Zero knowledge - May 3

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

Today:
- So far we have covered crypto primitives
  - PGP, PKE, 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:
    - [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
    - 1/4 of construction but 1/4 of depth of ZK!
  - Definition is > 1/2 the battle.
  - Foundational paper in theoretical crypto

To some extra.

\[
\begin{align*}
\text{What is possible in theory} \\
\quad \text{"polynomial time" (complex, theory view) }
\end{align*}
\]
\[
\begin{align*}
\text{What runs on my computer} \\
\quad \text{(practice, real world)}
\end{align*}
\]
Review: Elliptic Curves & Pairings

Why EC?
RSA: $O(x^3)$ bit keys
ECC: $2x$ bit key.

By factoring is easier than EC hard problems.

Idea: We can do DH in an abstract group $(G, \cdot)$

$g^a = g \cdot g$

$g \in$ group element

EC Group: $(x, y) \in \mathbb{F}_p^2$
$p = 2^{56}$
$s.t. \quad y^2 = x^3 + Ax + B \in \mathbb{F}_p$
(Also "point at infinity" = identity element)

9 points on curve, 9 \cdot p.

Can construct EC group to have order (prime) $q$.

Hard Problems

$Dlog:\quad a \in \mathbb{Z}_q$
Given $(g, g^a)$, produce $a$

$CDH:\quad a, b \in \mathbb{Z}_q$
Given $(g, g^a, g^b)$ produce $g^{ab}$

$DDH:\quad a, b, c \in \mathbb{Z}_q$
Distinguish $(g, g^a, g^b, g^c)$ from $(g, g, g, g)$
Review: EC & Pairings

→ DDH makes life easy!
  Saw PRQ and PRF (Naor-Reingold) from DDH
  Hard to get directly from Dlog, CDH

Pairing
  - Defined over *special* EC groups
    \[ e: G \times G \rightarrow G_t \]
  s.t.,
  1) \( e \) is computable
  2) Bilinear \( \forall a, b \in \mathbb{Z}_p, g \in G \)
    \[ e(g^a, g^b) = e(g, g)^{ab} \]
  3) Non-degenerate \( e(g, g) \) generates \( G_t \)
    if \( g \) generates \( G \)

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

→ DDH is easy in \( G \),
  CDH in \( G_t \) 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 (encrypted to policy/attributes)
    - SNARKs

Sparked a new wave of crypto research in 2006s

And... all implementable!!!
Interactive Proofs

Def. by Goldwasser, Micciancio, 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.

\[ x \in L \text{ } \text{language} \] \[ \text{For us, this is } \text{a statement to be proved} \]

**Examples**

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

\[ N \in \exists p, q \mid p, q \text{ are 1024-bit primes} \]

\[ \text{instance } \text{language} \]

"(g, h) is st. \text{h = } g^a \text{ and a is odd}"

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

"The Pythagorean Thm is true."

\[ \text{PYTHM} \in \exists \text{true statements in some } \text{logical system} \]

"f(x) = [(x + 3)^5 + (x - 2)^{100}]^{50} - 27 \text{ has 15 distinct roots mod } p."

\[ f \in \exists \text{polys mod } p \text{ with 15 distinct roots} \]
Conventional Proof

Prover (x) -> Verifier (x)

- It may be hard to find (exponential work or more!)
- It 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 proofs (ad-hoc) exactly NP languages
  - in NP (if V is deterministic)
  - Blum: NP = "nifty proofs"

We can generalize:

+ add randomness
+ add many rounds of interaction

- IP = Complexity class of things w/ this type of protocol
  - Verifier can use secrets!
  - Makes more adaptive queries!
  - AM (Goldwasser, Sipser) '86
    - things 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?!?

Example: one way for $P$ to convince $V$ that $N$ is an RSA modulus $N=pq$.

- 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 $Z^k$, as long as PRE/ARG/OWF exist.

"Anything provable is provable in $Z^k".$
We say that \((P, V)\) is an interactive proof system for \(L\) if \(V\) is ppt and we have

**Completeness:** \(\forall x \in L,\)
\[P_c[\langle P, V \rangle(x) = 1] \geq \frac{2}{3}\]

**Soundness:** \(\forall x \notin L, \forall (\text{possibly bad}) P^*\) (\(P^*\) proves \(P\))
\[P_c[\langle P^*, V \rangle(x) = 1] \leq \frac{1}{3}\]

→ Says nothing about efficiency of \(P\)
→ Can amplify probabilities \(\sqrt{n}\) repetition.

