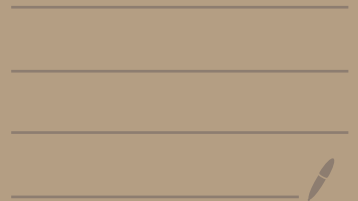


lec 08: Sigma Protocols

also: ZK-PoKs for NP

- 1) Motivation: A ZK-PoK for ϕ_{ID}
- 2) Sigma protocols ($\Sigma\Pi$ s)
- 3) $\Sigma\Pi$ combinators
- 4) $\Sigma\Pi$ for circuit SAT



0 Recap: ZK-PoK

- Consider an IP (P, V) for ϕ .
- Completeness: $\forall x \in \mathcal{L}_\phi, \Pr[\langle P, V \rangle(x; \omega) = 0] \leq \text{negl}$
- (Existential) Soundness: $\forall x \notin \mathcal{L}_\phi, \forall P^*, \Pr[\langle P^*, V \rangle(x) = 1] \leq \text{negl}$
- Knowledge Soundness: $\exists \text{PPT } E, \forall x, P^*, \Pr[\phi(x, \omega) = 1 : \omega \leftarrow E^{P^*}(x)] \geq \Pr[\langle P^*, V \rangle(x) = 1] - \text{negl}$
 $\uparrow$
 E can rewind P^* w/ same randomness
- HVZK: $\exists \text{PPT Sim}, \forall x \in \mathcal{L}_\phi, \{\text{Sim}(x)\} \approx_c \{\text{view}_V[\langle P, V \rangle(x; \omega)]\}$
- $\phi_{\text{dlog}} (h \in G, \omega \in \mathbb{Z}_p) \stackrel{?}{=} g^\omega = h$
 - soundness is trivial: $\mathcal{L}_{\phi_{\text{dlog}}} = G$
 - (all $h \in G$ have some d-log...)
- Last time: a ZK-PoK for ϕ_{dlog} : Schnorr

• E got

$\frac{P(g, h, \omega)}{r \in \mathbb{Z}_p}$	$\xrightarrow{u = g^r}$	$\frac{V(g, h)}{c \in \mathbb{Z}_p}$
	$\xleftarrow{c}$	
	$\xrightarrow{z \leftarrow r + c\omega}$	$g^z \stackrel{?}{=} u \cdot h^c$

$u, c_1, z_1 \rightarrow$
 $u, c_2, z_2 \rightarrow$ found ω assuming $c_1 \neq c_2$!

[1] Motivation: ZK-PoK for circuits.

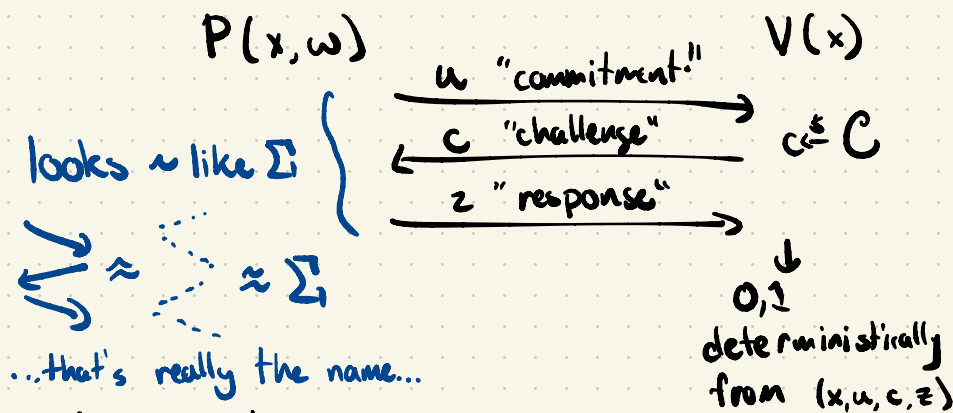
- Recall the PrivateID ϕ_{ID}

$$\phi_{ID}((photo, age, pk_{GA}), (id, \sigma)) \\ \triangleq (photo, age) \in id \wedge \text{SigVer}(pk_{GA}, id, \sigma)$$

- Q: Is $\mathcal{L}_{\phi_{ID}}$ trivial? (like ϕ_{alog})
 - For fixed pk_{GA} , does it contain all $(photo, age)$?
- A: Yes!
 - for any $photo, age$,
 - some id contains them, and
 - $\sigma \leftarrow \text{Sign}(sk_{GA}, id)$ is a signature on it!
 - so, ϕ_{ID} needs knowledge soundness.
- Today: a ZK-PoK for all circuits, ϕ_{ID} included...

