Non-Interactive ZK

→ the Fiat-Shamir Heuristic

Today:

- ☐ Σ Protocols
  - ☐ gate boolean gate constraints
- ☐ Non-interactive ZK?
  - ☐ Fiat-Shamir Heuristic
    - ☐ Schnorr Signatures
    - ☐ HVZK Σ-protocol $\rightarrow$ NIZK (ROM)

Last Time: Σ Protocols $P \xleftrightarrow[z]{t,\ c} V$ for an NP relation $R(x,w)$

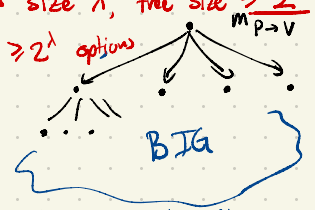
Properties:

- Special Soundness: $\exists \overset{\text{det., eff.}}{\mathcal{E}}$. $\forall$ pairs of accepting $(t,c,z),(t',c',z')$ w/ $c \neq c'$ $(x, \mathcal{E}(t,c,z,t',c',z')) \in R$

  (special case of Knowledge Soundness: "knowledge error"
  $\Pr[(x,w)\in R : w \leftarrow \mathcal{E}^{P^*}(x)] \geq \Pr[\langle P,V\rangle(x)=1] - k$)

- Special HVZK: $\exists$ det., eff. $\mathrm{Sim}(x,c) \to (t,z)$, s.t.
  - $\forall (x,w)\in R$. $\{(t,c,z) : c \xleftarrow{\$} C;\ t,z \leftarrow \mathrm{Sim}(x,c)\} = \{\langle P(x,w), V(x)\rangle = 1\}$
  - AND $\forall x\ \forall c$ $t,z = \mathrm{Sim}(x,c) \to (t,c,z)$ is accepting
    - the last lecture accidentally omitted this from the SHVZK def^n. However, this property is needed to show that to OR Π is complete. Also, Schnorr's protocol does have this property.

- Why is Proof-of-knowledge is defined this way? Let's see...
  - A more natural idea for PoK: "w can be efficiently computed from P^*'s state"
    $\to \exists \mathcal{E}\ \forall P^*\ (x, \mathcal{E}(x, \underline{\text{state}_{P^*}})) \in R$ Note: our real PoK def^n is probabilistic. Here, we omit the probabilities for brevity...
  - Problem: since P^* is unbounded, $\uparrow$ may be unbounded
    $\to$ so, $\mathcal{E}$ could not be efficient
  - Idea: each msg from P^* has poly(λ) size (and msgs are fn(state$_{P^*}$))
    $\to \exists \mathcal{E}\ \forall P^*\ (x, \mathcal{E}(x, \underline{\text{all msgs } P^* \text{ could send}})) \in R$
  - Problem: msgs are a tree. If first $\mathcal{V}$ msg has size λ, tree size $\geq \underline{2^\lambda}$ ($\geq 2^\lambda \cdot \text{poly}(\lambda)$)
    $\to$ again, $\mathcal{E}$ could not be efficient...



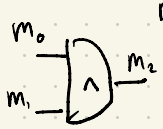
  - Idea: allow $\mathcal{E}$ to explore the message tree
    $\to$ This is exactly what 'rewinding' does
    $\to \exists \mathcal{E}\ \forall P^*\ (x, \underbrace{\mathcal{E}^{P^*}(x)}) \in R$ $\Rightarrow$ Rewrite this probabilistically to get PoK def^n
    a notation for exploring the message tree

Towards a Σ protocol for circuit SAT

Recall: Pedersen commitments $g, h \xleftarrow{\$} \mathbb{G}$

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

$r_0, r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p$



$c_i = \text{Commit}(m_i, r_i)$

m_0	m_1	m_2
0	0	0
0	1	0
1	0	0
1	1	1

$m_0 = 0$ AND $m_1 = 0$ AND $m_2 = 0$
OR
$m_0 = 0$ AND $m_1 = 1$ AND $m_2 = 0$
OR
$m_0 = 1$ AND $m_1 = 0$ AND $m_2 = 0$
OR
$m_0 = 1$ AND $m_1 = 1$ AND $m_2 = 1$

