

CSCI 1515 Applied Cryptography

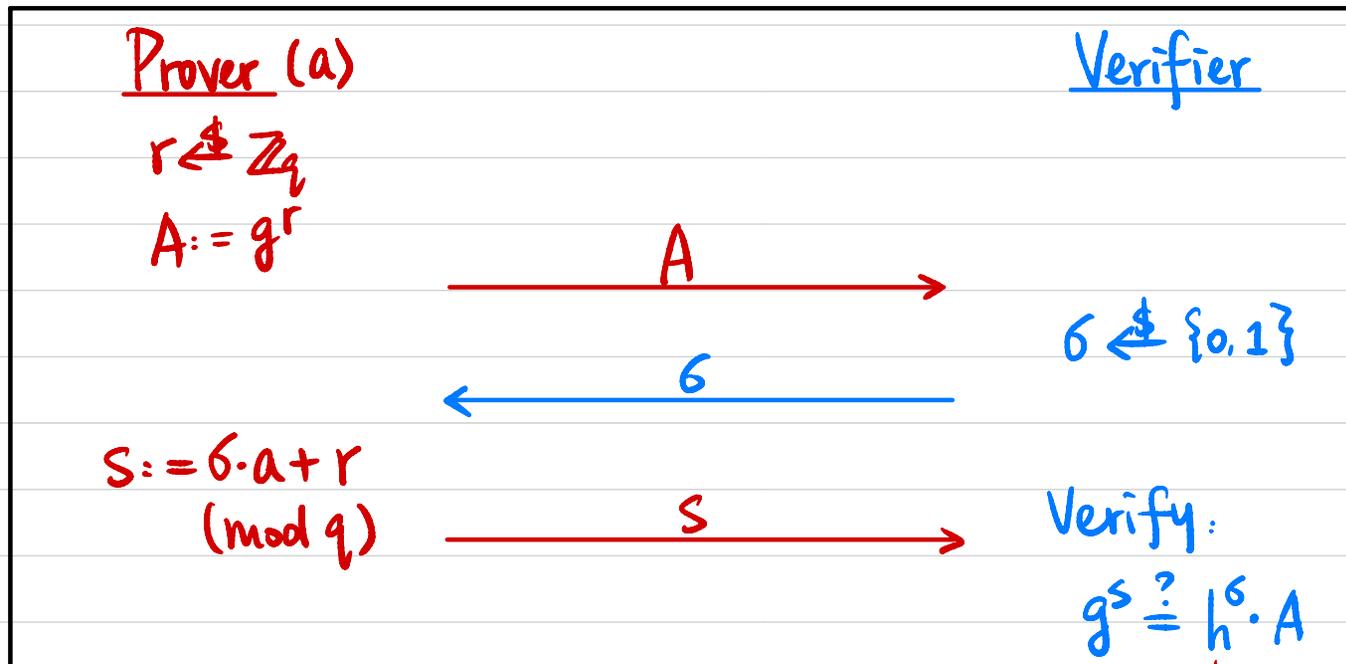
This Lecture:

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

Example: Schnorr's Identification Protocol

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

Prover's secret: a



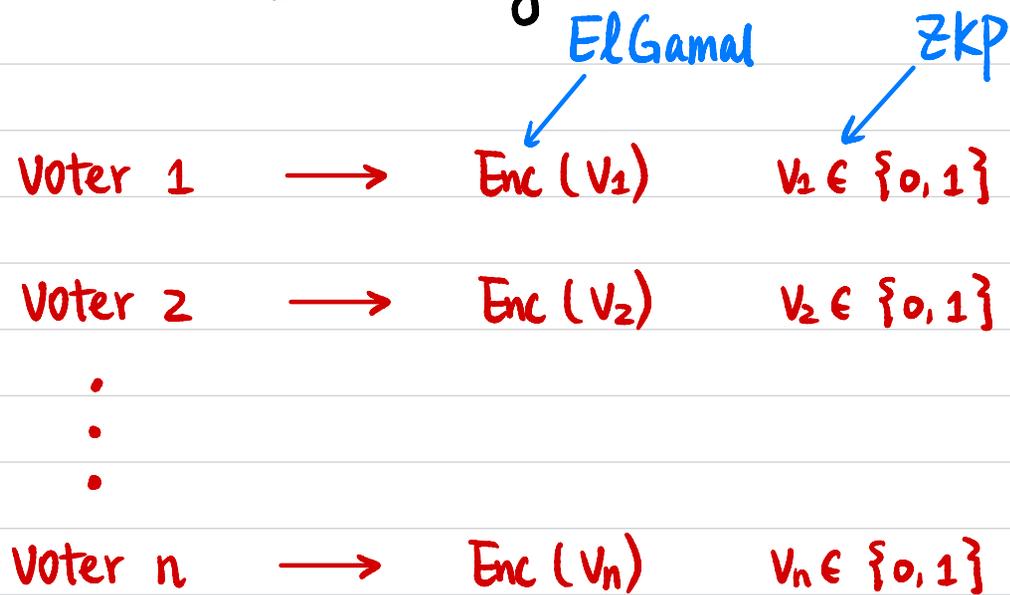
• **Completeness?** If P knows a ?