[2] Sigma Protocols.

- To achieve a ZK-PoK for SAT, we'll abstract
- We'll define a subset of ZK-PoKs that are "nice"
"nice": easy to prove secure + combine
- Our name for the abstraction is " Σ protocol"
- Defⁿ A Σ Protocol is an IP for $\phi(x, w)$ with 3 moves:



- It must have 3 security properties:
 - Perfect Completeness
 - Special Soundness:
 $\exists$ deterministic, poly $\mathcal{E}$ that $\forall x, \forall$ pairs of accepting transcripts $(u, c_1, z_1), (u, c_2, z_2)$ with $c_1 \neq c_2$, outputs w s.t. $\phi(x, w) = 1$.
 - this implies PoK
 - The proof is implicit in our proof that Schnorr is a PoK.
- Special HVZK:

$\exists$ PPT $\text{Sim}(x, c) \rightarrow (u, z)$ s.t.

$\forall x \quad (u, c, z)$ is accepting and

$$\{(u, c, z): \begin{matrix} c \in C \\ u, z \leftarrow \text{Sim}(x, c) \end{matrix}\} \approx \{\text{view}_V[(P, V)(x, w)]\}$$

• this implies HVZK, of course.

• Claim: Schnorr is a Σ_1 protocol.

• special soundness: this is how Schnorr's E works.

• special HVZK:

• $\text{Sim}_{\text{Schnorr}}$ samples $c \xleftarrow{\$} \mathbb{Z}_p$

• It also always outputs accepting transcripts

• In general:

• Special soundness $\Leftarrow$ knowledge soundness

• for KS, E may need more transcripts, fail some times, require assumptions, etc.

• Special HVZK $\Leftarrow$ HVZK

• for HVZK, Sim output some rejecting transcripts

• The special properties assume more structure

• SS: E is deterministic

• SHVZK: transcript is always accepting

• less is left to chance ...

• this helps us define $\Sigma \Pi$ combinators.

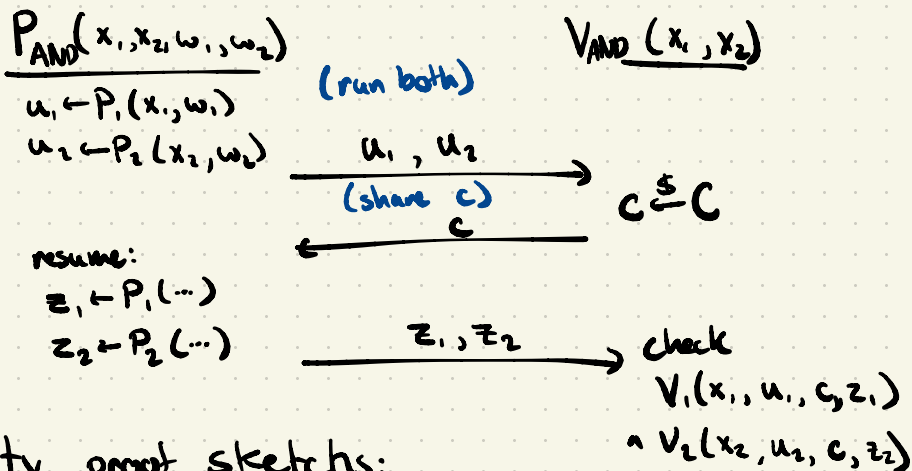
3 $\Sigma\Pi$ combinators.

Let $\phi_1(x_1, \omega_1)$ and $\phi_2(x_2, \omega_2)$ be relations with $\Sigma\Pi_s$ (P_1, V_1) and (P_2, V_2) .

