CSCI 1515 Applied Cryptography

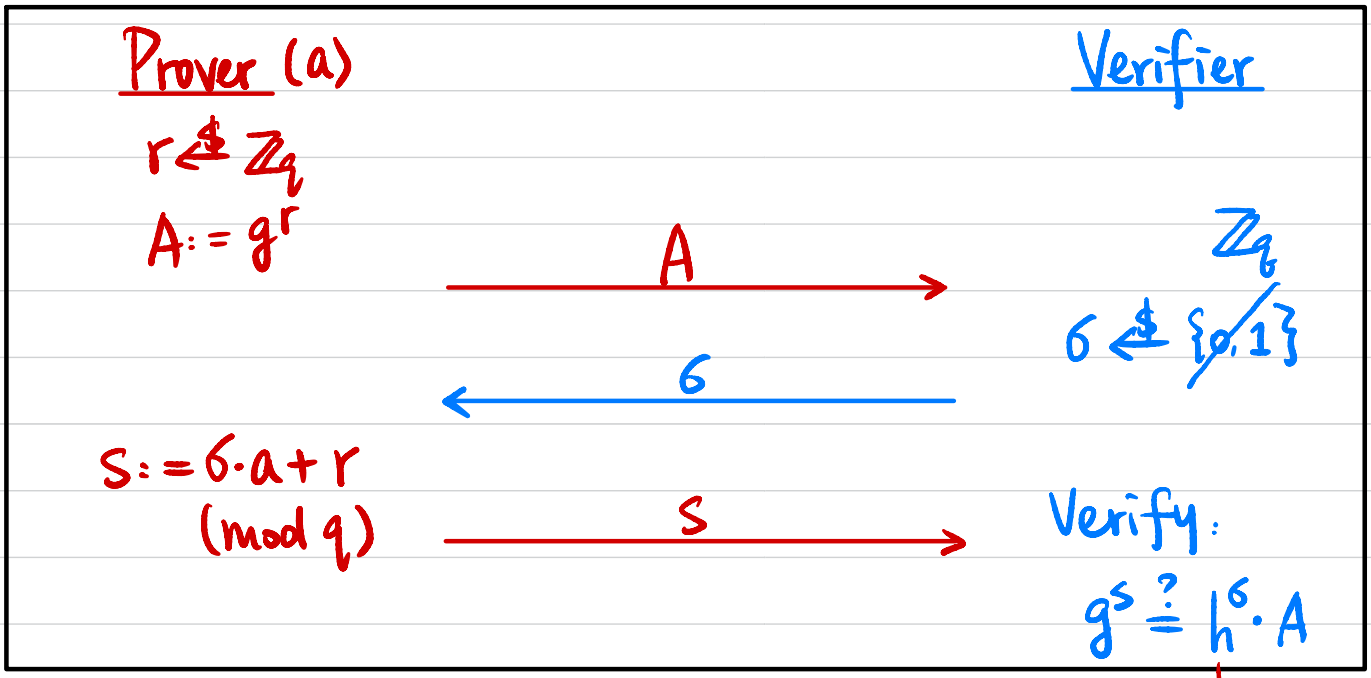
This Lecture:

- Example: Schnorr's Identification Protocol
- Anonymous Online Voting: An Overview
- Definition of Zero-Knowledge Proofs
- Code Review 1

Example: Schnorr's Identification Protocol

Public: Cyclic group G of order q, generator g, $h = g^a$

Prover's secret: a



$s \xleftarrow{\$} \mathbb{Z}_q$, $A := g^s / h$

A → ← $b = 1$ → s.t. $g^s = h \cdot A$

A → ← $b = 0$ / $b = 1$ → s s.t. $g^s = A$ / s.t. $g^s = h \cdot A$

$s = a + r$

- **Completeness?** If P knows a?

$$(g^a)^b \cdot g^r = g^{ab+r} = g^s$$

- **Soundness?** If P doesn't know a?

A → ← $b = 0$: s s.t. $g^s = A$; ← $b = 1$: s' s.t. $g^{s'} = h \cdot A$

$$g^{s'-s} = h \Rightarrow a = s' - s$$

- **Zero-Knowledge?** What does V learn?