**Zero Knowledge:** \(\forall (\text{possibly malicious}) \text{ verifiers } V^*\)
\[\exists \text{ a ppt, } S^{\text{in}}_\text{Sim, } S^+\]
\[\forall x \in L,\]
\[\text{Sim fails with prob } \leq \frac{1}{2}\]

1. Sim fails with prob \(\leq \frac{1}{2}\)
2. When Sim doesn’t fail:
\[\text{View}_{V^*}[P(x) \leftrightarrow V^*(x)] = \{S^{\text{in}}_\text{Sim}(x)\}^n\]

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 \(S^{\text{in}}_\text{Sim}\))
- \(V^*\) “learned nothing” from \(P\), except that \(x \in L\)
- Holds no matter how \(V^*\) cheats!
- Again:
  - If I can make an algorithm that perfectly simulates my interaction w/ you,
    then I don’t even need to talk to you!

- For Bell Statements
Re.1-World Zk

\[ \text{Not Zk.} \]

\[
\text{Sim}(\cdot) = \begin{cases} a = V^*(\cdot) & \text{"Cannot confirm",} \\ a = \text{"Cannot confirm",} & \text{or clan} \\ \text{output } (q_0, a) \end{cases}
\]

→ 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) Need work for all \( V^* \)
→ be careful...
Example: Graph Isomorphism

We say $G_0, G_1$ are isomorphic ($G_0 \cong G_1$) if

$\exists$ a relabeling $\pi$ of vertices of $G_0$ st.

$\pi(G_0) = G_1$.

$\implies$ GI is not known to be in $\mathbb{P}$

(best known alg. Babai 2016 $\geq \log^2(n)$)

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

Given $(G_0, G_1)$, there is a perfect ZK protocol for GI,

\[ \mathcal{P}(G_0, G_1) \]

\[ \begin{array}{c}
\pi \in \text{Perms}[n] \\
b \in \{0,13\} \\
h = \pi(G_0) \\
\text{Choose } \pi \text{ s.t.} \\
\pi(G_0) = H \\
\end{array} \]

\[ \mathcal{V}(G_0, G_1) \]

\[ \begin{array}{c}
b' \in \{0,13\} \\
\text{Accept :} \frac{H}{H} \text{ if} \\
\pi(G_0) = H \\
\end{array} \]

Intuitively:

- If $G_0 \cong G_1$, then $H$ is just a random permutation of $G_0, G_1$.
- $\pi$ is just that random permutation.
Example GI.

Completeness: By inspection.

Soundness: If $b \neq b'$, and $G_0 \neq G_1$, then $A^\pi$ that causes an honest $V$ to accept.

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

$$\Rightarrow$$ Iterate to amplify success prob.

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

[Sim$_V^*$($G_0, G_1$):

1) Choose $\pi \in$ Perm[$N$]
    $b \in \{0, 1\}$
    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 $(H, b', \pi)$.

Why is simulation accurate?

- If $G_0 \equiv G_1$, then $b'$ is independent of $b$
  $\Rightarrow$ they are $= w.p. \frac{1}{2}$ w.p. exactly.

- First, slow is exactly as in real protocol
  $b \neq b'$ also as in real prot.

- Last, slow 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": $\Pr[P(x) \leftrightarrow V_1(x)] \approx \mathcal{S}_\text{Sim}^*(x)$

These distributions are not identical, but almost — no ef alg can distinguish them ("statistical distance is neg!")

"Computational": $\Pr[P(x) \leftrightarrow V_2(x)] \approx \mathcal{S}_\text{Sim}^*(x)$

There are computationally indist.

Eg: if DDH hard $\Rightarrow$ cannot distinguish

"Honest Verifier": $\Pr[P(x) \leftrightarrow V_3(x)] = \mathcal{S}_\text{Sim}^*(x)$

Perfect, stat, computational

Only is ZK when verifier is honest

$\Rightarrow$ weaker than full ZK,

"Auxiliary Input": $\forall z \in \{0,1\}^k$ (prior knowledge of $v^*$)

$\Pr[P(x) \leftrightarrow V_4(x,z)] = \mathcal{S}_\text{Sim}^*(x,z)$

"ZK Argument": Soundness holds only against ppt provers

S"s prover breaks Dlos, can convince an honest verifier of a falsity.

