Non-Interactive ZK
+ the Fiat-Shamir Heuristic
Today:

- $\Sigma$ Protocols
  - Gate boolean gate constraints
- Non-interactive ZK?
  - Fiat-Shamir Heuristic
    - Schnorr Signatures
    - HVZK $\Sigma$-protocol $\rightarrow$ NIZK (ROM)
Last Time: Σ Protocols \( p \xrightarrow{t} v \) for an NP relation \( R(x, w) \)

Properties:

- **Special Soundness**: \( \exists E. \forall \text{ pairs of accepting } (b, c, z), (b', c', z') \)
  \[ w \not\in \{ x, E(b, c, z, b', c', z') \} \in R \]

(special case of **knowledge Soundness**: "knowledge error"
\[ \Pr [ (x, w) \in R : w \leftarrow E'(x)] > \Pr [ (p, w)(x) = 1] - k \]

- **Special HVZK**: \( \exists \text{ det., eff., Sim}(x, c) \rightarrow (b, z), v_0 \)
  \[ \forall (x, w) \in R. \{ (b, c, z) : c \neq c' \}; b_z - \text{Sim}(x, c) = \{ (p(x, w), v(x)) = 1 \} \]

  \[ \forall x \forall c \forall z = \text{Sim}(x, c) \rightarrow (b, c, z) \text{ is accepting} \]

  (the last lecture accidentally omitted this from the SHVZK def. However, this property is needed to show that to OR Π \( \Pi \) is complete. Also, Schnorr's protocol does have this property.

- Why is Proof-of-knowledge defined this way? Let's see...

  - A more natural idea for Pok: "\( w \) can be efficiently computed from \( P^* \)'s state"
  \[ \rightarrow \exists E \forall P^* (x, E(x, \text{state}_P)) \in R \]

  (note: our real Pok def is probabilistic. Here, we omit the probabilities for brevity...)

  - Problem: since \( ! \) is unbounded, \( ! \) may be unbounded
  \[ \rightarrow \text{ so, } E \text{ could not be efficient} \]

  - Idea: each msg from \( P^* \) has \( \text{poly}(\lambda) \) size (and msgs are \( \text{fn}(\text{state}_P) \))
  \[ \rightarrow \exists E \forall P^* (x, E(x, \text{all msgs } P \text{ could send}) \in R \]

  - Problem: msgs are a tree. If first \( v \) msg has size \( \lambda \), tree size \( \geq 2^{\lambda m_{P \rightarrow V}} \)
  \[ \rightarrow \text{ again, } E \text{ could not be efficient...} \]

  - Idea: allow \( E \) to **explore** the message tree
  \[ \rightarrow \text{ This is exactly what 'rewinding' does} \]

  - Idea: allow \( E \) to explore the message tree
  \[ \rightarrow \exists E \forall P^* (x, E^*(x)) \in R \]

\( \Rightarrow \) Rewrite this probabilistically to get Pok def

\( \Rightarrow \) a notation for exploring the message tree
Towards a $\Sigma$ protocol for circuit SAT

Recall: Pedersen commitments $g, h \in G$

$\text{Commit}(m \in \mathbb{Z}_p, r \in \mathbb{Z}_p) = g^m h^r$

$m_0 \leftarrow \bigwedge m_i$
$m_i \leftarrow \bigvee m_i$
$c_i = \text{Commit}(m_i, r_i)$

$\begin{array}{c|c|c|c}
  m_0 & m_1 & m_2 \\
  0 & 0 & 0 \\
  0 & 1 & 0 \\
  1 & 0 & 1 \\
  1 & 1 & 1 \\
\end{array}$

$m_0 = 0 \text{ AND } m_1 = 0 \text{ AND } m_2 = 0$
$m_0 = 0 \text{ AND } m_1 = 1 \text{ AND } m_2 = 0$
$m_0 = 1 \text{ AND } m_1 = 0 \text{ AND } m_2 = 0$
$m_0 = 1 \text{ AND } m_1 = 1 \text{ AND } m_2 = 1$

Q: How to show $m_i = 0$ or $m_i = 1$? (public: $g, h, c_i$)

Recall: Schnorr's Protocol

A: Pok for $\{ (x = (g, h) \in G^2, w \in \mathbb{Z}_p): g^w = h^2 \}$

$P(g, h, w)$

$r \in \mathbb{Z}_p$

$g^r \rightarrow c \leftarrow c^d \cdot c$

$z \leftarrow \text{w}$

$g^z \rightarrow h^z \cdot g^r$

$V(g, h)$

$A$: To show $m = 0$, show $c = g^m h^r$

$\Rightarrow$ use schnorr to show

$\downarrow w$

$\Rightarrow$ works for any gate truth table.