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

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

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

Anonymous Online Voting



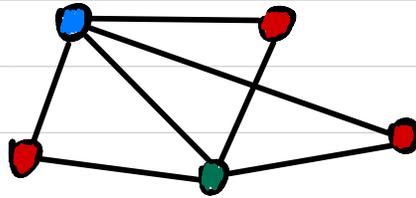
Enc (ΣV_i)



Decrypt to ΣV_i

NP as a Proof System

Example: Graph 3-coloring



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

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

(public) Statement (secret) Witness

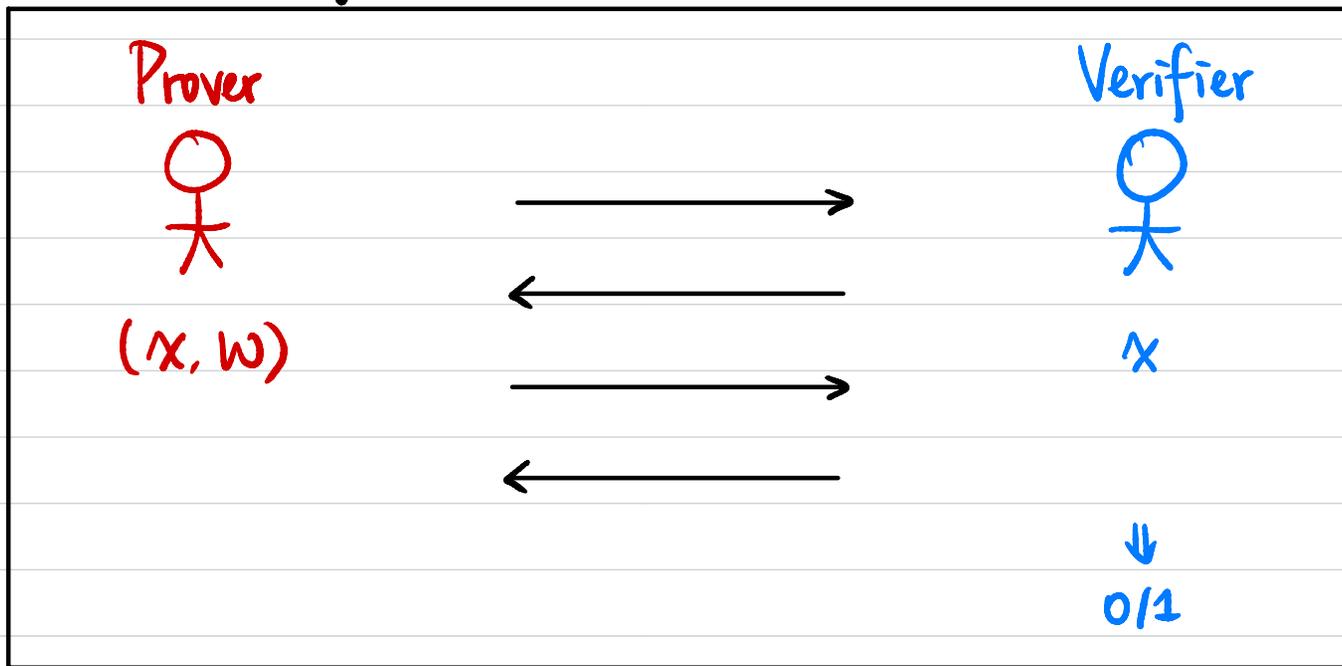
Example: DLOG: Cyclic group G of order q , generator g

