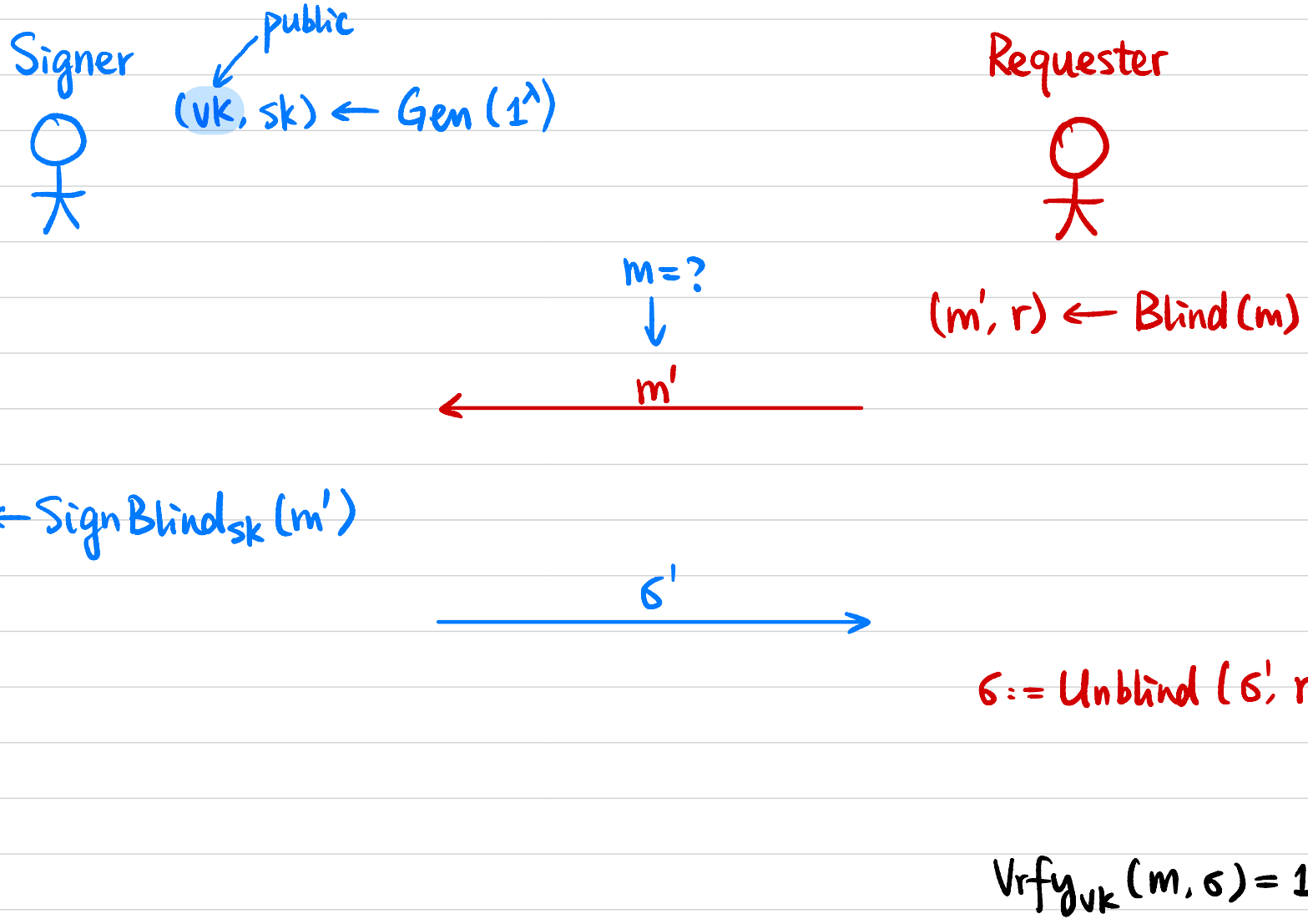
CSCI 1515 Applied Cryptography

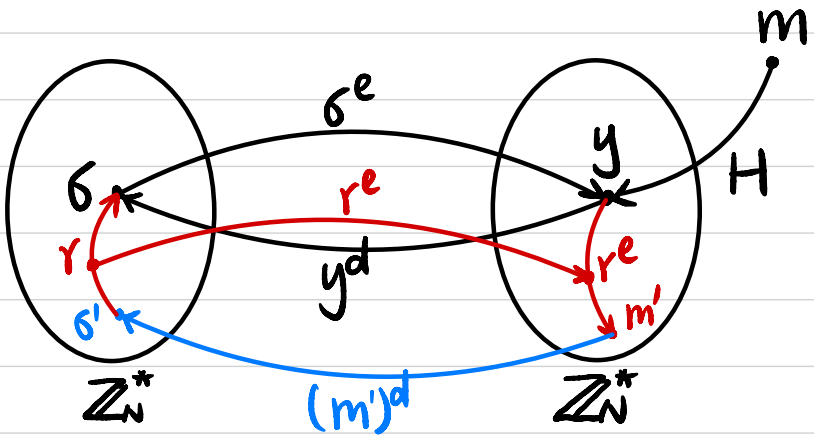
This Lecture:

- RSA Blind Signature (Continued)
- Putting it All Together: Anonymous Online Voting
- More Examples of Sigma Protocols
- Zero-Knowledge Proofs for All NP

Blind Signature



RSA Blind Signature



$VK = (N, e)$ $\quad$ $Sk = d$

$Sign_{sk}(m) = H(m)^d \bmod N$

$Vrfy_{VK}(m, \sigma)$: $\sigma^e \stackrel{?}{\equiv} H(m) \pmod N$

Signer

$(VK, sk) \leftarrow Gen(1^\lambda)$

$m = ?$

$\downarrow$

m' $\longleftarrow$

$SignBlind_{sk}(m')$:

$\sigma' := (m')^d$

σ' $\longrightarrow$

Requester

$Blind(m)$:

$r \xleftarrow{\$} \mathbb{Z}_N^*$

$m' := H(m) \cdot r^e \bmod N$

$= (H(m) \cdot r^e)^d$

$= H(m)^d \cdot r^{ed}$

$= H(m)^d \cdot r$

$Unblind(\sigma', r)$:

$\sigma := \sigma' \cdot r^{-1} \bmod N$

Anonymous Online Voting

$(g^{r_i},\ pk^{r_i} \cdot g^{v_i})$

ElGamal — Zkp

Voter 1 ⟶ $Enc(v_1)$ $\quad v_1 \in \{0,1\}$

Voter 2 ⟶ $Enc(v_2)$ $\quad v_2 \in \{0,1\}$

⋮

Voter n ⟶ $Enc(v_n)$ $\quad v_n \in \{0,1\}$

⇓

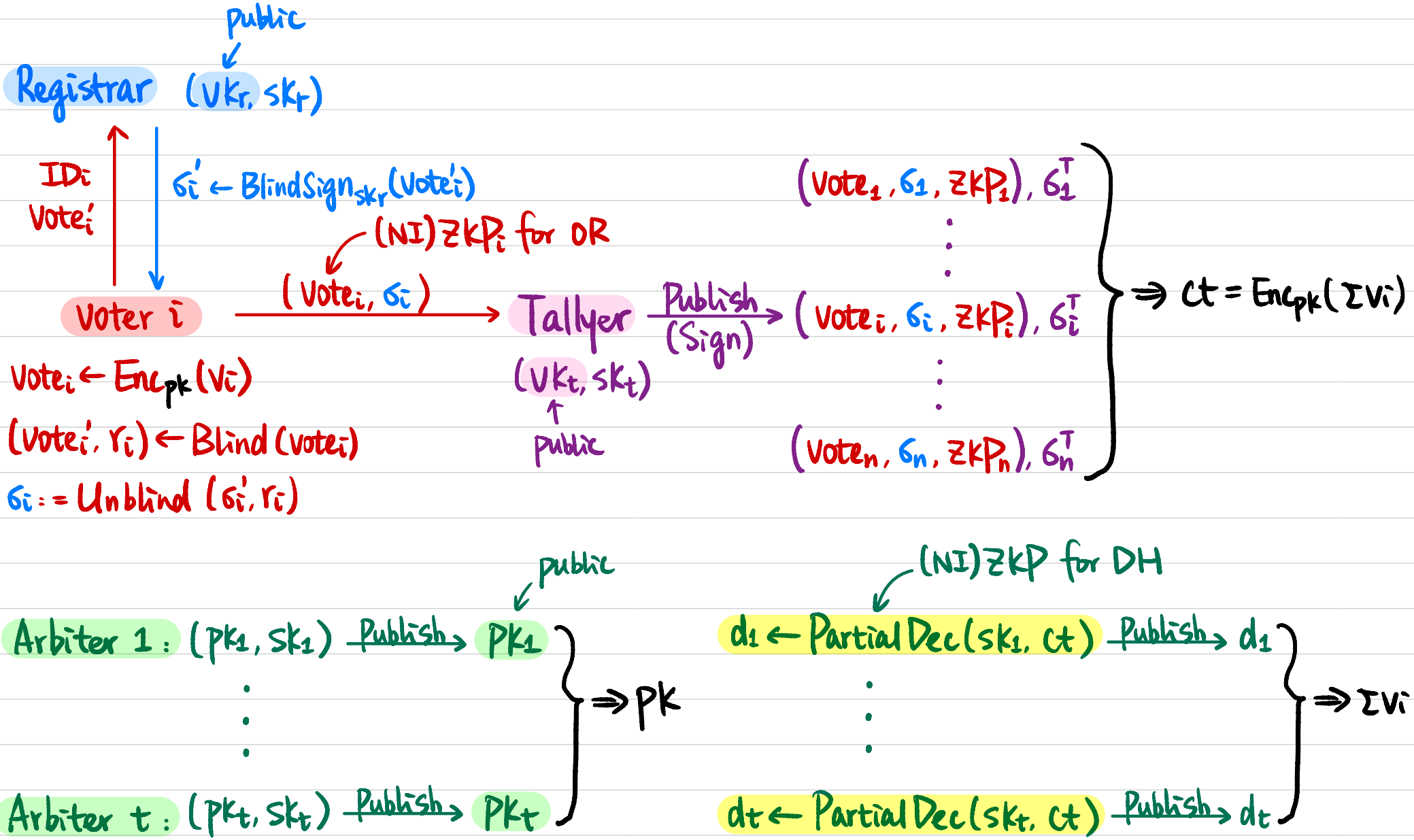
$Enc(\Sigma v_i)$ $\quad (g^{\Sigma r_i},\ pk^{\Sigma r_i} \cdot g^{\Sigma v_i})$

⇓

Decrypt to Σv_i

Threshold

Putting it All Together: Anonymous Online Voting



Multiple Candidates ?

k candidates

Voter 1 $\longrightarrow$ Enc (V_1) $\quad V_1 \in \{0, 1, \cdots, k-1\}$

Voter 2 $\longrightarrow$ Enc (V_2) $\quad V_2 \in \{0, 1, \cdots, k-1\}$

$\vdots$

Voter n $\longrightarrow$ Enc (V_n) $\quad V_n \in \{0, 1, \cdots, k-1\}$

$\Downarrow$

Enc (ΣV_i)

