CSCI 1515 Applied Cryptography

This Lecture:

- Putting it All Together: Anonymous Online Voting
- Zero-Knowledge Proofs for All NP
- Succinct Non-Interactive Arguments (SNARGs)

Putting it All Together: Anonymous Online Voting

public

Registrar (vk_r, sk_r)

ID_i, $Vote_i'$

Step 4: $\sigma_i' \leftarrow BlindSign_{sk_r}(Vote_i')$

$\boxed{\sigma_i' := (Vote_i')^d \bmod N}$

Voter i $\xrightarrow{(Vote_i,\ \sigma_i,\ \pi_i)}$ Tallyer

Step 6: $(Vote_i, \sigma_i, \pi_i)$

$\boxed{\text{NIZK}_i \text{ for OR}}$

public

Tallyer (vk_t, sk_t) $\xrightarrow{\text{Publish}}$

$(Vote_1, \sigma_1, \pi_1), \sigma_1^T$

$\vdots$

$(Vote_i, \sigma_i, \pi_i), \sigma_i^T$

$\vdots$

$(Vote_n, \sigma_n, \pi_n), \sigma_n^T$

Step 7: $\boxed{\sigma_i^T \leftarrow Sign_{sk_t}(Vote_i, \sigma_i, \pi_i)}$

Step 8: $\Rightarrow \boxed{ct := \prod Vote_i}$

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

$= Enc_{pk}(\sum v_i)$

Step 2: $Vote_i \leftarrow Enc_{pk}(v_i)$

$\boxed{r_i \xleftarrow{\$} \mathbb{Z}_q,\ Vote_i = (g^{r_i}, pk^{r_i} \cdot g^{v_i})}$

Step 3: $(Vote_i', r_i') \leftarrow Blind(Vote_i)$

$\boxed{r_i' \xleftarrow{\$} \mathbb{Z}_N^*,\ Vote_i' := H(Vote_i) \cdot (r_i')^e \bmod N}$

Step 5: $\sigma_i := Unblind(\sigma_i', r_i')$

$\boxed{\sigma_i := \sigma_i' \cdot (r_i')^{-1} \bmod N}$

Step 1:

public

Arbiter 1: $(pk_1, sk_1) \xrightarrow{\text{Publish}} pk_1$

$\vdots$

Arbiter t: $(pk_t, sk_t) \xrightarrow{\text{Publish}} pk_t$

$\Rightarrow \boxed{pk := \prod pk_i}$

$\boxed{sk_i \xleftarrow{\$} \mathbb{Z}_q,\ pk_i := g^{sk_i} \xrightarrow{\text{Publish}} pk_i}$

Step 9:

$\boxed{\text{NIZK}_i \text{ for DH}}$

$d_1 \leftarrow PartialDec(sk_1, ct) \xrightarrow{\text{Publish}} (d_1, \pi_1^A)$

$\vdots$

$d_t \leftarrow PartialDec(sk_t, ct) \xrightarrow{\text{Publish}} (d_t, \pi_t^A)$

$\boxed{ct = (c_1, c_2),\ d_i := c_1^{sk_i}}$

Step 10:

$\Rightarrow \boxed{g^{\sum v_i} = c_2 / (\prod d_i)}$

$\Rightarrow \sum v_i$

Multiple Candidates?

k candidates (No limit on #candidates to vote for)

Public: Cyclic group G of order q with generator g

ElGamal public key pk

ZKP ($\in \{0,1\}$)

Candidate #1 #2 . . . #k

Voter 1 $\longrightarrow$ $Enc(v_1) = (g^{r_1}, pk^{r_1} \cdot g^{v_1})$ $Enc(v_1^1)$ $Enc(v_1^2)$ $\cdots$ $Enc(v_1^k)$

Voter 2 $\longrightarrow$ $Enc(v_2) = (g^{r_2}, pk^{r_2} \cdot g^{v_2})$ $Enc(v_2^1)$ $Enc(v_2^2)$ $\cdots$ $Enc(v_2^k)$