Anonymous Online Voting

ElGamal ZKP

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

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

⋮

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

$\Downarrow$

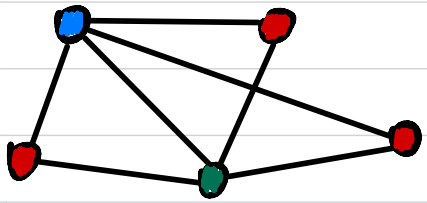
$Enc(\Sigma v_i)$

$\Downarrow$

Decrypt to Σv_i

NP as a Proof System

Example: Gragh 3-Coloring



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

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

(public) statement

(secret) witness

Example: DLOG: Cyclic group G of order q, generator g

NP language $L = \{ h : h \in G \}$

NP relation $R_L = \{ (h, a) : h = g^a \}$

(public) statement

(secret) witness

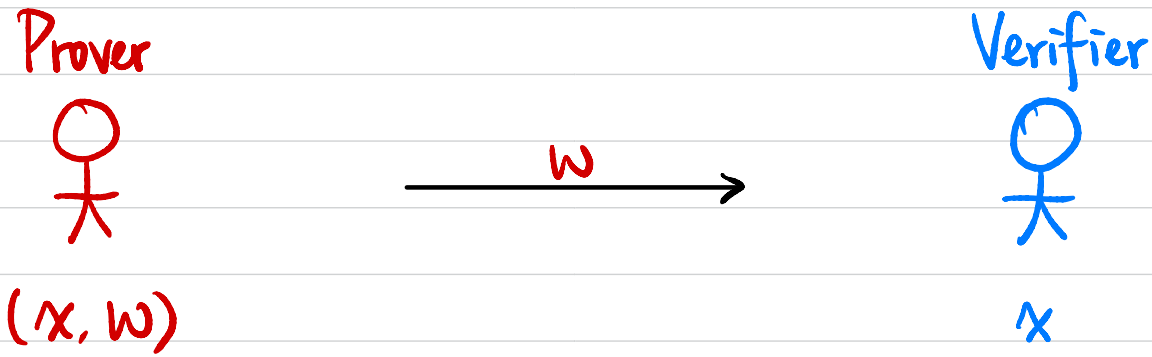
NP as a Proof System

A language L is in NP if $\exists$ poly-time V s.t.

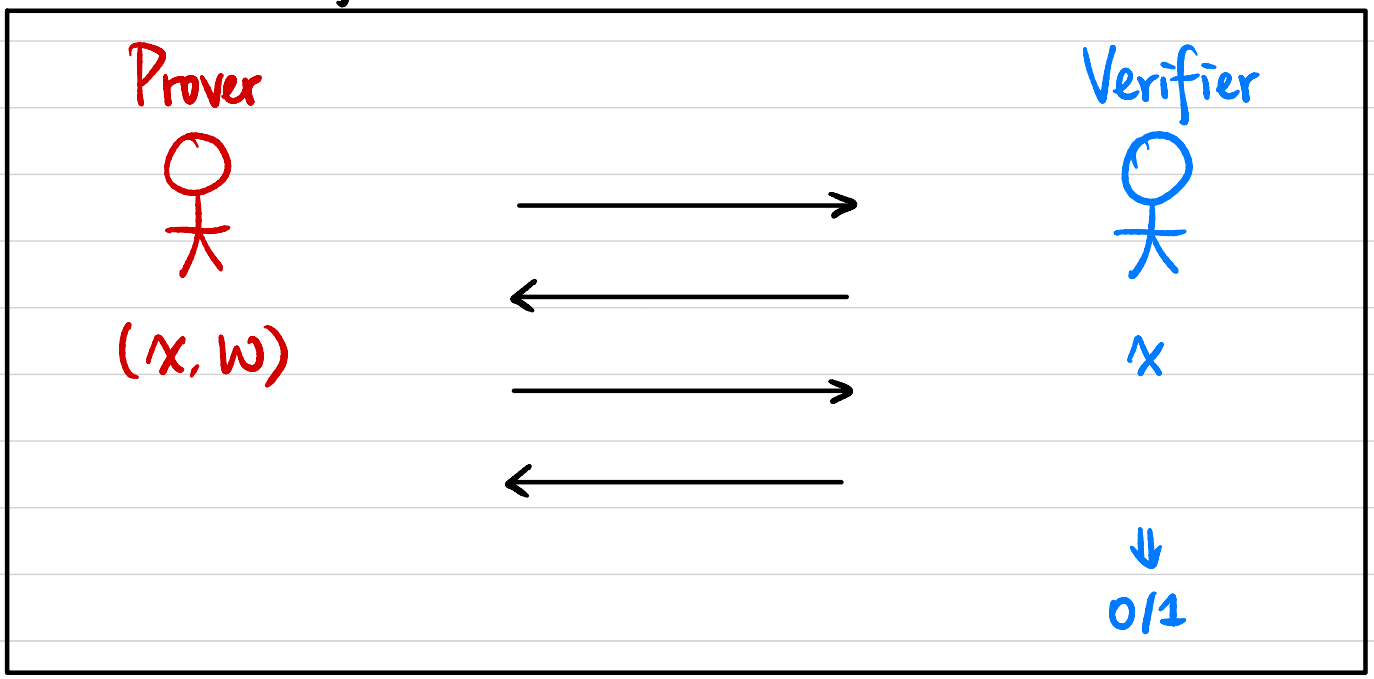
- **Completeness:** $\forall x \in L$, $\exists w$ s.t. $V(x, w) = 1$