$\Downarrow$

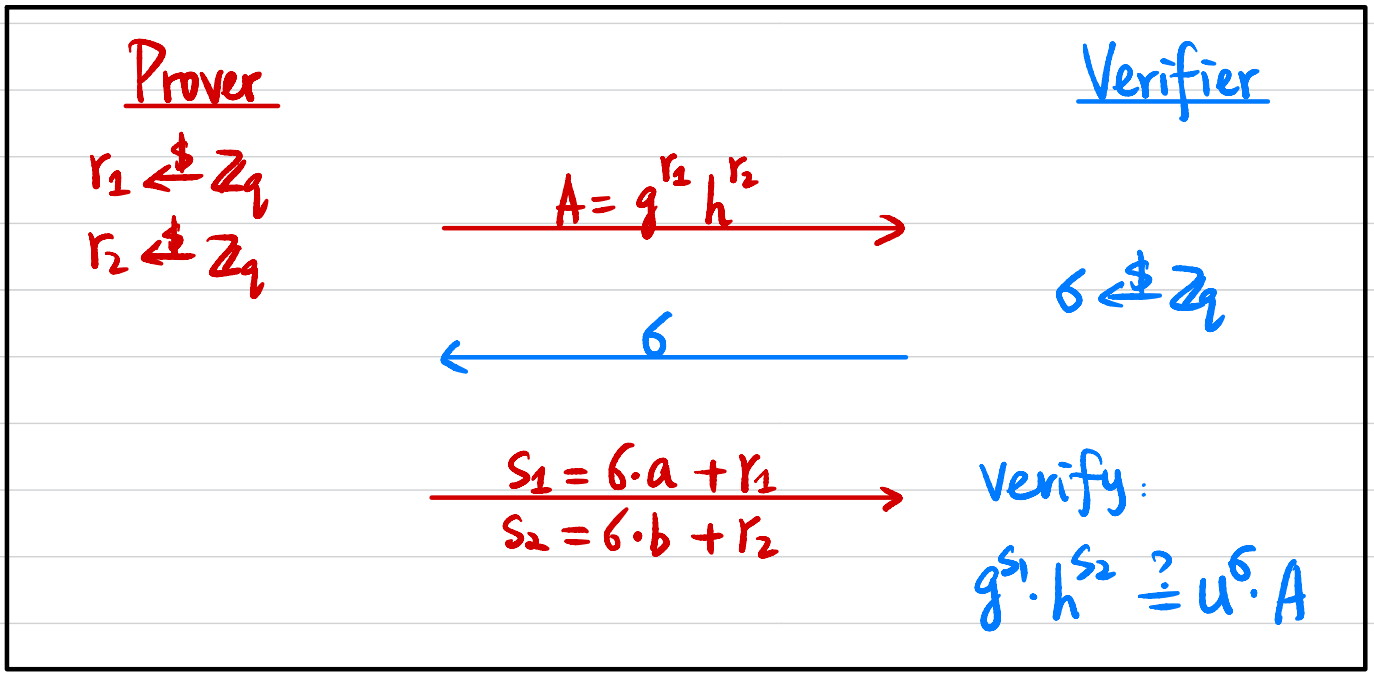
Decrypt to ΣV_i

Example: Okamoto's Protocol for Representation

Input: Cyclic group G of order q, generator g, h, u