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

Recall: Schnorr's Protocol

A PoK for $\{(x = (g,h) \in \mathbb{G}^2,\ \underbrace{w}_{\text{exponent}} \in \mathbb{Z}_p) : g^w = h\}$ (base, power)

$P(g,h,w)$		$V(g,h)$
$r \xleftarrow{\$} \mathbb{Z}_p$		
	$\xrightarrow{g^r}$	$c \xleftarrow{\$} \mathbb{C}$
$z \leftarrow wc + r$	$\xleftarrow{c}$	
	$\xrightarrow{z}$	$g^z \stackrel{?}{=} h^c g^r$

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

→ use schnorr to show $c = h^r$ ("h", "g", "w")

To show $m = 1$, show $c = g^m h^r$ $\quad c = gh^r \quad c/g = h^r$ ("h", "g", "w")

→ schnorr

→ Works for any ~~gate~~ truth table.

→ HW: Circuit SAT.

NIZKs

"Non-Interactive Zero-Knowledge (Proofs)"

$$\underline{P} \xrightarrow{\pi} \underline{V(x)}$$

$\text{Verify}(x, \pi) \to \{0,1\}$

Q: Suppose we have a complete sound, ZK NI pf

$\exists\, \text{Sim}(x) \to \pi'$

Verify (an eff. alg) can't tell

$$x \in \mathcal{L} \underbrace{\leftrightarrow}_{\text{sound \& complete}} \exists \pi \ \text{Verify}(x, \pi) \underbrace{\leftrightarrow}_{\text{ZK}} \underbrace{\text{Verify}(x, \text{Sim}(x))}_{\text{a PPT alg for } x \overset{?}{\in} \mathcal{L}}$$

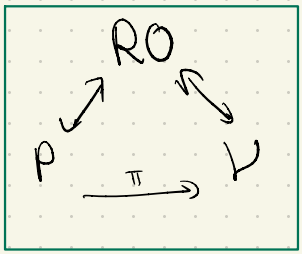
"bounded-error probabilistic poly time"

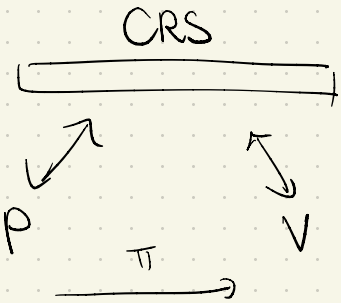
So, in the standard model, if

$\mathcal{L}$ has a NIZK, then $\mathcal{L} \in \text{BPP}$:(

→ So, ZK doesn't mean much....

But NIZKs are possible if we <u>change the model</u>!





NI, ZK, PoK in ROM

Fiat-Shamir Heuristic: replace ~~V~~ w/ H.
challenge

Schnorr: $h = g^w$

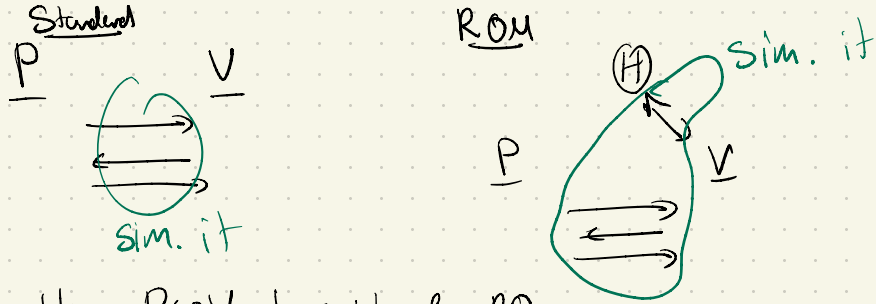
$P(g,h,w)$		$V(g,h)$
$r \xleftarrow{\$} \mathbb{Z}_p$	$\xrightarrow{g^r}$	$c \leftarrow H(g,h,g^r)$
$z \leftarrow cw + r$	$\xleftarrow{c}$	
	$\xrightarrow{z}$	$g^z \overset{?}{=} h^c g^r$