  (w: witness)

- **Soundness:** $\forall x \notin L$, $\forall w^*$, $V(x, w^*) = 0$



Zero-Knowledge Proof (ZKP)



Let (P, V) be a pair of probabilistic poly-time (PPT) interactive machines.

(P, V) is a zero-knowledge proof system for a language L with associated relation R_L if

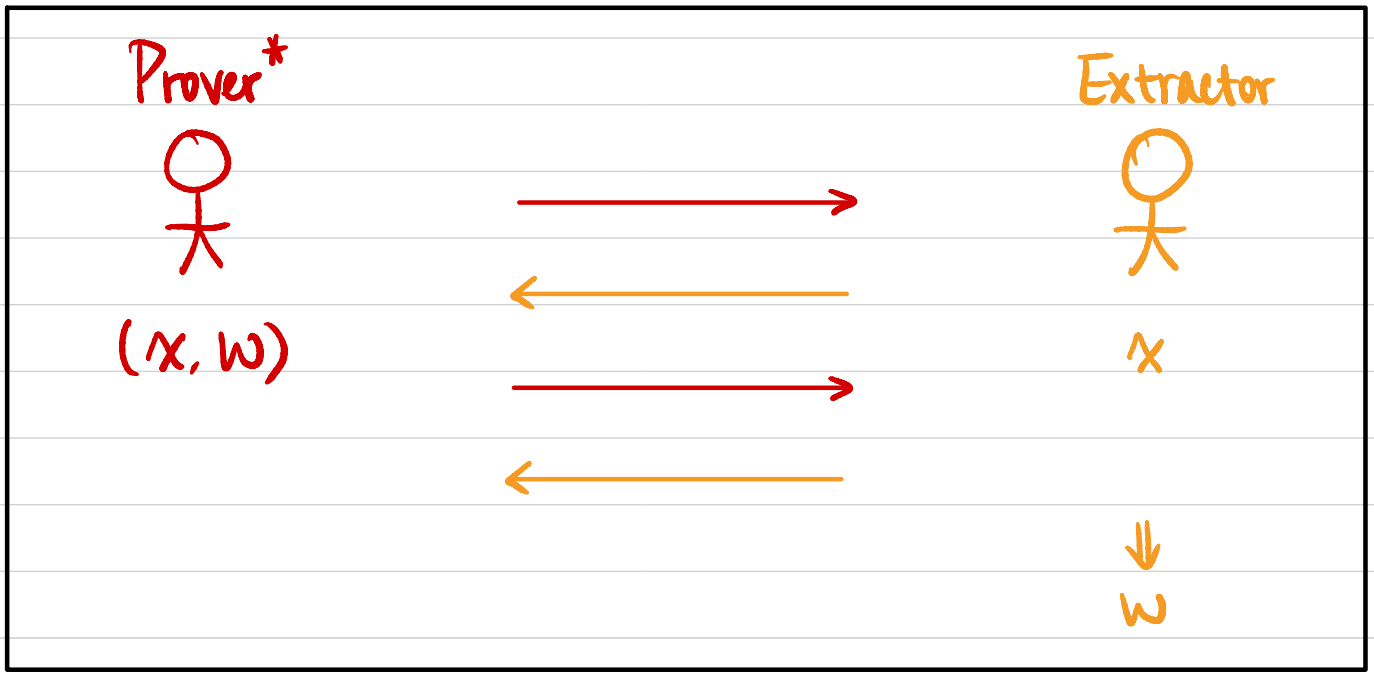
- **Completeness:** $\forall (x, w) \in R_L$, $\Pr[P(x, w) \longleftrightarrow V(x) \text{ outputs } 1] = 1$.

  $\forall (x, w) \in R_L$, P can prove it.

