Zero-Knowledge Protocols

Review: NP problems

Def. $L \in NP$ if $L \in \{0,1\}^*$

and $\exists$ poly-time M s.t.

$x \in L \iff \exists w \in \{0,1\}^{n} : M(x,w) = 1$

x: statement
w: witness

e.g. equality of $DLog$

G: cyclic group of prime order q

$L_{EDL} = \{ (g, h, g^{\alpha}, h^{\alpha}) \in G^4 : \begin{matrix} \alpha \in \mathbb{Z}_q \\ h, g \neq 1 \end{matrix} \}$

Given witness α, easy to check $(g,h,u,v) \in L_{EDL}$

Zero Knowledge Proof System for $L \in NP$

Def. Proof System: pair of prob. poly time (PPT) algs.

P, V

$P(x, w) \longrightarrow V(x)$

$\longleftarrow$

$\longrightarrow$

$V(x) \hookrightarrow$ yes or no

s.t.

1. Complete: $\forall x, w$: if $M(x,w) = 1$ (i.e. $x \in L$):
   then $Pr[(P(x,w) \leftrightarrow V(x)) = \text{yes}] = 1$
2. Sound: $\forall x \notin L$, $\forall \hat{P}$: $Pr[(\hat{P} \leftrightarrow V(x)) = \text{yes}] \leq \text{negl}$.
   ↓all, incl exp. time, provers
   (cheating prover cannot convince verifier $x \in L$)

Trivial ex:

$P \xrightarrow{w} V$

V accepts if $R(x, w) = 1$

Honest verifier zero knowledge (HVZK):

Protocol should reveal nothing except that $x \in L$

For x, w, let $\text{transcript}(P(x,w) \leftrightarrow V(x))$ be seq. of msgs between $P(x,w)$ and $V(x)$ (random vars).

Def. (P, V) is HVZK for L if

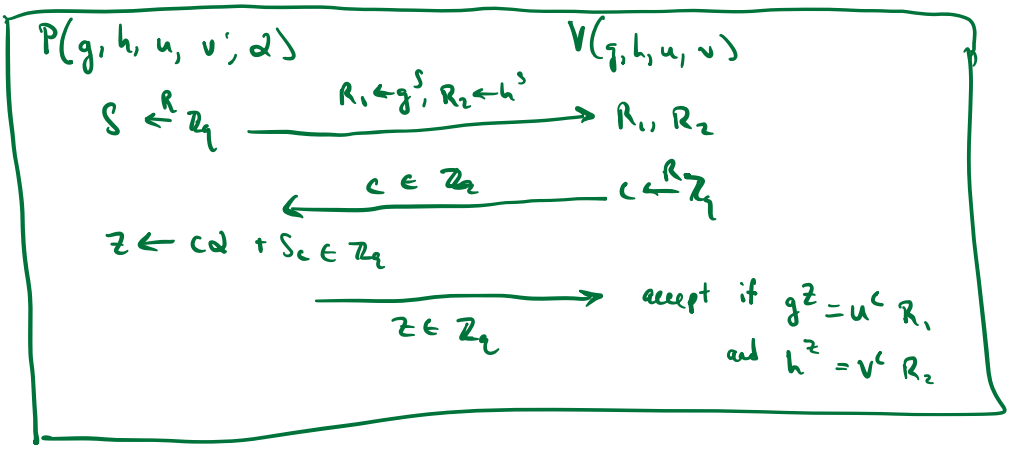
$\exists$ PPT alg S. (simulator) s.t. $\forall x \in L$:

Distr. $\{S(x)\}$ is computationally indistinguishable

from Distr. $\{\text{transcript}(P(x,w) \leftrightarrow V(x))\}$

$\Rightarrow$ Sim. shows V learns nothing from transcript b/c it can gen. transcript on its own.

Language example: HVZK proof system for L_{EDL}



note: verifier has no secrets → public coin protocol

Proof completeness: $(g, h, u = g^\alpha, v = h^\alpha) \in L_{EDL}$:

$\left\{ \begin{array}{l} g^z = g^{c\alpha + s} = (g^\alpha)^c g^s = u^c R_1 \\ \text{same for } h^z \end{array} \right.$

$\Rightarrow$ V outputs yes. ✓

Soundness: Mal. prover $\hat{P}$

Statement $u = g^\alpha$, $v = h^\beta$, $\alpha \neq \beta$

$\Rightarrow (g, h, u, v) \notin LEDL$ ← desired outcome

transcript = $(R_1 = g^{s_1}, R_2 = h^{s_2}, c, z)$

$Pr[V \text{ acc.}] = Pr[z = \alpha c + s_1, \; z = \beta c + s_2]$

$= Pr[\alpha c + s_1 = \beta c + s_2] = Pr\left[c = \frac{s_1 - s_2}{\beta - \alpha}\right] = \frac{1}{q} \leq \text{negl.}$ ✓

HVZK: if $(g, h, u, v) \in LEDL$,

then Sim needs to output (R_1, R_2, c, z)

Sim (g, h, u, v) does:

1. choose $c, z \xleftarrow{R} \mathbb{Z}_q$
2. Set $R_1 \leftarrow g^z / u^c$, $R_2 \leftarrow h^z / v^c$
3. output (R_1, R_2, c, z)

Then: $R_1 = g^s$ and $R_2 = h^s$ where $s = z - c\alpha$

c uniform in $\mathbb{Z}_q$

$z \in \mathbb{Z}_q$ s.t. cond (1), (2) hold

Same as transcript! ✓

Soundness vs. Knowledge

ZK proof proves to V that

$\exists w$ s.t. $M(x, w) = 1$ (i.e. $x \in L$)

What if V wants a proof that P knows a witness w?