Witness: (a, b)

$R_L = \{ ((h, u), (a, b)) : u = g^a h^b \}$



Completeness?

Proof of Knowledge (PoK)?

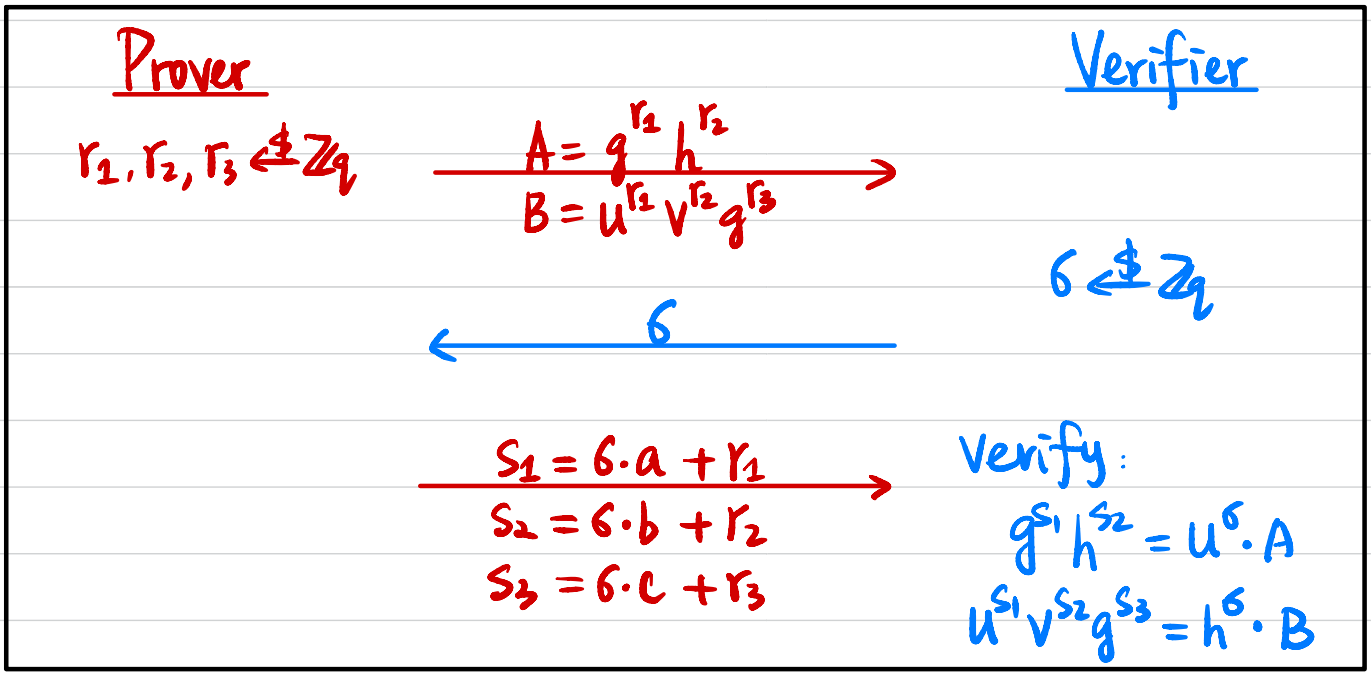
Honest-Verifier Zero-Knowledge (HVZK)?