$\vdots$

Voter n $\longrightarrow$ $Enc(v_n) = (g^{r_n}, pk^{r_n} \cdot g^{v_n})$ $Enc(v_n^1)$ $Enc(v_n^2)$ $\cdots$ $Enc(v_n^k)$

$\Downarrow$

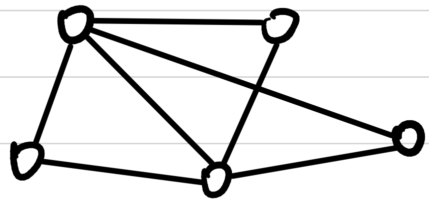
$Enc(\Sigma v_i^1)$ $Enc(\Sigma v_i^2)$ $\cdots$ $Enc(\Sigma v_i^k)$

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

$\Downarrow$

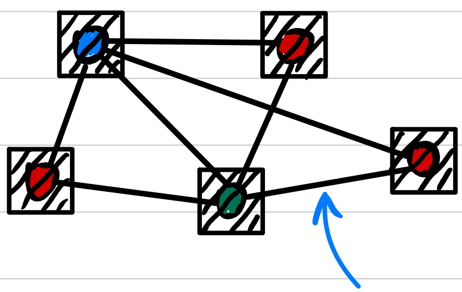
Decrypt to Σv_i

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



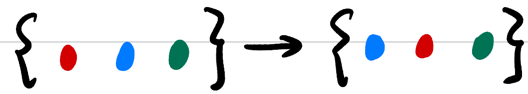
NP language $L = \{ G : G \text{ has 3-coloring} \}$

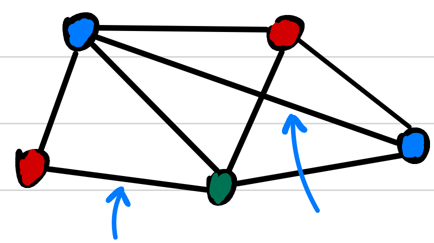
NP relation $R_L = \{ (G, 3COL) \}$



Verifier randomly pick

To achive ZK:





If $G \notin L$, $\Pr[P^* \text{ is caught}] \geq \frac{1}{|E|}$

How to amplify soundness? Repeat $\lambda \cdot |E|$ times

$\Pr[P^* \text{ is not caught}] \leq \left(1 - \frac{1}{|E|}\right)^{|E| \cdot \lambda} \simeq \left(\frac{1}{e}\right)^{\lambda}$

$\boxed{\left(1 - \frac{1}{n}\right)^{n} \simeq \frac{1}{e}}$

Commitment Scheme

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

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

Hiding: $\mathrm{Com}(0;r) \simeq \mathrm{Com}(1;s)$

Binding:

Hard to find r, s s.t. $\mathrm{Com}(0;r) = \mathrm{Com}(1;s)$

Commitment Scheme

Example 1: Hash-based commitment

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

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

Random Oracle

Hiding: RO + randomness of r

Binding: collision-resistance of H

Example 2: Pedersen Commitment

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

can be generated by Receiver

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

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

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

Hiding: h^r as OTP

Binding: DLOG of $\mathbb{G}$

Assume for contradiction that

$\mathrm{Com}(0;r) = \mathrm{Com}(1;s)$

$g^0 \cdot h^r = g^1 \cdot h^s$

$\Downarrow$

$h^{r-s} = g \Rightarrow h = g^{(r-s)^{-1}}$

Why are the schemes hiding & binding?