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

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

(public) Statement (secret) Witness

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

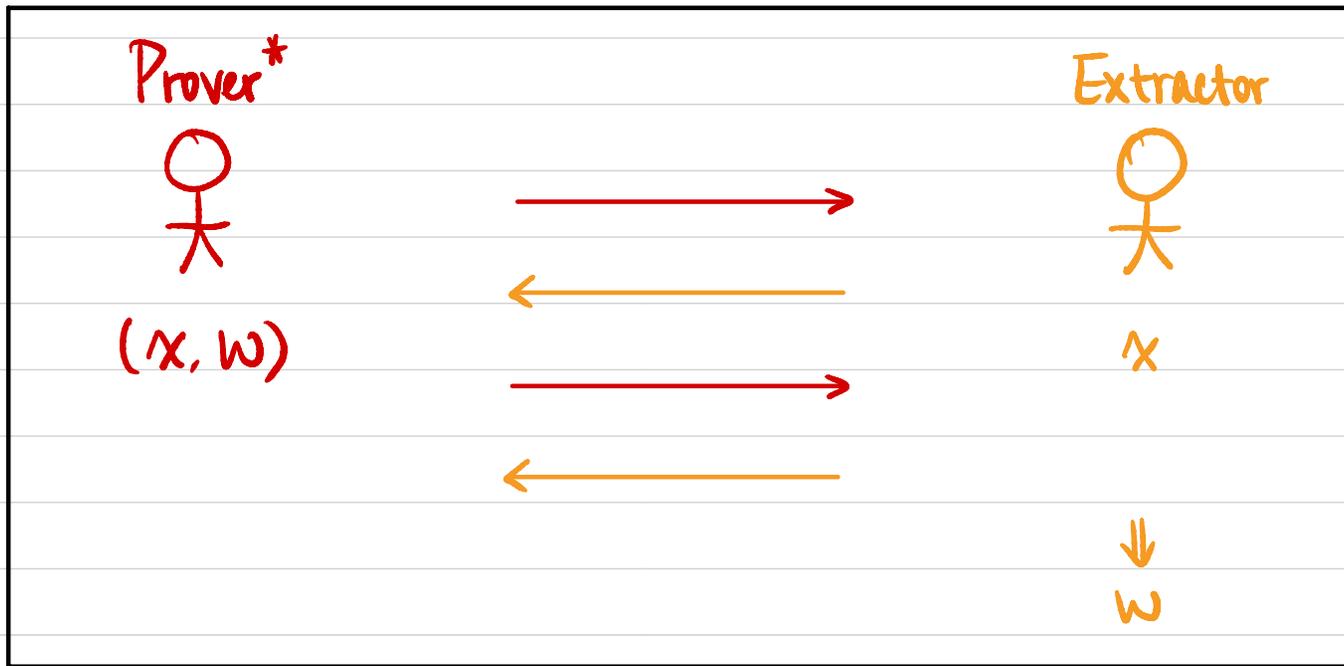
• **Completeness:** $\forall (x, w) \in R_L, \Pr [P(x, w) \longleftrightarrow V(x) \text{ outputs } 1] = 1.$

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

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

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

Proof of Knowledge (PoK)



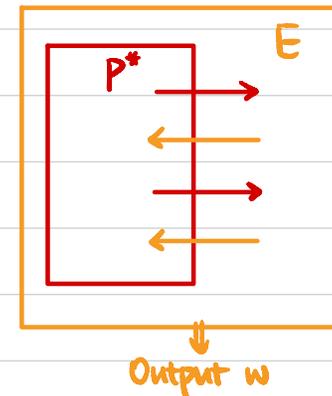
• Proof of Knowledge:

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

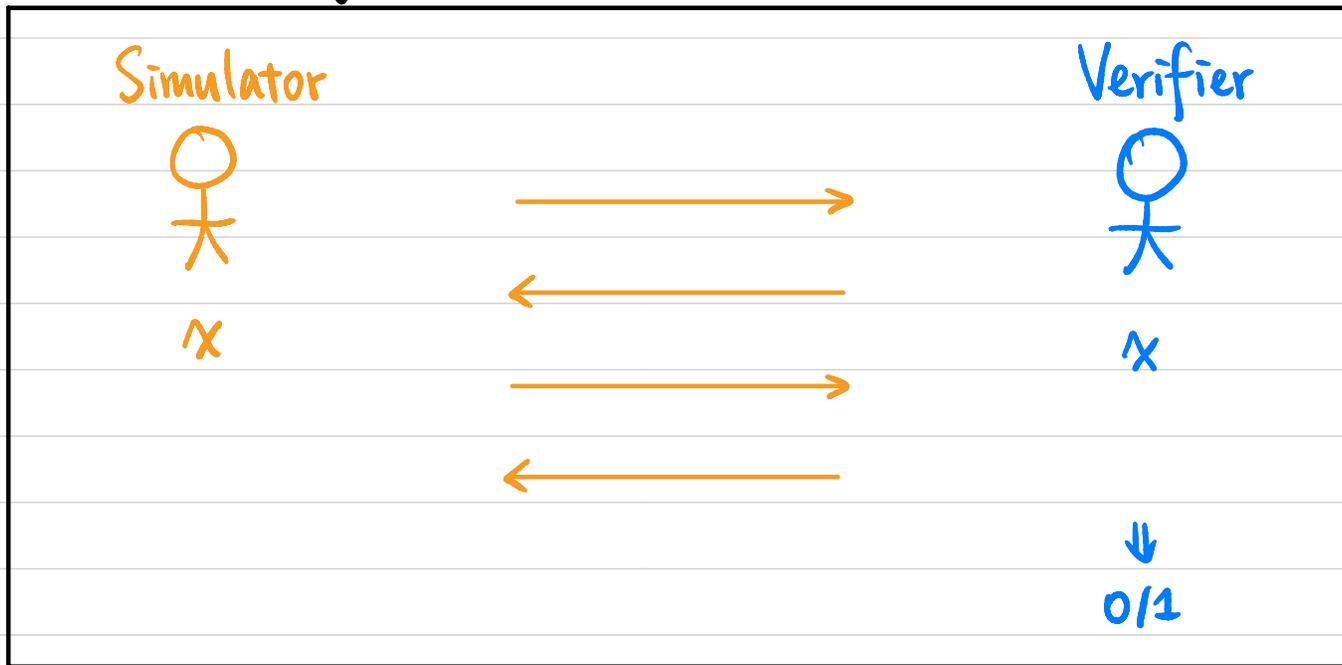
$\Pr[E^{P^*(\cdot)}(x) \text{ outputs } w \text{ s.t. } (x, w) \in R_L] \approx \Pr[P^* \leftrightarrow V(x) \text{ outputs } 1].$