(See Goldreich book for full defns)
Zero Knowledge for all of NP
Goldreich, Micali, Wigderson (1988)

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(1), given OWF

Is I can prove something to you with a written proof, I can prove this to you in ZK.

e.g. Can prove \( N=pq \) for 3-bit primes wo revealing them
Can prove that \( (g, g^e) \) has \( 0 \leq x \leq 2^{100} \) wo revealing \( x \)
Can prove that \( \phi \in SAT \) wo revealing SAT assignment

Idea: - Give a ZK protocol for NP-complete language
- We use 3-coloring
- Since all NP problems are reducible to 3COLOR in all logNP, have ZK proof
- First time NP-completeness used to show feasibility

Let usually show problem is hard.

Recall: 3COLOR

A graph \( (G=(V, E)) \) is 3-colorable if

3-coloring \( \kappa : V \rightarrow \{0,1,2\} \),

s.t. \( \forall (u,v) \in E \) \( \kappa(u) \neq \kappa(v) \).

For the ZK proof, will convince you that \( \exists \)

a 3-coloring wo revealing it.

Picture:

\[ \text{Not 3-colorable} \]
A perfectly binding commitment scheme.

\[ \text{Comm}: \mathbb{M} \times \mathbb{R} \rightarrow \mathcal{C} \]

Hiding:
\[ \forall m_0, m_1 \in \mathbb{M} \]
\[ \{ \exists r \in \mathbb{R} : \text{Comm}(m_0, r) \} \land \{ \exists r \in \mathbb{R} : \text{Comm}(m_1, r) \} \]
\[ \rightarrow \text{Given a "commitment" to } m, \text{ have no idea what it is.} \]

Binding:
\[ \forall m_0, m_1 \in \mathbb{M}, \forall r, r' \in \mathbb{R}, \text{ if } m_0 \neq m_1, \]
\[ \text{Comm}(m_0, r) \neq \text{Comm}(m_1, r') \]
\[ \rightarrow \text{Value of } \text{Comm}(m, r) \text{ only has one valid "opening"} \]

* Think of a commitment as an envelope.
  \[ \rightarrow \text{can't see inside, but binds you to a value.} \]

* Very useful!

* Can build from OWF, Dlog, RSA, ...<BR>\[ \text{O.RIP} \]

** This IS perfectly hiding commitments exist, any \text{NP} problem has a computational \text{NP} proof.

\[ \rightarrow \text{The condition (commitments exist) Seems necessary.} \]
\[ \rightarrow \text{See Goldreich for details.} \]
We show a ZKP for 3-COLORING.

\( G = (V, E), \ n = |V| \)

**P(G)**  

Fix 3-coloring \( k \) of \( G \).  
\( \pi \leftarrow \text{Perm}[3] \)

for \( i = 1, \ldots, n \)
  
  \[ c_i = \text{ Commit}(\pi(k(v_i))) \]

Let \((p_i, r_i)\) be random reals used to commit to \( k(v_i) \)

\( (v, y), (r, s), (k(v), k(y)) \)

Check \( k(v) \neq k(y) \)

\[ c_i = \text{ Commit}(k(v), r_i) \]
\[ c_j = \text{ Commit}(k(y), r_j) \]

0/1

**V(G)**  

Note: Prove here can be efficient... you could actually implement this!
The 3Color

\[ Pr[\text{cheating program escapes after } t \text{ trials}] \leq \left( Pr[\text{catch } P] \right)^t \leq \left( 1 - \frac{1}{2^k} \right)^t \]

\[ = \left( e^{-\frac{t}{2^k}} \right)^t \leq \left( e^{-\frac{t}{2^k}} \right) \]

Using \((1+x) \leq e^x\)

If you repeat \(t = n^3\) times, this prob is \(\leq e^{-n}\)
ZK for 3COLOR

We need to show

Complete: Honest P convinces honest V. \( \checkmark \)

Sound:
1. Say \( G \in 3\text{COLOR} \).
2. Then \( \exists \) an edge \( e = (v_i, v_j) \in E \) st. \( K(v_i) \neq K(v_j) \).
3. With prob \( \frac{1}{161} > \frac{1}{n^3} \), Verifier chooses \( e^* \) and
   Prover gets caught
4. Need that Com is perfectly binding here.

ZK: We need a simulator

\( \hat{\text{Sim}}(G) \):
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 \( 3 \cdot \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{3} \) \( \Rightarrow \) it's efficient.
* If the simulator doesn't fail, yields a perfect simulation

Here we need that Com is computationally binding.

\( \Rightarrow \) Can rerun many times to reduce prob that a cheating prover escapes detection.