Zero-Knowledge Proof for Graph 3-Coloring

Input: $G = (V, E)$

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

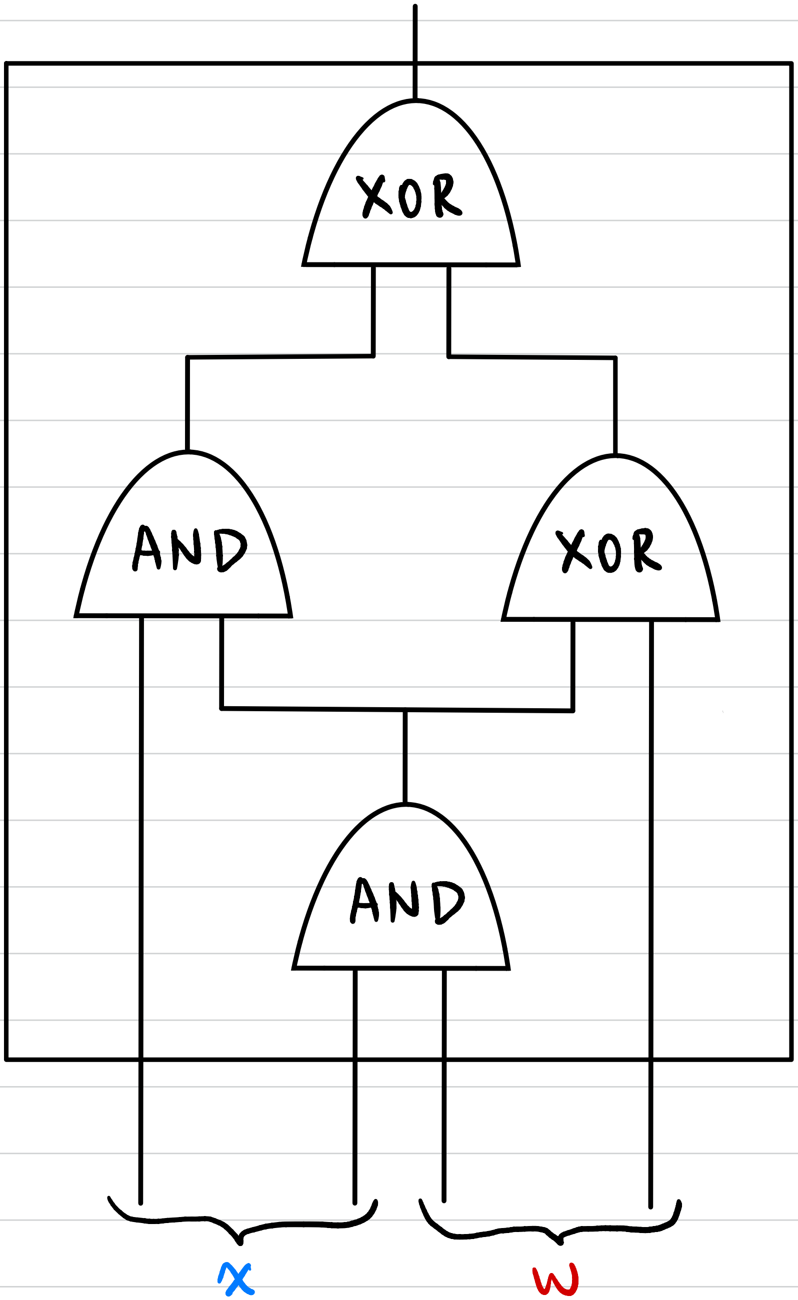
Prover		Verifier
Randomly sample $\pi : \{0,1,2\} \to \{0,1,2\}$		
$\forall v \in V$, $r_v \xleftarrow{\$} \{0,1\}^\lambda$, $c_v := \mathrm{Com}(\pi(\phi(v)); r_v)$		
	$\xrightarrow{\{c_v\}_{v \in V}}$	
	$\xleftarrow{(u,v)}$	Randomly pick an edge $(u,v) \in E$
Open commitments c_u & c_v		
	$\xrightarrow[\beta = \pi(\phi(v)),\ r_v]{\alpha = \pi(\phi(u)),\ r_u}$	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? Binding of Commitment Scheme

Zero-Knowledge? Hiding of Commitment Scheme

Circuit Satisfiability (NP Complete)