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

How is it possible?



Zero-Knowledge Proof (ZKP)



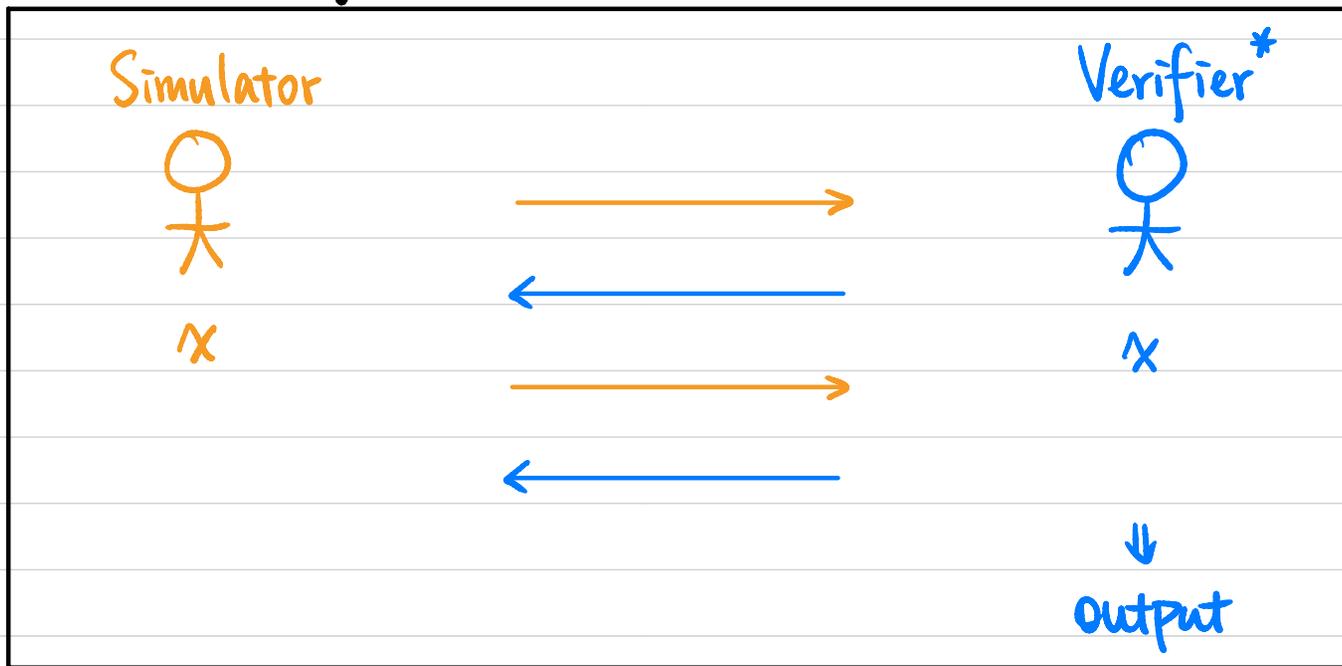
- **Honest-Verifier Zero-Knowledge (HVZK):**

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

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

An honest V doesn't learn anything about w .

