Lecture 11

- Σ-protocols for boolean gate constraints

- Non-interactive ZK?
  - ↳ Fiat-Shamir Heuristic
    - Schnorr Signatures
    - HVZK Σ-protocol $\rightarrow$ NIZK (ROM)

Made with Goodnotes

Towards a Σ-protocol for Circuit-SAT

Recall: Pedersen Commitments

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

Say you have 3 commitments c_1, c_2, c_3. A prover wants to convince verifier that it knows $m_1, m_2, m_3 \in \{0,1\}$ and $r_1, r_2, r_3 \in \mathbb{Z}_q$ s.t. $\forall i \in \{1,2,3\}$ $c_i = g^{m_i} h^{r_i}$ and $m_1 \wedge m_2 = m_3$

↑ this corresponds to L_{AND} you are given in HW3!

Idea: since m_1, m_2, m_3 are bits, there are only 8 possible combinations of values, and only 4 of these combos are in the language L_{AND}

So it suffices to prove:

$$(m_1 = 0 \text{ AND } m_2 = 0 \text{ AND } m_3 = 0)$$
$$\text{OR}$$
$$(m_1 = 0 \text{ AND } m_2 = 1 \text{ AND } m_3 = 0)$$
$$\text{OR}$$
$$(m_1 = 1 \text{ AND } m_2 = 0 \text{ AND } m_3 = 0)$$
$$\text{OR}$$
$$(m_1 = 1 \text{ AND } m_2 = 1 \text{ AND } m_3 = 1)$$

Made with Goodnotes

- we know how to do AND/OR of Σ-protocols from last time, so we just need to see how to prove that c_i commits to 0 or 1!

Q: How to show $m_i = 0$ or $m_i = 1$?

Recall Schnorr's Protocol

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

base power exponent

$$\begin{array}{ccc}
\underline{P(g,h,w)} & & \underline{V(g,h)} \\
r \xleftarrow{\$} \mathbb{Z}_p & & \\
 & \xrightarrow{\quad g^r \quad} & c \xleftarrow{\$} C \\
z \leftarrow wc + r & \xleftarrow{\quad c \quad} & \\
 & \xrightarrow{\quad z \quad} & g^z = h^c g^r
\end{array}$$

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

$\rightarrow$ use Schnorr to show $c = h^r$

"h" "g" "w"

To show $m = 1$, show $c = g^m h^r \Rightarrow c = g h^r$

$\rightarrow$ use Schnorr to show $c/g = h^r$

"h" "g" "w"

↳ Works for any truth table!

↳ HW: Circuit-SAT

Made with Goodnotes

NIZKs

Q: Σ-protocols give us 3-message ZK protocols. Can we do better? Can we get 1-message ZK protocols? If so, for which languages?

"Non-Interactive Zero Knowledge (Proofs)"

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

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

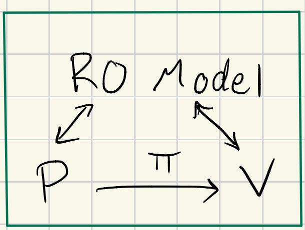
↳ Suppose we have a complete, sound, ZK, non-interactive proof.

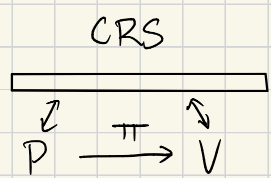
↳ this means $\exists$ Sim$(x) \to \pi'$ that verifier (efficient distinguisher) can't distinguish from real proof

↳ $x \in L \underset{\text{sound/complete}}{\longleftrightarrow} \exists\, \pi$ Verify$(x,\pi)=1 \underset{\text{ZK}}{\longleftrightarrow} \underbrace{\text{Verify}(x, \text{Sim}(x))}_{\text{a PPT alg for } x \overset{?}{\in} L \text{ (BPP)}}$

*Intuition: when proof is 1 message, Sim alg should be able to output the message π

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





NI, ZK, PoK in ROM with Fiat-Shamir Heuristic

Fiat-Shamir allows us to convert Σ-protocol for NP relation R into a NIZKPoK in RO model

Σ-Protocols

$P((x,w) \in R)$ $\qquad\qquad\qquad\qquad$ $V(x)$

$\xrightarrow{\text{commitment } t}$

$c \xleftarrow{\$} C$ challenge chosen uniformly at random

$\xleftarrow{\text{challenge } c}$

$\xrightarrow{\text{response } z}$

$\downarrow$

$\{0,1\}$ as deterministic function of (x, t, c, z)

Properties

1. Completeness: $\forall x \in L$, $\Pr[\langle P(x,w), V(x)\rangle = 1] = 1$
2. Special-soundness: $\exists$ deterministic efficient $\mathcal{E}$ s.t. $\forall$ 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
3. Special Honest Verifier Zero Knowledge: $\exists$ deterministic efficient $Sim(x,c) \to (t,z)$ s.t.
   - $\forall (x,w) \in R$ $\quad \{(t,c,z) : c \xleftarrow{\$} C;\ t,z \leftarrow Sim(x,c)\} = \{\langle P(x,w), V(x)\rangle = 1\}$
   - $\forall x, \forall c$ $\quad t,z \leftarrow Sim(x,c) \to (t,c,z)$ is an accepting transcript