Example: Arbitrary Linear Equations

Input: Cyclic group G of order q, generator g, h, u, v

Witness: (a, b, c)

$R_L = \{ ((h, u, v), (a, b, c)) : u = g^a h^b \wedge h = u^a v^b g^c \}$

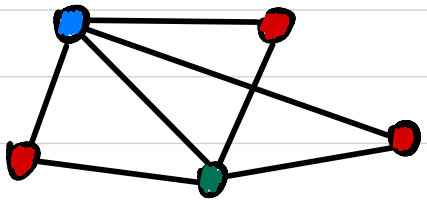


Completeness?

Proof of Knowledge (PoK)?

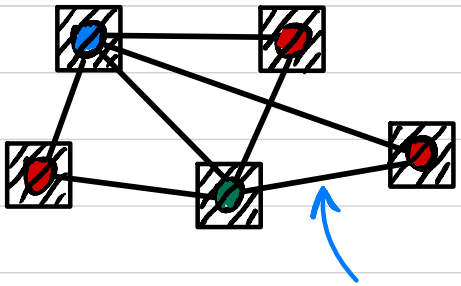
Honest-Verifier Zero-Knowledge (HVZK)?

Zero-Knowledge Proof for Graph 3-Coloring (All NP)



NP language $L = \{ G : G \text{ has 3-coloring} \}$

NP relation $R_L = \{ (G, 3COL) \}$



If $G \notin L$, $\Pr[P^* \text{ is caught}] \geq ?$

How to amplify soundness?

Commitment Scheme

Sender		Receiver
$m \in \{0,1\}$		
Commit: $r \xleftarrow{\$} \{0,1\}^\lambda$, $c := Com(m;r)$	$\xrightarrow{c}$	
Open:	$\xrightarrow{(m,r)}$	Verify: $c = Com(m;r)$

- **Hiding:** $Com(0;r) \simeq Com(1;s)$
- **Binding:** Hard to find r, s s.t. $Com(0;r) = Com(1;s)$

Commitment Scheme

Example 1: Hash-based commitment

$r \xleftarrow{\$} \{0,1\}^{\lambda}$

$\mathrm{Com}(m; r) := H(r \| m)$

↑ Random Oracle

Example 2: Pedersen Commitment

Cyclic group G of order q with generator g. $h \xleftarrow{\$} G$

↑ can be generated by Receiver

$h = g^x$. x hidden to Sender

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

$\mathrm{Com}(m; r) = g^m \cdot h^r$

Why are the schemes hiding & binding?

Zero-Knowledge Proof for Graph 3-Coloring

Input: $G = (V, E)$

Witness: $\phi: V \to \{0,1,2\}$

Prover

Randomly sample $\pi: \{0,1,2\} \to \{0,1,2\}$

$\forall v \in V,\ r_v \xleftarrow{R} \{0,1\}^\lambda,\ C_v := \mathrm{Com}(\pi(\phi(v)); r_v)$

$\{C_v\}_{v \in V}$ →

Verifier

Randomly pick an edge $(u,v) \in E$

← (u,v)