Zero-Knowledge Proof (ZKP)

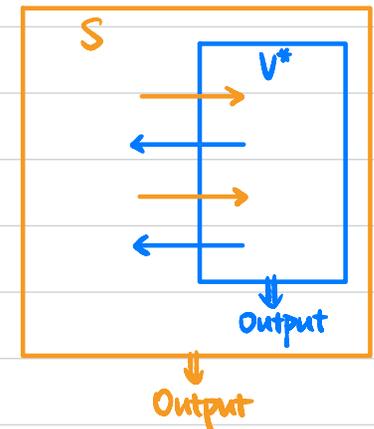


- **Zero-Knowledge** (Malicious Verifier):

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

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

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



How is it possible?

Zero-Knowledge Proof of Knowledge

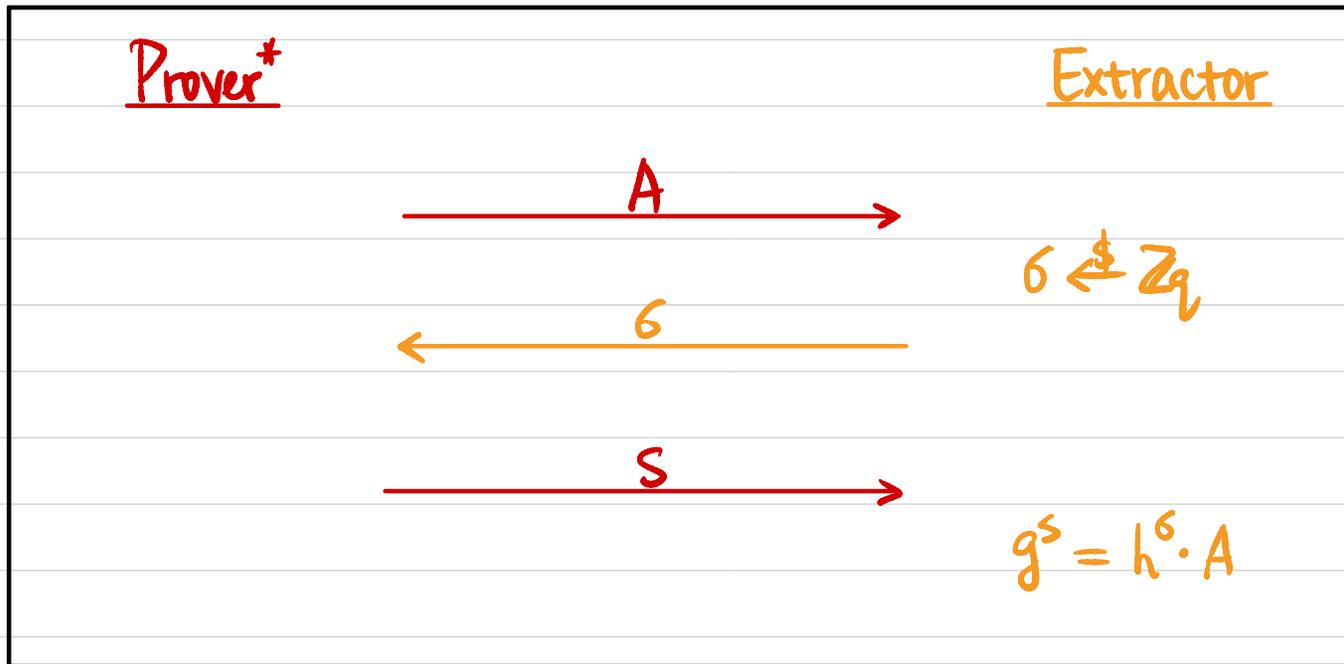
- **Completeness:** $\forall (x, w) \in R_L, \Pr [P(x, w) \leftrightarrow V(x) \text{ outputs } 1] = 1.$
 $\forall (x, w) \in R_L, P \text{ can prove it.}$
- **Soundness:** $\forall x \notin L, \forall P^*, \Pr [P^*(x) \leftrightarrow V(x) \text{ outputs } 1] \approx 0.$
 $\forall x \notin L, \text{ any } P^* \text{ cannot prove it.}$
- **Proof of Knowledge:** $\exists \text{PPT } E \text{ s.t. } \forall P^*, \forall x,$
 $\Pr [E^{P^*(\cdot)}(x) \text{ outputs } w \text{ s.t. } (x, w) \in R_L] \approx \Pr [P^* \leftrightarrow V(x) \text{ outputs } 1].$
 $\text{If } P^* \text{ can prove it, } P^* \text{ must know } w.$
- **Honest-Verifier Zero-Knowledge (HVZK):** $\exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$
 $\text{View}_V [P(x, w) \leftrightarrow V(x)] \approx S(x)$
 $\text{An honest } V \text{ doesn't learn anything about } w.$
- **Zero-Knowledge:** $\forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$
 $\text{Output}_{V^*} [P(x, w) \leftrightarrow V^*(x)] \approx S(x)$
 $\text{A malicious } V^* \text{ doesn't learn anything about } w.$

Example: Schnorr's Identification Protocol

Proof of Knowledge?

