
"polynomial time"
(complexity theory view)

→ What runs on my computer
(practitioners)

Review: Elliptic Curves & Pairings

Why EC?

RSA: $\Omega(\lambda^3)$ bit keys

ECC: 2λ bit key.

B/c Factoring is easier than EC hard problems.

Idea: We can do DH in an abstract group $(G, *)$ — G: elements, $*$: op

g ← group element

$g^2 = \underbrace{g * g}_{op}$

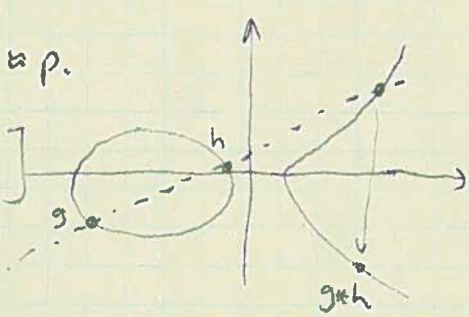
EC Group: $(x, y) \in \mathbb{F}_p$ $\quad p \approx 2^{256}$

s.t. $y^2 = x^3 + Ax + B \in \mathbb{F}_p$.

(Also "point at infinity" = identity element)

q points on curve, $q \approx p$.

[can construct EC group to have order (prime) q]



Hard Problems

Dlog $\quad a \xleftarrow{R} \mathbb{Z}_q$

Given (g, g^a), produce a

CDH $\quad a, b \xleftarrow{R} \mathbb{Z}_q$

Given (g, g^a, g^b) produce g^{ab}

DDH $\quad a, b, c \xleftarrow{R} \mathbb{Z}_q$

Distinguish (g, g^a, g^b, g^{ab}) from (g, g^a, g^b, g^c)

Review: EC & Pairings

→ DDH makes life easy!

Saw PRG and PRF (Naor-Reingold) from DDH

↳ Hard to get directly from Dlog, CDH

Pairing

— Defined over special EC groups

$e: \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$

s.t. 1) eff computable

2) Bilinear $\forall a, b \in \mathbb{Z}_q, g \in \mathbb{G}$

$$e(g^a, g^b) = e(g,g)^{ab}$$

3) Non-degenerate $e(g,g)$ generates $\mathbb{G}_T$ if g generates $\mathbb{G}$

→ Many new crazy assumptions on "pairing-equipped groups"

→ DDH is easy in $\mathbb{G}$,
CDH in $\mathbb{G}$ is hard (it seems)

With pairings

- IBE - encrypt to a string (not a pk)
- 3-way DH key agreement (non-interactive)
- Short signatures
- Attribute-based encryption (encrypt to policy/attributes)
- SNARKs

⋮

Sparked a new wave of crypto research in 2000s

↳ All implementable!!!

Interactive Proofs

Defined by Goldwasser, Micali, Rackoff (also Babai)

→ We are going to be talking about **proofs**

Goal of a proof: convince someone ("verifier") that a statement is true.

First, let's define statements/truth more formally

$$\text{“}\underset{\text{instance}}{x} \in \underset{\text{language}}{L}\text{”}$$

For us, this is a statement to be proved

Examples "N is the product of two 1024-bit primes"

$\underset{\text{instance}}{N} \in \underset{\text{language}}{\{pq \mid p,q \text{ are 1024-bit primes}\}}$

"(g,h) is s.t. $h = g^a$ and a is odd"

$(g,h) \in \{(g,g^a) \mid a \in \mathbb{Z}_q \text{ and } a \text{ is odd}\}$

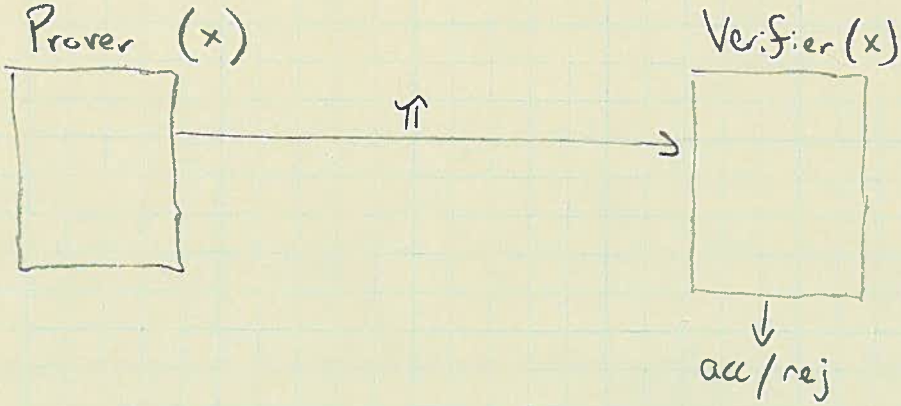
"The Pythagorean Thm is true."