Open commitments C_u & C_v

$\alpha = \pi(\phi(u)),\ r_u$

$\beta = \pi(\phi(v)),\ r_v$ →

Verify: $C_u = \mathrm{Com}(\alpha; r_u)$

$C_v = \mathrm{Com}(\beta; r_v)$

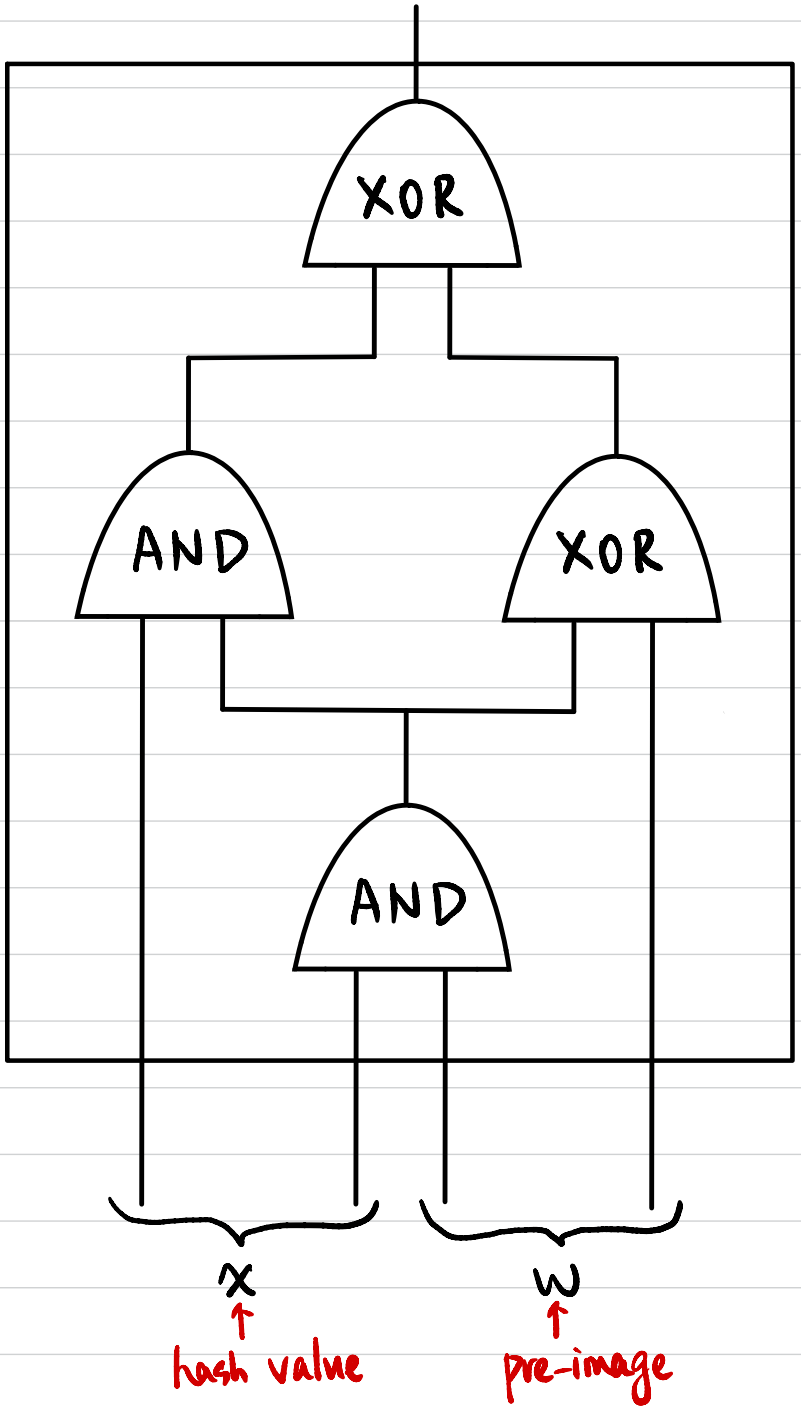
$\alpha, \beta \in \{0,1,2\},\ \alpha \neq \beta$

Completeness?

Soundness?

Zero-Knowledge?