$P(g,h,w)$		$V(g,h)$
$r \xleftarrow{\$} \mathbb{Z}_p$		
$c \leftarrow H(g,h,g^r)$		
$z \leftarrow cw + r$	$\xrightarrow{(g^r, z)}$	$c \overset{?}{=} H(g,h,g^r)$
		$g^z \overset{?}{=} h^c g^r$

Analysis: Completeness is direct

ZK:

Q: What does ZK mean in ROM



A: Simulate $P \leftrightarrow V$ transcript & $\underbrace{\text{RO queries}}$

called "programming" the RO

Sim:

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

$c \xleftarrow{\$} \mathbb{Z}_p$

$u, z \leftarrow \text{Sim}_{\text{Schnorr}}((g,h), c)$

set $M[(g,h,u)] \leftarrow c$

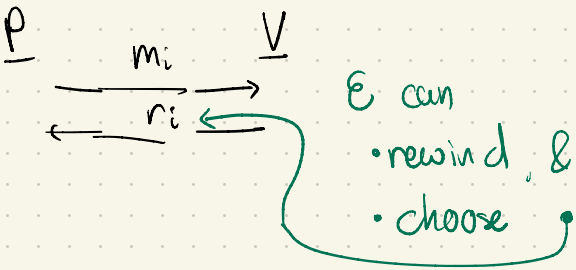
output (u, c, z)

on RO query x: if $x \notin M$, set $M[x] \xleftarrow{\$} \mathbb{Z}_p$, output $M[x]$

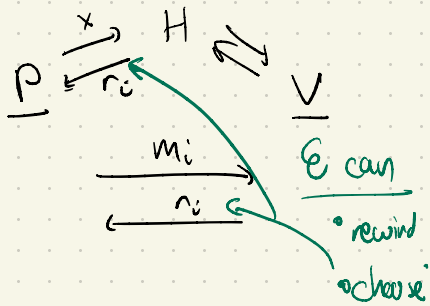
PoK

Q: What does PoK mean in ROM?

Standard



ROM



A: rewind, choosing $\mathcal{U}$ messages and H outputs

Schnorr-FS:

$\mathcal{E}$:

$c \neq c' \xleftarrow{\$} C$

run P^*: when it queries for challenge, give c

" give c'

use two transcripts + $\mathcal{E}_{\text{Schnorr}}$ to get witness

Bonus: Signatures

Simply add m to the hash

$H: G^3 \times M \to \mathbb{Z}_p$

$\cancel{P}$ Sign(pk, sk, m, g):

$r \xleftarrow{\$} \mathbb{Z}_p$

$$c \leftarrow H(pk, g, g^r, m)$$

$$z \leftarrow sk \cdot c + r$$

$$\sigma \leftarrow (z, g^r)$$

Verify(pk, m, σ, g):

$$c \leftarrow H(pk, g, g^r, m)$$

$$g^z \stackrel{?}{=} pk^c \cdot g^r$$

Notes:

- send c, not g^r, compute $g^r \leftarrow g^z / (pk)^c$
  check $c \stackrel{?}{=} H(\cdot\cdot)$
  - → now, $\sigma \in \mathbb{Z}_{p^+}$
  - → since soundness is $\frac{1}{|C|}$, take c to be 128 bits
  - → total size is 384 bits

Compare:

RSA-FDH: ≈ 3072 bits

BLS: 384 bits (pairing group size)

A general perspective:

Fiat Shamir lifts a Σ-protocol w/ completeness + SHVZK + SKS to a non-interactive ZK-PoK (in the ROM)

It's also useful for other constant-round (public-coin) protocols (and some $\omega(1)$-round protocols too!)