$\boxed{\text{PYTHM}} \in \{\text{true statements in some logical system}\}$

"$f(x) = [(x+3)^2 + (x-2)^{100}]^{50} + 27$ has 15 distinct roots mod p"

$f \in \{\text{polys mod } p \text{ with 15 distinct roots}\}$

Conventional Proof



- π may be hard to find (exponential work or more!)
- π should be easy to check

Properties we want

1) Completeness: Honest P convinces honest V.
   (True statements are provable)
2) Soundness: Dishonest P never convinces honest V
   (False statements are not provable)

WRITE DEFNS

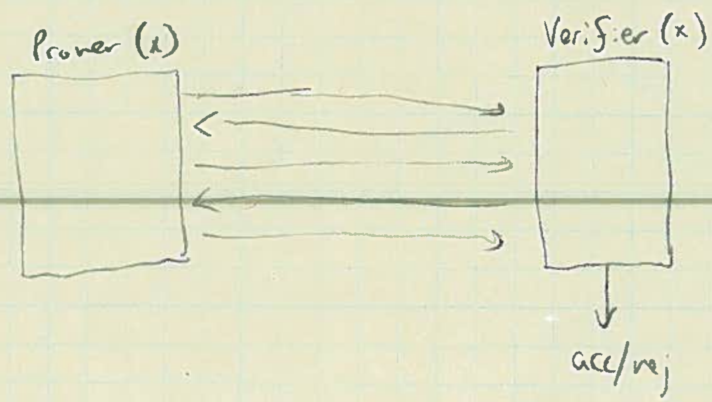
Traditionally, proofs (e.g in book, HW) are one-shot...

↳ Turns out, these proof capture exactly NP languages in NP (if V is deterministic)

↳ Blum: NP = "nifty proof"

We can generalize!

+ add randomness
+ add many rounds of interaction



- IP = complexity class of langs w/ this type of protocol

→ ~~Verifier~~ can use secrets!

→ Makes adaptive queries!

- AM (Goldwasser Sipser) '86
  ↳ turns out secrets not needed

Zero - Knowledge

So now we have IP, who cares?

→ Turns out IPs can have a third amazing property

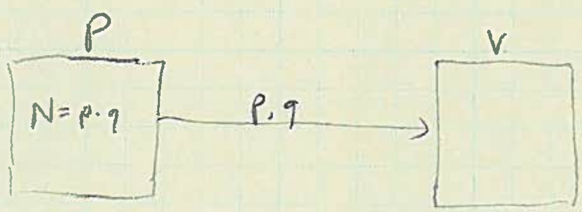
1) Complete

2) Sound

3) Zero Knowledge { verifier "learns nothing" from proof, except that $x \in L$.

↳ What does this mean?!?

e.g. One way for P to convince V that N is an RSA modulus $N = p \cdot q$.



- BUT now V knows factors of N!
- Can we do this w/o leaking factors to V? YES!

We will show that P can convince V of any NP statement ($x \in L$, for $L \in NP$) in ZK, as long as PRF/PRG/OWFs exist.

"Anything provable is provable in ZK."

IP Def'ns

We say that (P, V) is an interactive proof system for L if V is ppt and we have

COMPLETENESS: $\forall x \in L$,

$$\Pr\left[\langle P, V\rangle(x) = 1\right] \geq \frac{2}{3}$$

SOUNDNESS: $\forall x \notin L$, $\forall$ (possibly bad) provers P^*

$$\Pr\left[\langle P^*, V\rangle(x) = 1\right] \leq \frac{1}{3}$$

→ Says nothing about efficiency of P

→ Can amplify probabilities w/ repitition.