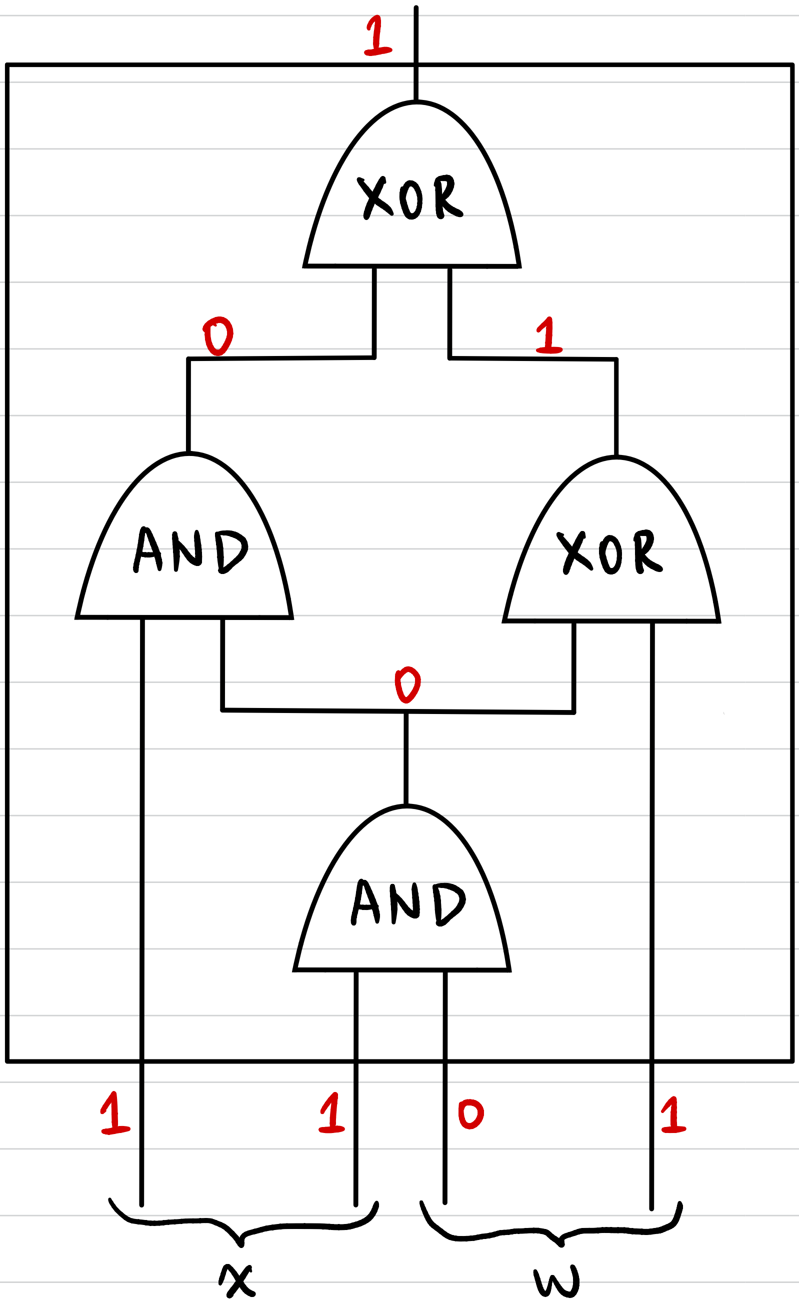
NP language $L_C = \{ x \in \{0,1\}^n : \exists w \in \{0,1\}^m \text{ s.t. } C(x,w) = 1 \}$