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

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



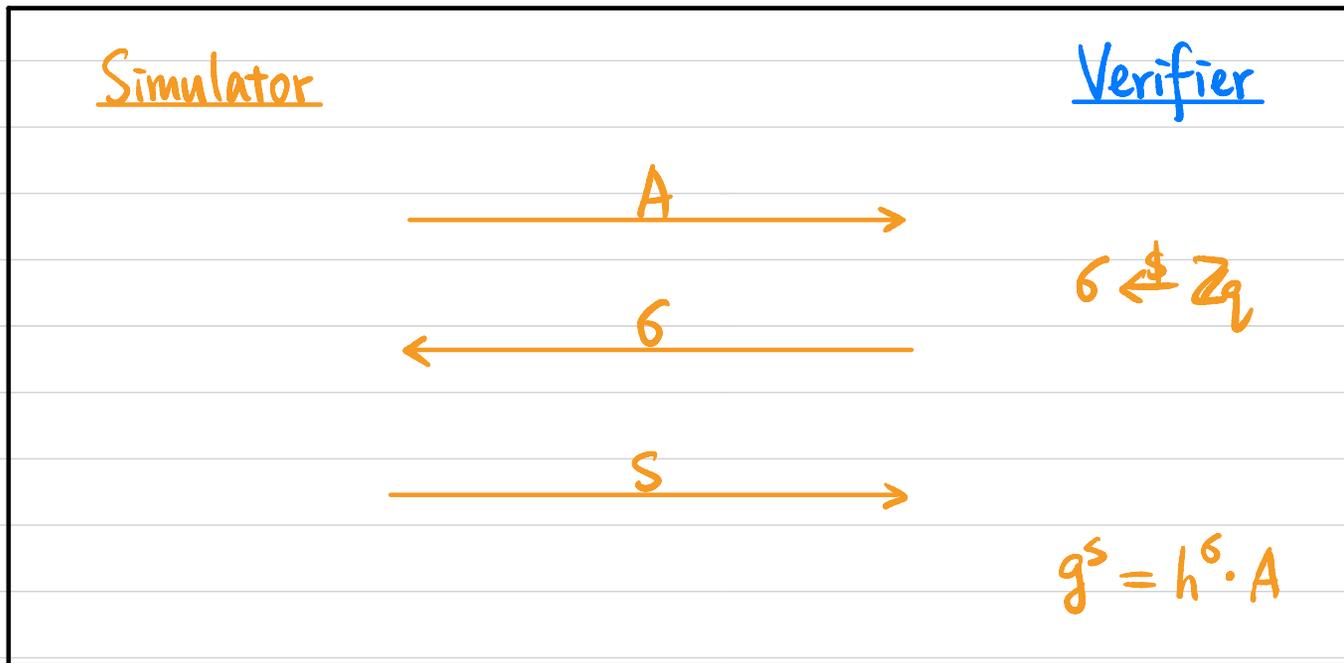
How to extract a st. $h = g^a$?

Example: Schnorr's Identification Protocol

Honest-Verifier Zero-Knowledge (HVZK)?