ZERO KNOWLEDGE: $\forall$ (possibly malicious) verifiers V^*
$\exists$ a ppt. Sim_{V^*} s.t.
$\forall x \in L$,

(1) Sim fails with prob $< \frac{1}{2}$

(2) When Sim doesn't fail:

$$View_{V^*}\left[P(x) \leftrightarrow V^*(x)\right] = \left\{ Sim_{V^*}(x) \right\}$$

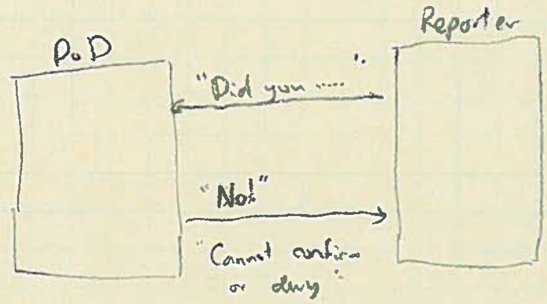
transcript of $P(x)$ talking to $V^*(x)$

Idea of the definition:

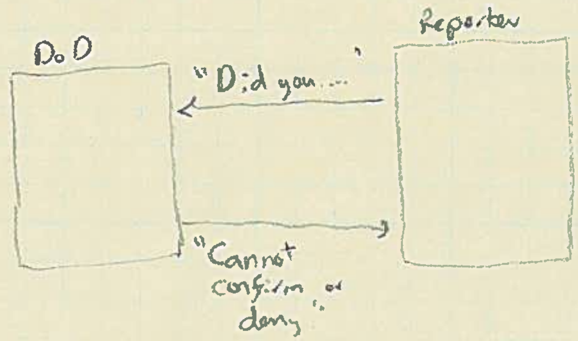
- Whatever V^* could have learned from P, V^* could have learned on its own sitting at home, just given that $x \in L$. (By running Sim_{V^*})
- V^* "learned nothing" from P, except that $x \in L$
- Holds no matter how V^* cheats!
- Again:
  If I can have an algorithm that perfectly simulates my interaction w/ you, then I don't even need to talk to you!

- For full formality

Real-World ZK



Not ZK!



$= \{\mathrm{Sim}_{V^*}(\cdot)\}^?$

$$\mathrm{Sim}_{V^*}(\cdot)\begin{cases} q = V^*(\cdot) \\ a = \text{"Cannot confirm or deny"} \\ \text{output } (q, a) \end{cases}$$

→ If you can perfectly simulate the conversation, no need to have it at all!

→ Important: (1) The input to Sim tells you what leaks to V^*
(2) Must work __for__ __all__ V^*
↳ be careful...

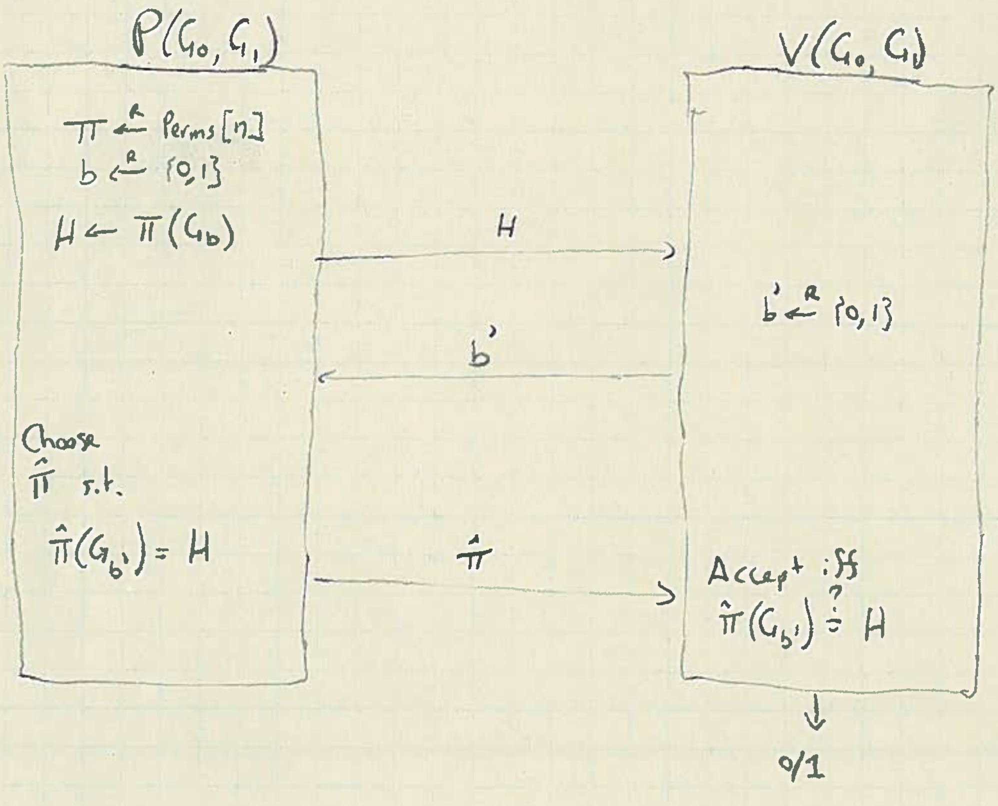
<u>Example</u>: Graph Isomorphism

We say G_0, G_1 are isomorphic ($G_0 \cong G_1$) if $\exists$ a relabeling π of vertices of G_0 s.t. $\pi(G_0) = G_1$