Circuit Satisfiability (NP Complete)



NP language $L_C = \{x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ s.t. } C(x,w) = 1\}$

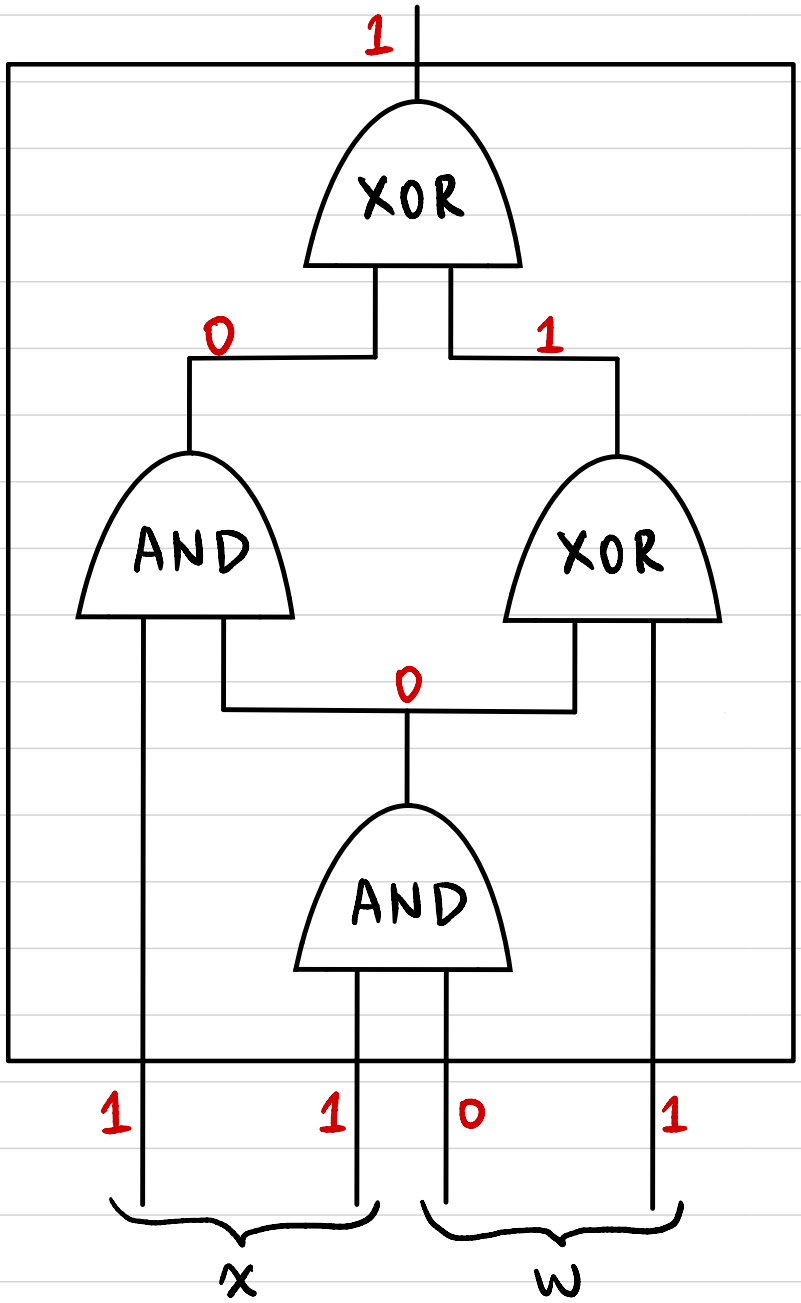
NP relation $R_{L_C} = \{(x,w) : C(x,w) = 1\}$