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

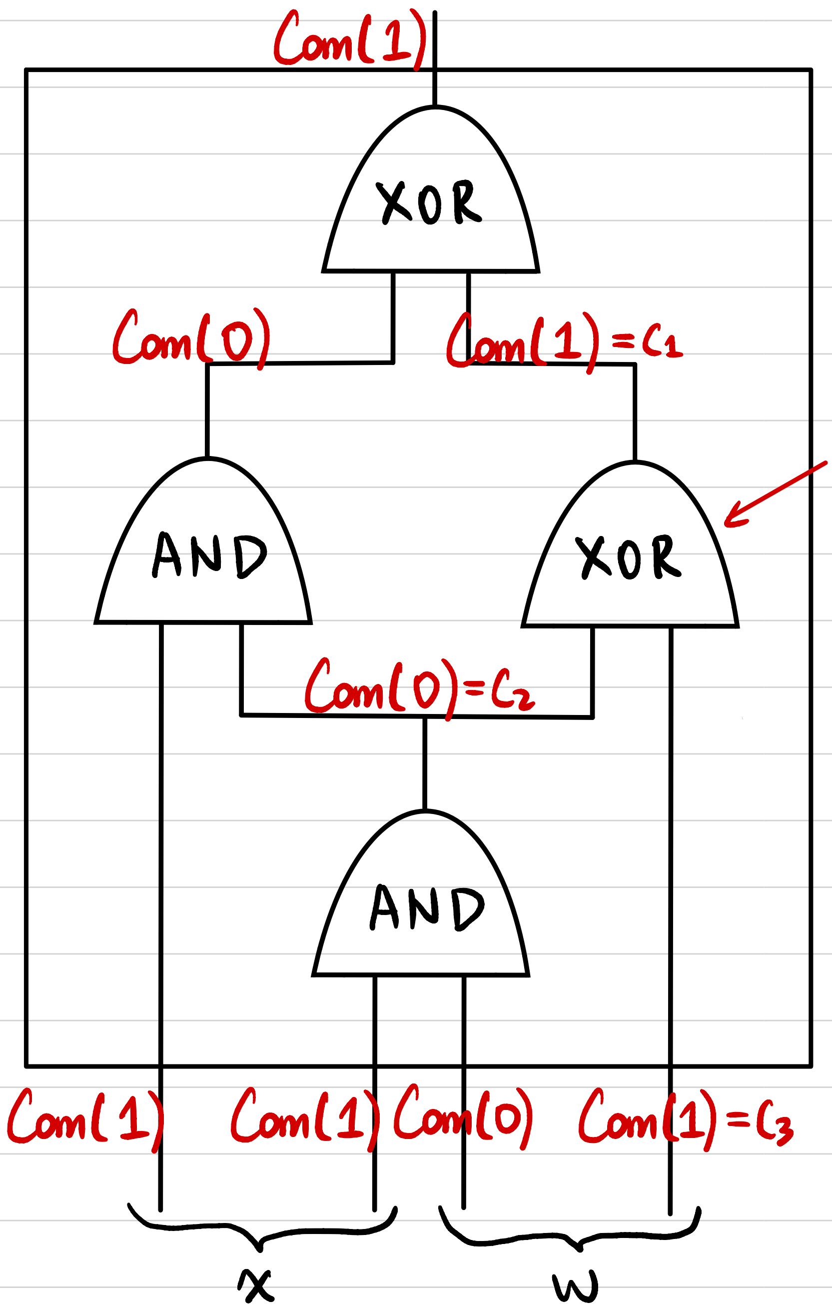
(public) statement

(secret) witness

ZKP for Circuit Satisfiability



ZKP for Circuit Satisfiability



$$\begin{pmatrix} c_1 = \mathrm{Com}(0) \\ c_2 = \mathrm{Com}(0) \\ c_3 = \mathrm{Com}(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} c_1 = \mathrm{Com}(1) \\ c_2 = \mathrm{Com}(0) \\ c_3 = \mathrm{Com}(1) \end{pmatrix}$$

OR

$$\begin{pmatrix} c_1 = \mathrm{Com}(1) \\ c_2 = \mathrm{Com}(1) \\ c_3 = \mathrm{Com}(0) \end{pmatrix}$$

OR

$$\begin{pmatrix} c_1 = \mathrm{Com}(0) \\ c_2 = \mathrm{Com}(1) \\ c_3 = \mathrm{Com}(1) \end{pmatrix}$$

Proof Systems for Circuit Satisfiability

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

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

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

Succinct Non-Interactive Argument

Prover		Verifier
Input: (x, w)	$\xrightarrow{\quad \pi \quad}$	Input: x
		Verify

- SNARG: Succinct Non-Interactive Argument
- SNARK: Succinct Non-Interactive Argument of Knowledge
- zk-SNARG / zk-SNARK: SNARG / SNARK + Zero-Knowledge

- Succint: $|\pi| = \text{poly}(\lambda, \log |C|)$
  Verifier runtime $\text{poly}(\lambda, |x|, \log |C|)$
- Argument: In Soundness / Proof of Knowledge: $\forall$ PPT P^*