→ GI is <u>not</u> <u>known</u> to be in P
(best known alg Babai 2016 $2^{(\log n)^{O(1)}}$ $= n^{\text{polylog}(n)}$)

→ Can prove $G_0 \cong G_1$ by sending π to V. This leaks info! Not ZK

Given (G_0, G_1), there is a Perfect ZK protocol for GI.



Intuitively:

- If $G_0 \cong G_1$ then H is just a random permutation of G_0, G_1
- $\hat{\pi}$ is just that random permutation

Example GI

Proofs

Completeness: By inspection.

Soundness: If $b \neq b'$, and $G_0 \not\cong G_1$, then $\nexists\ \hat{\pi}$ that causes an honest V to accept.

$\Rightarrow \Pr[\langle P^*, V \rangle = 1] < \frac{1}{2}$ for cheating P^*

↳ Iterate to amplify success prob.

ZK: To show ZK, construct a Simulator that works for all V^*

$\mathrm{Sim}^{V^*}(G_0, G_1)$:

1) Choose $\pi \xleftarrow{R} \mathrm{Perms}[N]$

$b \xleftarrow{R} \{0,1\}$

Set $H \leftarrow \pi(G_b)$

2) Run $b' \leftarrow V^*(G_0, G_1, H)$

3) If $b' \neq b$: output "fail" and restart

Else output (H, b', π).

Why is simulation accurate?

— If $G_0 \cong G_1$, then b' is independent of b

↳ they are $=$ w.p. $\frac{1}{2}$ ... is efficient

— First flow is exactly as in real protocol

↳ b' also as in real proto

— Last flow also is exactly as in real protocol.

$\Rightarrow$ As long as Sim doesn't fail simulation is accurate

$\Rightarrow V^*$ learns nothing

Flavors of ZK Proofs

"Perfect" = as I defined it so far

"Statistical": $\text{View}_{V^*}[P(x) \leftrightarrow V^*(x)] \overset{s}{\approx} \{Sim^{V^*}(x)\}$

These distributions are not identical but almost — no eff alg can distinguish them ("statistical distance is negl")

"Computational": $\text{View}_{V^*}[P(x) \leftrightarrow V^*(x)\} \overset{c}{\approx} \{Sim^{V^*}(x)\}$

These are computationally indist
e.g. if DDH hard $\Rightarrow$ cannot distinguish

"Honest Verifier": $\text{View}_{V^*}[P(x) \leftrightarrow V(x)] \approx \{Sim\ (x)\}$
perfect, stat, computational

Only is ZK when verifier is honest.
$\Rightarrow$ Weaker than full ZK,

"Auxiliary Input" $\forall z \in \{0,1\}^*$ (prior knowledge of V^*)
perfect, stat, comp
$$\text{View}_{V^*}[P(x) \longleftrightarrow V^*(x, z)] \approx \{Sim^{V^*}(x, z)\}$$

"ZK Argument": Soundness holds only against ppt Provers
e.g., if prover breaks Dlog, can convince an honest verifier of a falsity.

(See Goldreich book for full defns)

Zero Knowledge for all of NP

Goldreich, Micali, Widgerson (1986)

NP is the set of problems whose answers are checkable in P
("nifty proofs")

We show that there is a ZKIP for all languages in NP(!), given OWF
↳ If I can prove something to you w/ a written proof, I can prove this to you in ZK.

e.g. Can prove $N = pq$ for λ-bit primes w/o revealing them
Can prove that (g, g^x) has $0 \le x < 2^{128}$ w/o revealing x.
Can prove that $\phi \in SAT$. w/o revealing SAT assignment

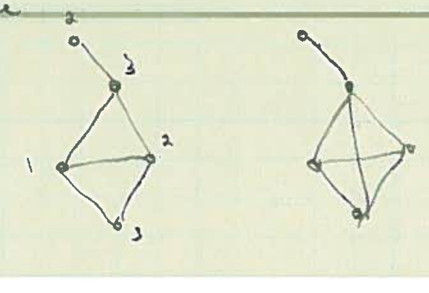
Idea: – Give a ZK protocol for NP-complete language.
– We use 3-coloring
– Since all NP problems are reducible to 3COLOR → all $L \in NP$ have ZK proof
– First time NP-completeness used to show feasibility
↳ Usually show problem is hard.