⇒ ZK Proof of Knowledge (ZKPK systems)

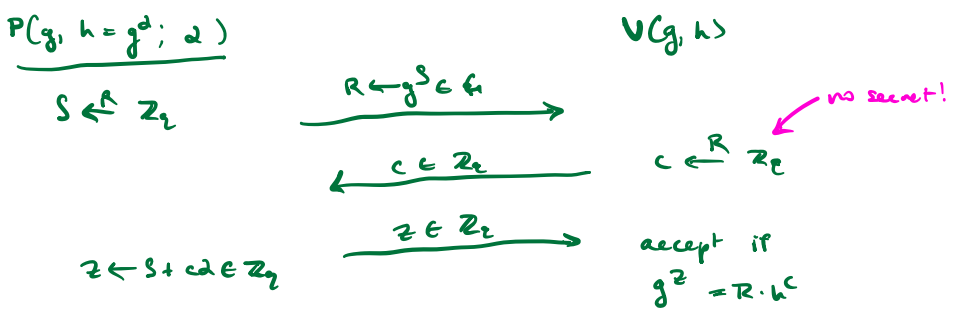
1. complete
2. HVZK
3. Can "extract" a witness w from $\hat{P}$.

ex. ZKPK of DLog

Let G be an FCG of prime order q w/gen $g \in G$

$$L = \{(g, h = g^a) : a \in \mathbb{Z}_q, g \neq 1\} \subseteq G^2$$

Schnorr proof of knowledge for DLOG



note: public coin protocol.

Thm. Schnorr is complete, HVZK, and knowledge sound.

Proof. 1. complete: easy. $\in G$ $\in \mathbb{Z}_q$

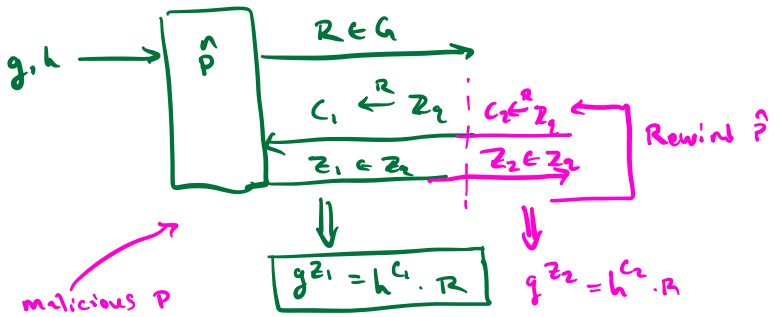
2. HVZK: transcript = (R, c, z) where $g^z = h^c \cdot R$

Sim(g, h) works as follows:

1. $c, z \xleftarrow{R} \mathbb{Z}_q$
2. $R \leftarrow g^z / h^c$
3. output (R, c, z)

⇒ honest V learns nothing new from protocol

3. extractor: extract α from malicious $\hat{P}$



divide one relation by the other (R cancels)

$$g^{z_1 - z_2} = h^{c_1 - c_2}$$

if $c_1 \neq c_2$ (v. high prob): raise both sides to power of $\frac{1}{c_1 - c_2} \in Z_q$

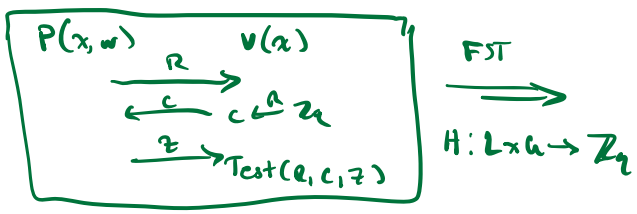
$$\Rightarrow g^{(z_1 - z_2)/(c_1 - c_2)} = h$$

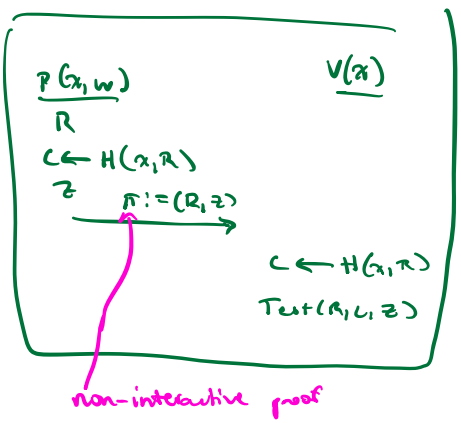
$$\boxed{\alpha = \frac{z_1 - z_2}{c_1 - c_2} \in Z_q}$$

Public Coin Protocol

Can be made non-interactive

Fiat-Shamir Transform:





public coin $\mathcal{Z}_{KPK} \rightarrow$ dig. sig. scheme
(Schnorr. sigs)

$c \leftarrow H(x, \underline{m}, R)$

need: $H : G \times M \times G \rightarrow \mathbb{Z}_q$

1. KeyGen(): $\alpha \xleftarrow{R} \mathbb{Z}_q$, $h \leftarrow g^\alpha \in G$

SK: α, PK: h

2. $S(sk = \alpha, m)$: $s \xleftarrow{R} \mathbb{Z}_q$, $R \leftarrow g^s \in G$

$c \leftarrow H(h, m, R) \in \mathbb{Z}_q$

prover gets own challenge

$z \leftarrow s + c \cdot \alpha \in \mathbb{Z}_q$

output: $\sigma := (c, z)$

3. $V(pk = h, m, \sigma = (c, z))$:

accept if $H(h, m, g^z / h^c) \stackrel{?}{=} c$

R

Thm. (Gen, S, V) secure assuming DLog hard in G and H is modeled as a "random oracle."