- **Soundness:** $\forall x \notin L$, $\forall P^*$, $\Pr[P^*(x) \longleftrightarrow V(x) \text{ outputs } 1] \simeq 0$.

  $\forall x \notin L$, any P^* cannot prove it.

Proof of Knowledge (PoK)

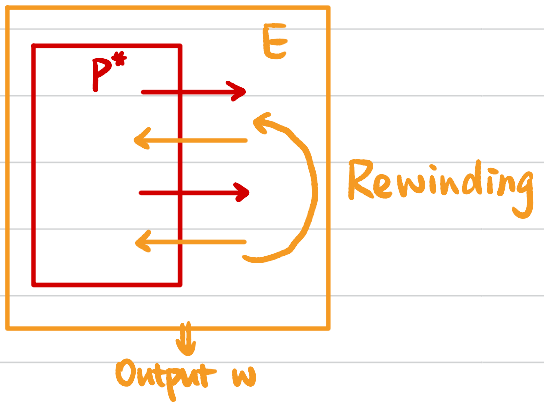


- **Proof of Knowledge:**

  $\exists$ PPT E s.t. $\forall p^*$, $\forall x$,

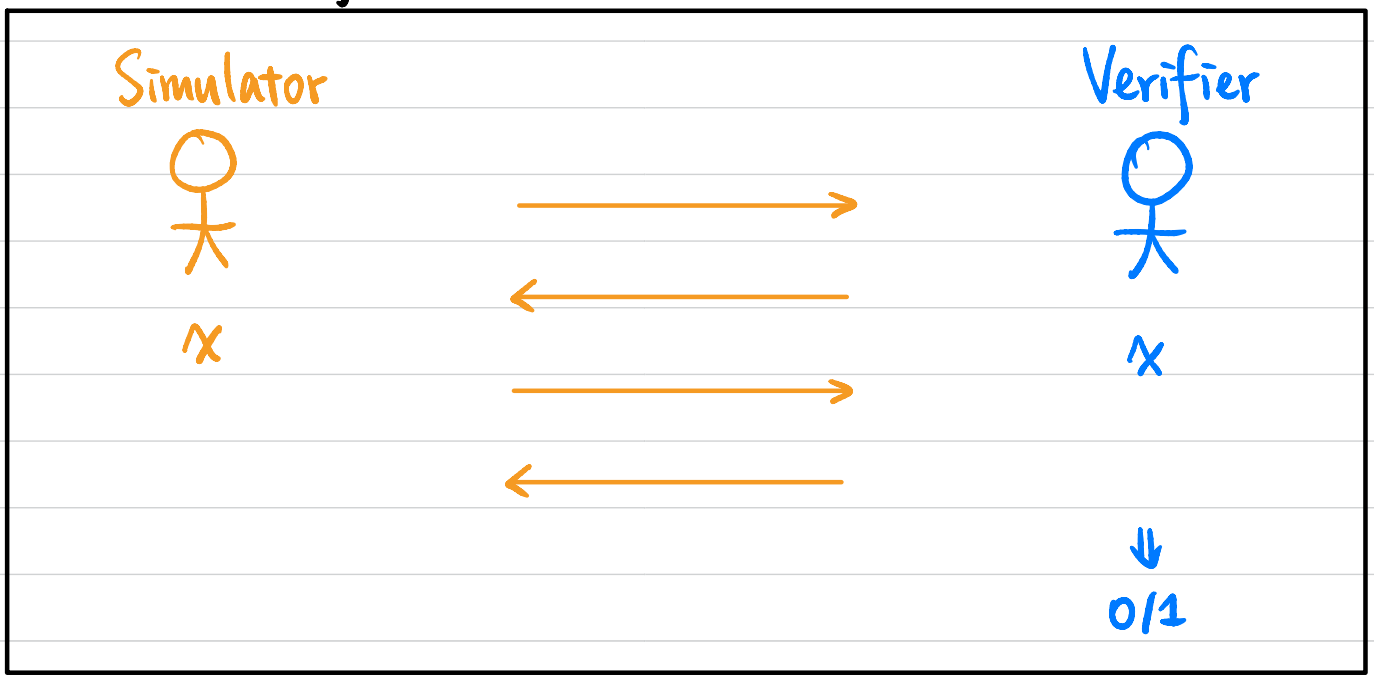
  $$\Pr\left[E^{p^*(\cdot)}(x) \text{ outputs } w \text{ s.t. } (x,w) \in R_L\right] \simeq \Pr\left[p^* \leftrightarrow V(x) \text{ outputs } 1\right].$$

  If p^* can prove it, p^* must know w.