Recall: 3COLOR

A graph $(G = (V, E)$ is 3-colorable if
$\exists$ coloring $\kappa: V \to \{0,1,2\}$
s.t. $\forall (u,v) \in E \quad \kappa(u) \neq \kappa(v)$.

For the ZK proof, will convince you that $\exists$ a 3-coloring w/o revealing it.

Picture



Not 3 colorable

Preliminary: Commitments

A perfectly binding commitment scheme.

$C~~Com~~$ Com: $\mathcal{M} \times \mathcal{R} \longrightarrow \mathcal{C}$

Hiding: $\forall m_0, m_1 \in \mathcal{M}$

$$\{ r \xleftarrow{R} \mathcal{R} : Com(m_0, r) \} \overset{c}{\approx} \{ r \xleftarrow{R} \mathcal{R} : Com(m_1, r) \}$$

→ Given a "commitment" to m, have no idea what it is

Binding $\forall m_0, m_1 \in \mathcal{M}$ $\forall r_0, r_1 \in \mathcal{R}$, if $m_0 \neq m_1$

$$Com(m_0, r_0) \neq Com(m_1, r_1)$$

→ Value of $Com(m, r)$ only has one valide "opening"

* Think of a commitment as an envelope

↳ can't see inside, but binds you to a valu

* Very useful!

* Can build from OWF, Dlog, RSA, ... ~~CRHF~~

* Assume we get commitments for free.

Thm If perfectly hiding commitments exist, any NP problem has a computational ZK proof.

→ The condition (commitments exist) seems necessary.

↳ look at Goldreich for details.

ZK and NP

Pf We show a computational! ZKP for 3COLORING. $G = (V, E)$, $n = |V|$

$P(G)$		$V(G)$
Fix 3-coloring κ of G. $\pi \xleftarrow{\$} \mathrm{Perm}[3]$ for $i = 1, \ldots, n$ $c_i = \mathrm{Com}(\pi(\kappa(v_i)))$	$\xrightarrow{c_1, \ldots, c_n}$	
		$(v_i, v_j) \xleftarrow{\$} E$
	$\xleftarrow{(v_i, v_j)}$	
Let (r_i, r_j) be randomness used to commit to $\kappa(v_i), \kappa(v_j)$	$\xrightarrow{(r_i, r_j),\ (\kappa(v_i), \kappa(v_j))}$	
		Check $\kappa(v_i) \neq \kappa(v_j)$ $c_i = \mathrm{Com}(\kappa(v_i), r_i)$ $c_j \stackrel{?}{=} \mathrm{Com}(\kappa(v_j), r_j)$ $\downarrow$ 0/1

Note: Prover here can be efficient... you could actually implement this!

ZK for 3COLOR

$$\Pr\left[\begin{array}{l}\text{Cheating prover}\\ \text{escapes after } t \text{ trials}\end{array}\right] \le \left(\Pr\left[\text{don't catch } P\right]\right)^t$$

$$\le \left(1 - \frac{1}{n^2}\right)^t$$

$$\le \left(e^{-\frac{1}{n^2}}\right)^t$$

$$\le \left(e^{-\frac{t}{n^2}}\right)$$

Using $(1+x) \le e^x$

If you repeat $t \approx n^3$ times, this prob is $\le e^{-n}$

ZK for 3COLOR

We need to show

Complete: Honest P convinces honest V. ✓

Sound: - Say $G \notin$ 3COLOR.
- Then $\exists$ an edge $e^* = (v_i, v_j) \in E$ s.t. $K(v_i) = K(v_j)$.
- With prob $\frac{1}{|E|} \geq \frac{1}{n^2}$, Verifier chooses e^* and prover gets caught

↳ Need that Com is perfectly binding here.

ZK: We need a simulator

$Sim^{V^*}(G)$:

1) Commit to a random coloring K of G.
2) Invoke $(v_i, v_j) \leftarrow V^*(G, c_1, \ldots, c_n)$
3) If $K(v_i) = K(v_j)$, abort
4) Else, open $K(v_i)$, $K(v_j)$.

* Simulator fails w/ prob $3 \cdot (\frac{1}{3})(\frac{1}{3}) = \frac{1}{3}$ $\longrightarrow$ it's efficient.
* If the simulator doesn't fail, yields a perfect simulation

↳ Here we need that Com is computationally hiding.

$\Longrightarrow$ Can rerun many times to reduce prob that a cheating prover escapes detection.