• AND combinator:

• $\phi_{AND}(x_1, x_2, (\omega_1, \omega_2)) \stackrel{\Delta}{=} \phi_1(x_1, \omega_1) \wedge \phi_2(x_2, \omega_2)$

• We'll give a $\Sigma\Pi$ (P_{AND}, V_{AND}) for ϕ_{AND}



Security proof sketches:

• Special soundness:

• $\mathcal{E}_{AND}$: given $((u_1, u_2), c, (z_1, z_2))$ and $((u_1, u_2), c', (z'_1, z'_2))$ with $c \neq c'$, run

$\omega_1 \leftarrow \mathcal{E}_1((u_1, c, z_1), (u_1, c', z'_1))$

$\omega_2 \leftarrow \mathcal{E}_2((u_2, c, z_2), (u_2, c', z'_2))$

output (ω_1, ω_2)

• you can show $c \neq c' \Rightarrow \phi_{AND}(x_1, x_2, (\omega_1, \omega_2)) = 1$

• SHVZK:

• $\text{Sim}_\Lambda(x_1, x_2, c)$: $\uparrow$ SS's definition is used here:
 $(u_1, z_1) \leftarrow \text{Sim}_1(x_1, c)$ $c \neq c'$ applies to both $\mathcal{E}_1$ and $\mathcal{E}_2$.

$$(u_2, z_2) \leftarrow \text{Sim}(x_2, c)$$

$$\text{output } ((u_1, u_2), (z_1, z_2))$$

- You can show output is accepting and x_c from real.

• OR combinator

- Let $\phi_{\text{or}}((x_0, x_1), (b, \omega_b)) \stackrel{\Delta}{=} \phi(x_b, \omega_b)$ which ϕ_b holds
- Assume C is a group w/ $(+)$ its witness

$$P_{\text{or}}(x_0, x_1, b, \omega_b) \quad \forall (x_0, x_1)$$

idea: run P_b , simulate $P_{\bar{b}}$

$$\text{let } \bar{b} = b \oplus 1$$

$$c_{\bar{b}} \stackrel{\$}{\leftarrow} C$$

$$(u_{\bar{b}}, z_{\bar{b}}) \leftarrow \text{Sim}_{\bar{b}}(x_{\bar{b}}, c_{\bar{b}})$$

$$u_b \leftarrow P_1(x_b, \omega_b)$$

$$\begin{array}{c} \text{order } \perp b \\ \downarrow \\ (u_0, u_1) \xrightarrow{\quad} c \stackrel{\$}{\leftarrow} C \\ \xleftarrow{\quad} c \end{array}$$

$$c_b \leftarrow C - c_{\bar{b}}$$

$$z_b \leftarrow P_1(x_b, \omega_b, c)$$

$$\xrightarrow{c_0, c_1, z_0, z_1}$$

check:

$$\begin{array}{l} \text{both} \\ \text{valid} \end{array} \left[\begin{array}{l} V_1(x_1, u_1, c_1, z_1) \\ V_2(x_2, u_2, c_2, z_2) \end{array} \right]$$

$$\underline{c = c_1 + c_2}$$

P can choose 1 c_i , but not both, since V chose C !

• Proof sketches:

• Perfect completeness:

Sim_b 's output is always accepting¹.

• SS:

(x_0, x_1)
 $\text{EOR}((u_0, u_1), c, (c_0, c_1, z_0, z_1), c', (c'_0, c'_1, z'_0, z'_1))$: need $c_i \neq c'_i$ for some $i \in \{0, 1\}$

$$\begin{aligned} \text{since } c &= c_0 + c_1 \\ c' &= c'_0 + c'_1 \\ c &\neq c', \end{aligned}$$

then $c_0 \neq c'_0$ or $c_1 \neq c'_1$

if $c_0 \neq c'_0$: output $(0, \text{E}_0(x_0, (u_0, c_0, z_0), (u_0, c'_0, z'_0)))$

if $c_1 \neq c'_1$: output $(1, \text{E}_1(x_1, (u_1, c_1, z_1), (u_1, c'_1, z'_1)))$

• SHVZK:

• $\text{Sim}_{\text{OR}}((x_0, x_1), c)$

• $c_0 \xleftarrow{\$} C, c_1 \leftarrow c - c_0$

• $(u_0, z_0) \leftarrow \text{Sim}_0(x_0, c_0)$

• $(u_1, z_1) \leftarrow \text{Sim}_1(x_1, c_1)$

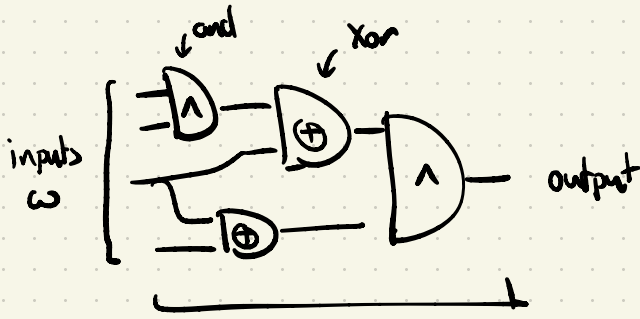
• output $((u_0, u_1), (z_0, z_1))$

• analysis: $\approx_c$

• C, C_1, C_2 are distributed perfectly

• for the rest: a hybrid argument
relying on $\approx_c$ for $\text{Sim}_0, \text{Sim}_1$

4) A $\Sigma\Pi$ for circuit SAT

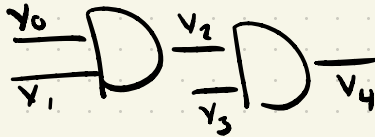


circuit $C(\omega) \rightarrow \{0,1\}$

$$\phi_{\text{SAT}}(C, \omega) \triangleq C(\omega) \stackrel{?}{=} 1$$

How can we make a $\Sigma\Pi$ for ϕ_{SAT} ?

- P evaluates $C(\omega)$ to get wire values $v_i \in \{0,1\}$



- P commits to v_i as c_i
 - For example, with Pedersen: $g^{v_i} h^{r_i}$, $r_i \in \mathbb{Z}_p$
- P shows:
 - output is 1 ($v_4 = 1$)
 - want a $\Sigma\Pi$ for $\phi(b, c, r) \triangleq c = \text{Ped.Com}(b, r)$
 - we'll come back to this
 - each gate's in/out commitments are to values in the gate's truth table

$$v_i \stackrel{?}{=} \bigwedge_j v_k \quad (\text{public: } c_i, c_j, c_k)$$

V_i	V_j	V_k
0	0	0
0	1	0
1	0	0
1	1	1

so, we want:

$$V_i = 0 \text{ AND } V_j = 0 \text{ AND } V_k = 0$$

OR

$$V_i = 0 \text{ AND } V_j = 1 \text{ AND } V_k = 0$$

OR

$$V_i = 1 \text{ AND } V_j = 0 \text{ AND } V_k = 0$$

OR

$$V_i = 1 \text{ AND } V_j = 1 \text{ AND } V_k = 1$$

If we can show " $v_i = b$ " for any i, b ,
we can do via $\Sigma \Pi$ AND/OR combinators...

So: " $v_i = b$ "

Formally: $\phi(b, c, r) \stackrel{\Delta}{=} C = \text{Ped.Com}(b, r)$
 $C = g^b h^r$

Schnorr's protocol can actually do this!

Let's see how

case $b=0$:

$$C = g^0 h^r$$

$$\underline{C = h^r}$$

Schnorr protocol with:

- base h
- $d \log r$
- result C

case $b=1$: $C = g^1 h^r$

$$\underline{C/g = h^r}$$

Schnorr again

- base h
- $d \log r$
- result C/g

• So:

- Schnorr gives a $\Sigma \Pi$ for wire values
- AND combinator gives a $\Sigma \Pi$ for a truth table row
- OR combinator gives a $\Sigma \Pi$ for a truth table
- AND combinator gives a $\Sigma \Pi$ for ϕ_{SAT} :
 - another Schnorr for "output=1"
 - a big AND for all the gates (truth tables)
 - the initial wire c_i 's are in the $\Sigma \Pi$'s 1st move.

• Remarks:

- This is a BIG protocol
 - for n gates: $12n+1$ Schnorr sub-Ms!
- Still just one c !
- Very sound! error $\approx (12n+1)/|G|$
- Efficient enough to run in $\sim 1-2$ minutes over the AES circuit.
 - Not as good as FAEST signatures:
 - ≈ 0.5 ms for a ZKP(AES)
 - Much better than Π_3 -color
 - > 100 years