How is it possible?

Zero-Knowledge Proof (ZKP)



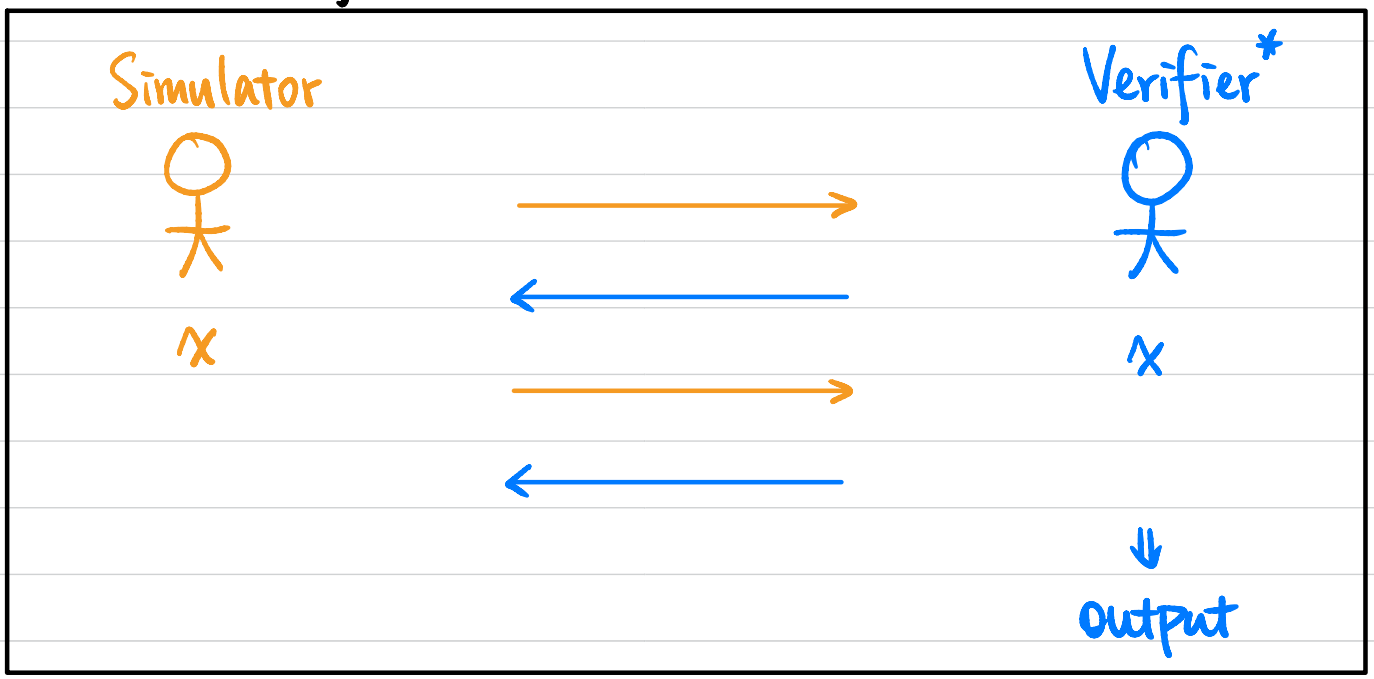
- **Honest-Verifier Zero-Knowledge (HVZK):**

$$\exists \text{PPT } S \quad \text{s.t.} \quad \forall (x, w) \in R_L,$$

$$\text{View}_V[P(x,w) \longleftrightarrow V(x)] \simeq S(x)$$

An honest V doesn't learn anything about w.

Zero-Knowledge Proof (ZKP)

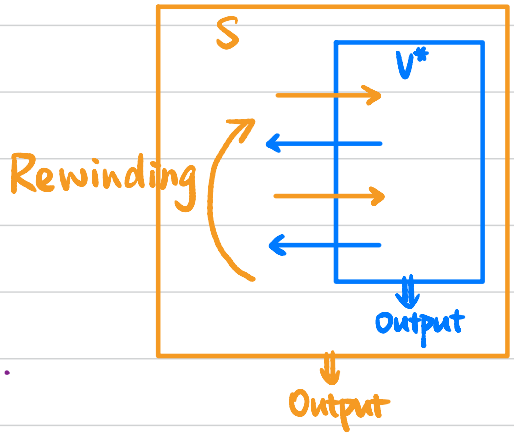


- **Zero-Knowledge** (Malicious Verifier):

$$\forall \text{PPT } V^*, \ \exists \text{PPT } S \ \text{ s.t. } \ \forall (x, w) \in R_L,$$

$$\text{Output}_{V^*}[P(x,w) \longleftrightarrow V^*(x)] \simeq S(x)$$

A malicious V^* doesn't learn anything about w.