Example: pre-image of hash function

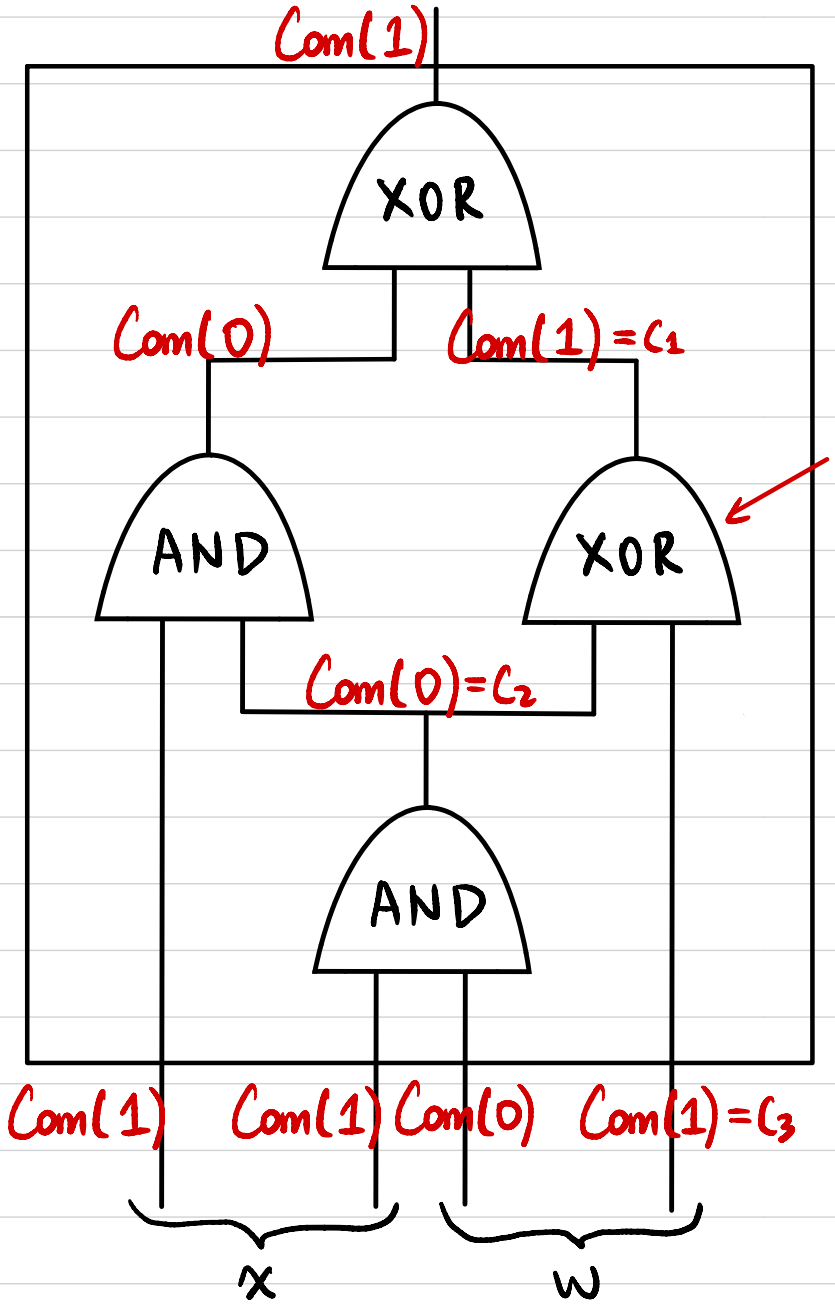
$C(x,w) = H(w) - x + 1$

$H(w) = x$

ZKP for Circuit Satisfiability



ZKP for Circuit Satisfiability



$$\begin{pmatrix} C_1 = Com(0) \\ C_2 = Com(0) \\ C_3 = Com(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_1 = Com(1) \\ C_2 = Com(0) \\ C_3 = Com(1) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_1 = Com(1) \\ C_2 = Com(1) \\ C_3 = Com(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} C_1 = Com(0) \\ C_2 = Com(1) \\ C_3 = Com(1) \end{pmatrix}$$

Proof Systems for Circuit Satisfiability

NP relation $R_{\mathcal{C}} = \{(x, w) : C(x, w) = 1\}$

	NP	Σ-Protocol	(Fiat-Shamir) NIZK
	$P(x, w) \xrightarrow{w} V(x)$ $C(x,w)=1$	$P(x, w) \rightleftarrows V(x)$	$P(x, w) \xrightarrow{\pi} V(x)$
Zero-Knowledge	NO	YES	YES
Non-Interactive	YES	NO	YES
Communication	$O(\|w\|)$	$O(\|C\| \cdot \lambda)$	$O(\|C\| \cdot \lambda)$
V's computation	$O(\|C\|)$	$O(\|C\| \cdot \lambda)$	$O(\|C\| \cdot \lambda)$

Can we have communication complexity & verifier's computational complexity sublinear in $|C|$ & $|w|$?