$\Rightarrow$ HW: Circuit SAT.
NI2Ks

"Non-Interactive Zero-Knowledge (Proofs)"

Q: Suppose we have a \( \Pi \) sound, \( \mathcal{ZK} \) NI pf

\[ \exists \text{Sim}(x) \rightarrow \Pi' \]

Verify (an eff. alg.) can't tell

\[ x \in L \iff \exists \Pi \quad \text{Verify}(x, \Pi) \Rightarrow \text{Verify}(x, \text{Sim}(x)) \]

sound complete \( \mathcal{ZK} \) \( \text{a PPT alg for } x \in L \)

"bound-error probabilistic poly time"

So, in the standard model, if

\( L \) has a NI2K, then \( \mathcal{LEBPP} \)

\( \Rightarrow \) So, \( \mathcal{ZK} \) doesn't mean much....

But NI2Ks are possible if we change the model!

\[ \text{RO} \]

\[ \text{CRS} \]

\[ P \overset{\Pi}{\rightarrow} \mathcal{V} \]

\[ P \overset{\Pi}{\rightarrow} \mathcal{V} \]
NT, ZK, Poh in ROM

Fiat-Shamir Heuristic: replace $V$ w/ $H$.
Schnorr: $h = g^w$

$P(g,h,w)$
$\begin{array}{c}
    r \in \mathbb{Z}_p \\
    z \in \text{cwtr}
\end{array}$
$\begin{array}{c}
    c = H(gh, gr) \\
    z \rightarrow g^z h^c gr
\end{array}$

$V(g,h)$

Analysis: Completeness is direct

ZK:
Q: What does ZK mean in ROM

A: Simulate $P \leftrightarrow V$ transcript & RO queries called "programming" the RO

Sim:
map $M : G^3 \rightarrow \mathbb{Z}_p$

$c \in \mathbb{Z}_p$
$u, z \leftarrow \text{Sim}_{\text{Schnorr}} (g, h, c)$
set $M[(g, h, u)] \leftarrow c$
output $(u, c, z)$
on RO query $x$: if $x \notin M$, output $M[x]$
Pok
Q: What does Pok mean in ROM?

Standard
\[
P \xrightarrow{\text{mi}} V \xrightarrow{\text{ri}}
\]

A: rewind, choosing \(V\) messages and \(H\) outputs

Schnorr-FS:
\[
E: \quad c \neq c' \text{ if } C
\]
run \(P^*\): when it queries for challenge, give \(c\)
and two transcripts \(s, E_{\text{schnorr}}\) to get witness

Bonus: Signatures
Simply add \(m\) to the hash
\[
H: G^3 \times M \rightarrow \mathbb{Z}_p
\]
\[
P \text{Sign}(\text{pk, sk, } m, g) := r \in \mathbb{Z}_p
\]
\( c = H(pk, g, g^r, m) \)
\( z = sk \cdot c + r \)
\( \sigma = (z, g^z) \)

Verify \( (pk, m, \sigma) \):
\( c = H(pk, g, g^n, m) \)
\( g^z = pk^{c_2} \cdot g^r \)

Notes:
- send \( c \), not \( g^r \), compute \( g^r = g^{\frac{c}{pk}} \)
  check \( c = H(\cdot, \cdot) \)

\( \sigma \in \mathbb{Z}_p^* \)
\( \text{since soundness is } \frac{1}{101}, \text{take } c \text{ to be } 128 \text{ bits} \)
\( \text{total size is } 384 \text{ bits} \)

Compare:
RSA-FDH: \( \approx 3072 \text{ bits} \)
BLS: 384 bit (pairing group size)
A general perspective:

Fiat Shamir lifts a $\Sigma^1$-protocol with completeness + SHVZK to sks to a non-interactive ZK-PAK (in the ROM). It's also useful for other constant-round protocols (public-coin) and some $O(1)$-round protocols too!