How is it possible?

Zero-Knowledge Proof of Knowledge

- **Completeness:** $\forall (x,w) \in R_L$, $\Pr[P(x,w) \longleftrightarrow V(x) \text{ outputs } 1] = 1$.
  $\forall (x,w) \in R_L$, P can prove it.

- **Soundness.** $\forall x \notin L$, $\forall P^*$, $\Pr[P^*(x) \longleftrightarrow V(x) \text{ outputs } 1] \simeq 0$.
  $\forall x \notin L$, any P^* cannot prove it.

- **Proof of Knowledge:** $\exists$ PPT E s.t. $\forall P^*$, $\forall x$,
  $\Pr[E^{P^*(\cdot)}(x) \text{ outputs } w \text{ s.t. } (x,w) \in R_L] \simeq \Pr[P^* \leftrightarrow V(x) \text{ outputs } 1]$.
  If P^* can prove it, P^* must know w.

- **Honest-Verifier Zero-Knowledge (HVZK):** $\exists$ PPT S s.t. $\forall (x,w) \in R_L$,
  $\text{View}_V[P(x,w) \longleftrightarrow V(x)] \simeq S(x)$
  An honest V doesn't learn anything about w.

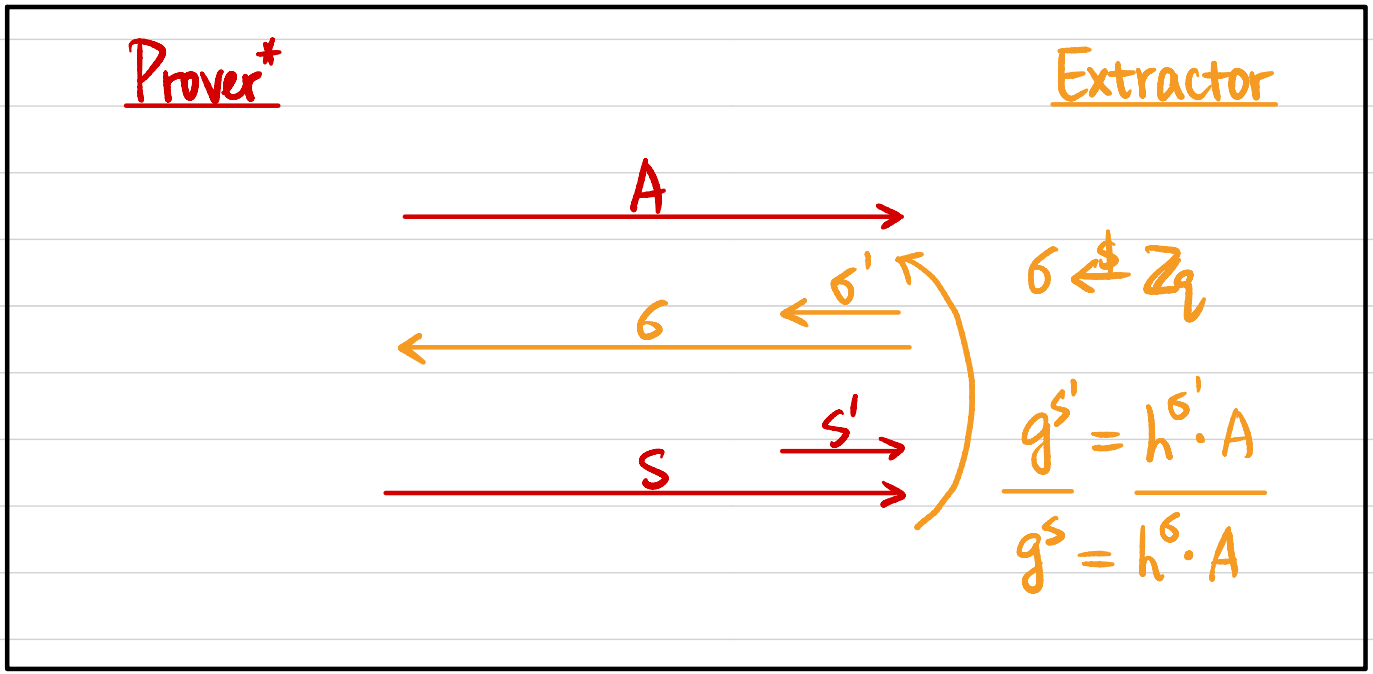
- **Zero-Knowledge:** $\forall$ PPT V^*, $\exists$ PPT S s.t. $\forall (x,w) \in R_L$,
  $\text{Output}_{V^*}[P(x,w) \longleftrightarrow V^*(x)] \simeq S(x)$
  A malicious V^* doesn't learn anything about w.