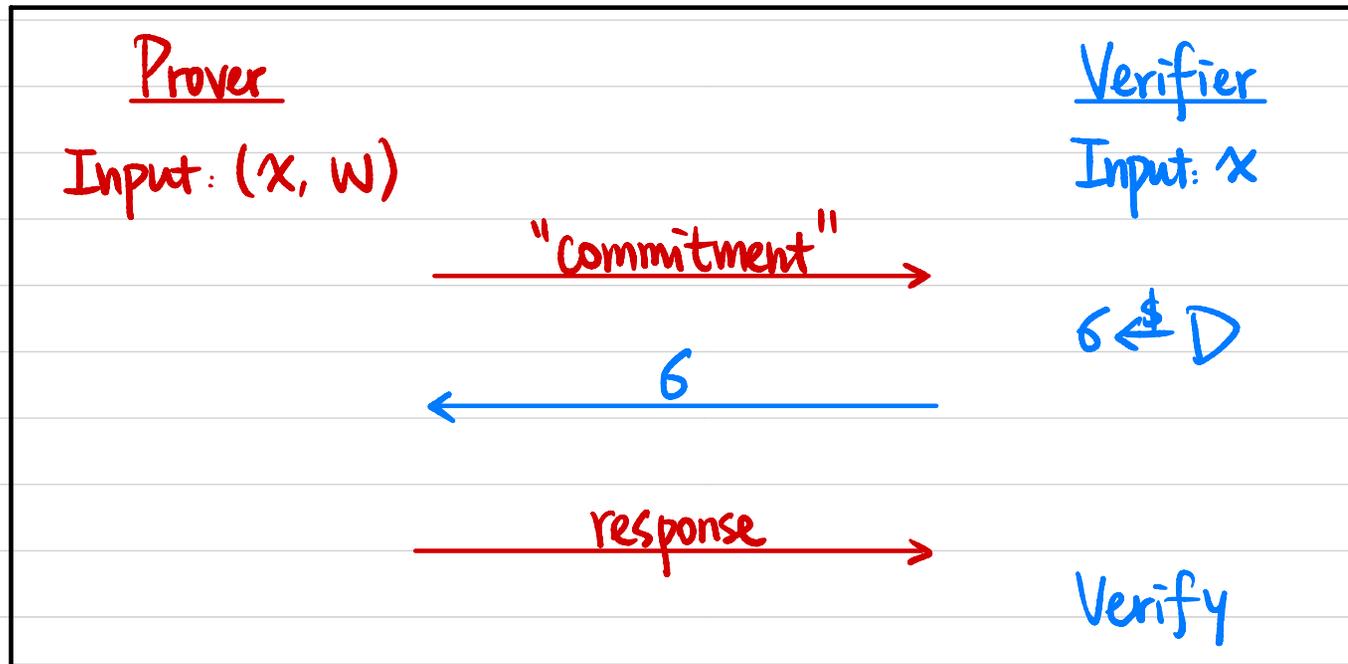
\exists PPT S s.t. $\forall (x, w) \in R_L,$