Notice that V $\quad$ 1) sends only random values to P
$\qquad\qquad\qquad\quad$ 2) has no secret state

we call this "public coin"

Made with Goodnotes

Fiat-Shamir Idea: replace verifier's message with the random oracle $\Rightarrow c \leftarrow H(x,t) \in \mathbb{Z}_q$

Schnorr (Prove knowledge of w s.t. $h = g^w$)

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

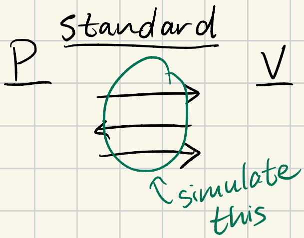
Schnorr-FS (Prove knowledge of w s.t. $h = g^w$ non-interactively)

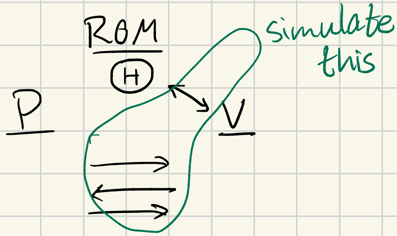
$P(g,h,w)$		$V(g,h)$
$r \xleftarrow{\$} \mathbb{Z}_p$	
$c \leftarrow H(g,h,g^r)$	
$z \leftarrow cw + r$	$\xrightarrow{(g^r,\, z)}$	$c := H(g,h,g^r)$
		$g^z \stackrel{?}{=} h^c g^r$

Analysis

1. Completeness is direct
2. ZK - follows from HVZK of underlying Σ-protocol $\rightarrow$ RO behaves like honest verifier!

Q: What does ZK mean in ROM?





A: Simulate $P \leftrightarrow V$ transcript + RO queries

called "programming" the RO

Sim:
- map $M: \mathbb{G}^3 \rightarrow \mathbb{Z}_p$
- $c \xleftarrow{\$} \mathbb{Z}_p$
- $t, z \leftarrow \mathrm{Sim}_{\mathrm{Schnorr}}((g,h), c)$
- set $M[(g,h,t)] \leftarrow c$
- output (t, c, z)

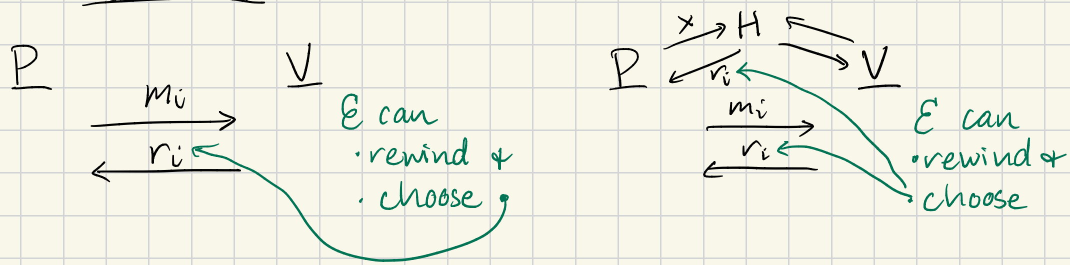
on RO query x: if $x \notin M$, set $M[x] \xleftarrow{\$} \mathbb{Z}_p$, output $M[x]$

Made with Goodnotes

3. PoK

Q: How do we prove PoK in ROM?

Standard | ROM



A: Ext behaves just like Σ-protocol extractor, except instead of rewinding & choosing V messages, extractor rewinds, chooses V messages, and reprograms RO for new challenge

Schnorr - FS Soundness

$\mathcal{E}$:

- $c \neq c' \xleftarrow{\$} C$
- run P^*: when it queries RO for challenge, give c
- run P^*: when it queries RO for challenge, give c'
- use 2 transcripts & $\mathcal{E}_{Schnorr}$ to get witness

Bonus: Signatures

- simply add m to the hash and let $pk = g^{sk}$
- $H: \mathbb{G}^3 \times \mathcal{M} \to \mathbb{Z}_p$

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):

$$c \leftarrow H(pk, g, g^r, m)$$
$$g^z \stackrel{?}{=} pk^c \cdot g^r$$

Made with Goodnotes

Notes

- in this specific case, don't need pk in hash
- could send c, not g^r, and compute $g^r \leftarrow g^z / pk^c$ and check $c \stackrel{?}{=} H(pk, g, g^r, m)$
- soundness error is $1/|c|$ → so c can be 128 bits
- z is in $\mathbb{Z}_q$, which would be 256 bits for EC group
  → total size of signature = 128 + 256 = 384 bits

Compare:

RSA-FDH ≈ 3072 bits

BLS: 384 bits (pairing group size)

In practice, ECDSA signatures are widely used; same idea as Schnorr but worse; why is it used? Patents!

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!)

Made with Goodnotes