Example: Schnorr's Identification Protocol

Proof of Knowledge?

$\exists$ PPT E s.t. $\forall P^*$, $\forall x$,

$$\Pr\left[E^{P^*(\cdot)}(x) \text{ outputs } w \text{ s.t. } (x,w) \in R_L\right] \simeq \Pr\left[P^* \leftrightarrow V(x) \text{ outputs } 1\right].$$



$$\Rightarrow g^{s'-s} = h^{\sigma'-\sigma}$$

$$g^{(s'-s)\cdot(\sigma'-\sigma)^{-1}} = h$$

$$\Downarrow$$

$$a = (s'-s)\cdot(\sigma'-\sigma)^{-1} \bmod q$$

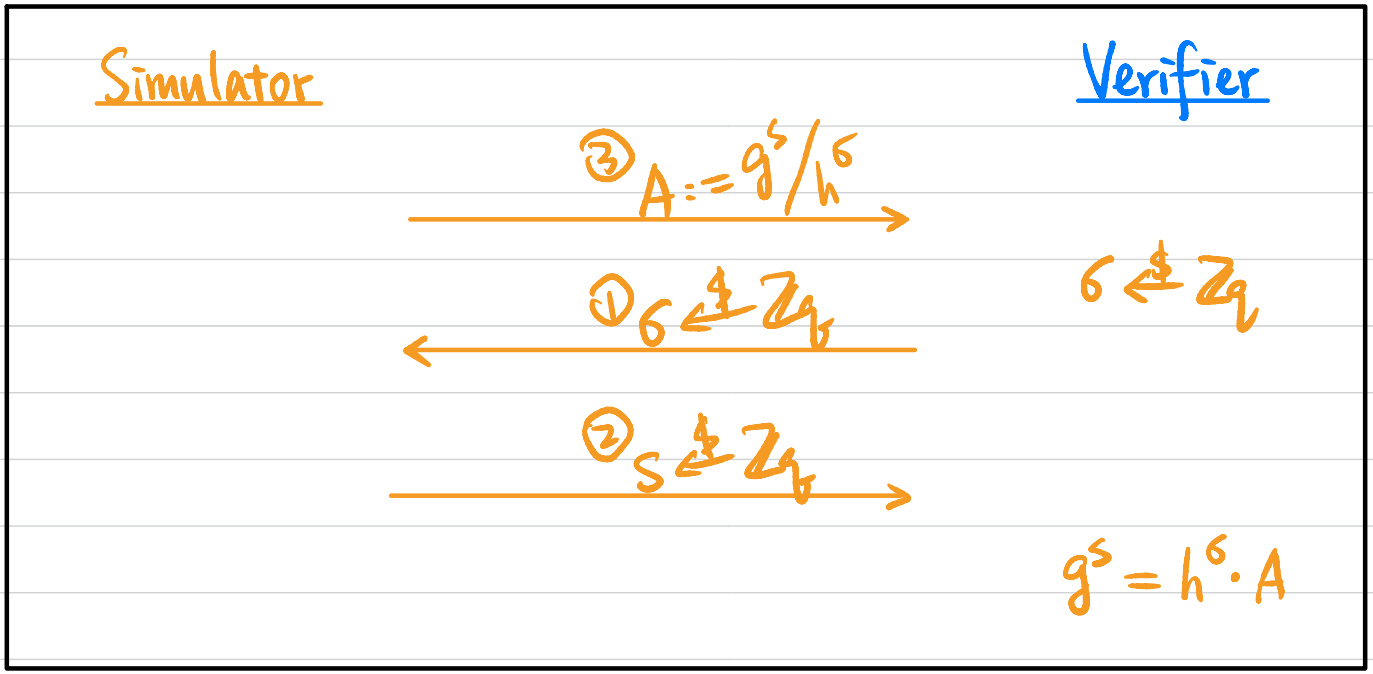
How to extract a s.t. $h = g^a$?

Example: Schnorr's Identification Protocol

Honest-Verifier Zero-Knowledge (HVZK)?

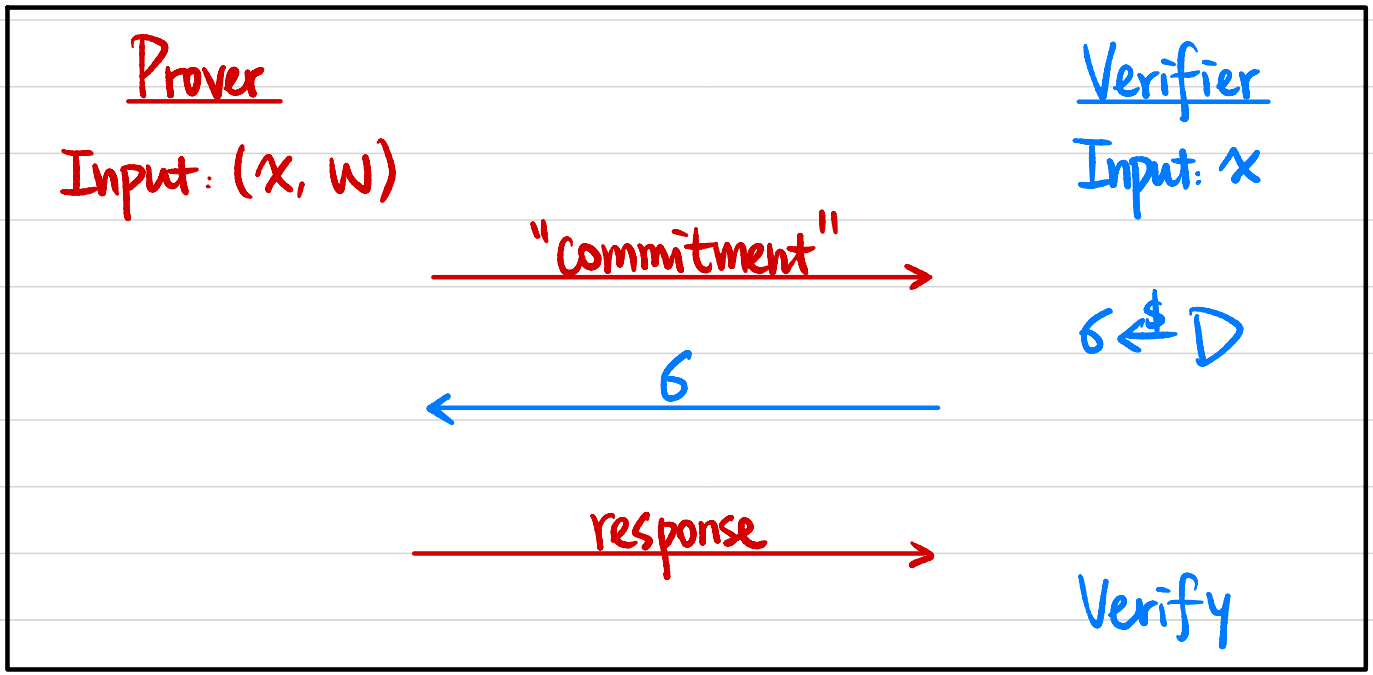
$\exists$ PPT S s.t. $\forall (x, w) \in R_L$,

$$\text{View}_V[P(x,w) \longleftrightarrow V(x)] \simeq S(x)$$



How to generate (A, s) s.t. $g^s = h^\sigma \cdot A$?

Sigma Protocols Σ



Code Review 1 (Signal & Auth)

- Sat 2/28 - Fri 3/6
- No TA Hours for the week
- Sign up for a 15min 1-on-1 slot with a TA
- All in-person except students from online session
- Closed-book, 2 conceptual questions

Example: ElGamal Encryption

- $Gen(1^\lambda)$:

  $(G, q, g) \leftarrow G(1^\lambda)$

  $x \xleftarrow{\$} \mathbb{Z}_q$, compute $h = g^x$

  $pk = (G, q, g, h)$ $\quad sk = x$

- $Enc_{pk}(m)$: $m \in G$

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

  $c = \langle g^y, h^y \cdot m \rangle$

- $Dec_{sk}(c)$:

  $c = \langle c_1, c_2 \rangle$

  $m = c_2 \cdot (c_1^{sk})^{-1}$

Correctness?

CPA Security?