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



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

Sigma Protocols Σ

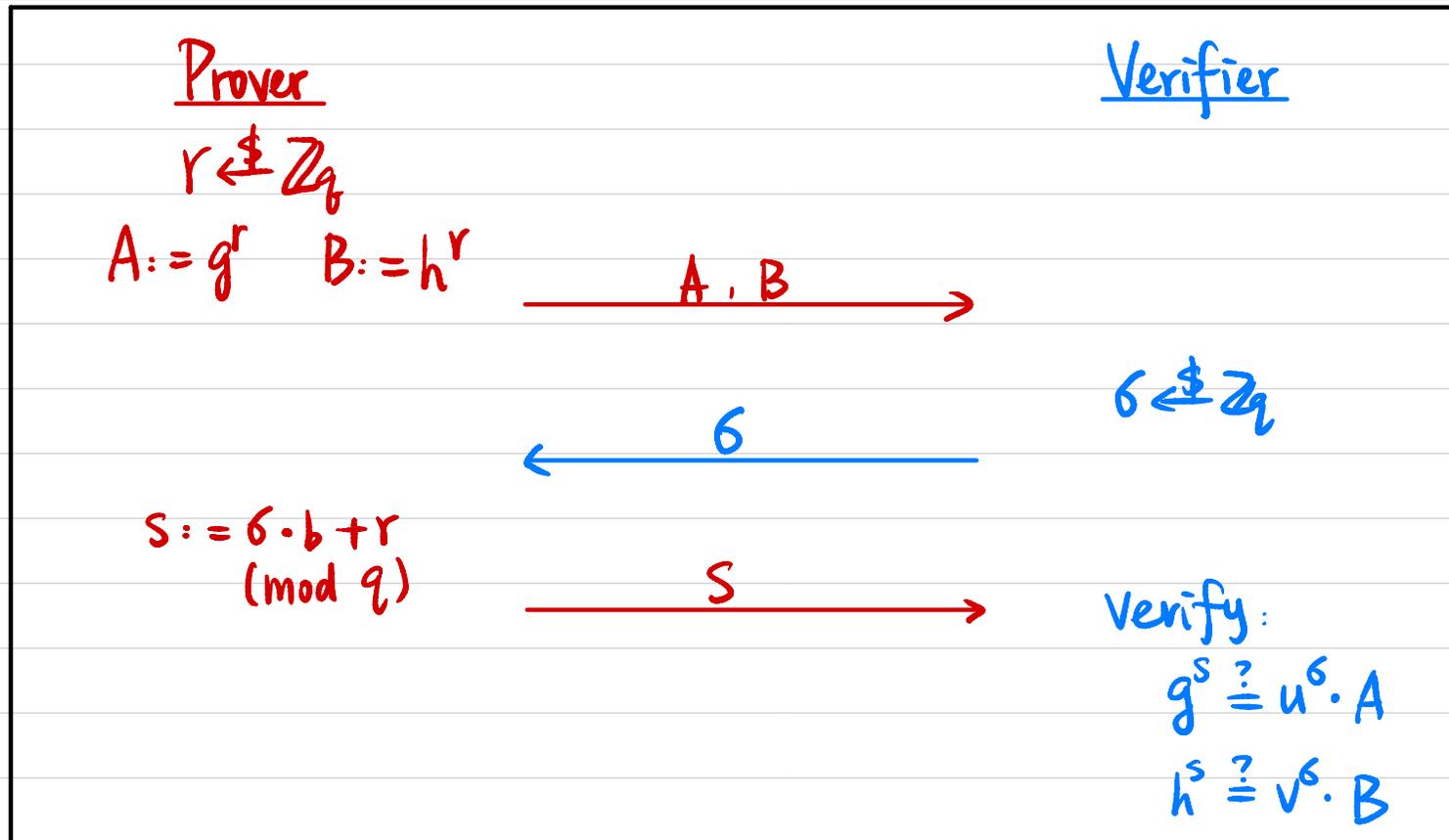


Example: Diffie-Hellman Tuple

Public: Cyclic group G of order q , generator g , $(h, u, v) = (g^a, g^b, g^{ab})$

Prover's secret witness: b s.t. $u = g^b \wedge v = h^b$

$$R_L = \{ (h, u, v), b \}$$



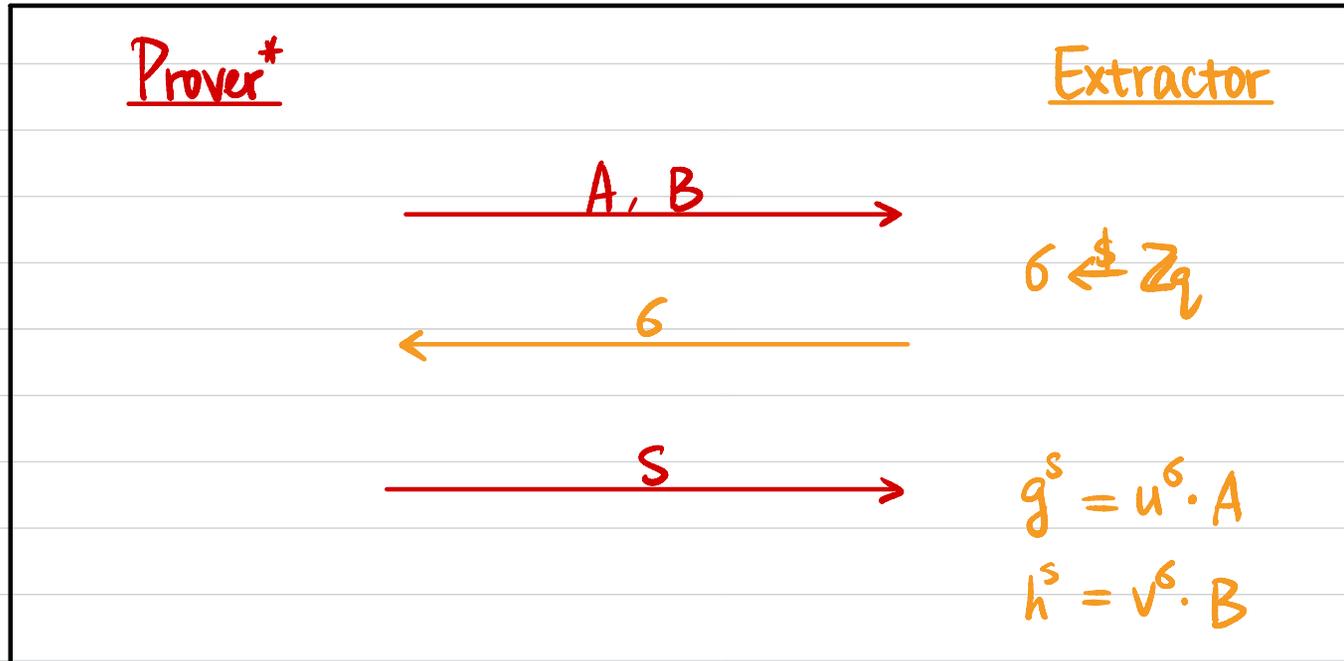
Completeness? $\forall (x, w) \in R_L \quad \Pr [P(x, w) \leftrightarrow V(x) \text{ outputs } 1] = 1.$

Example: Diffie-Hellman Tuple

Proof of Knowledge?

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

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



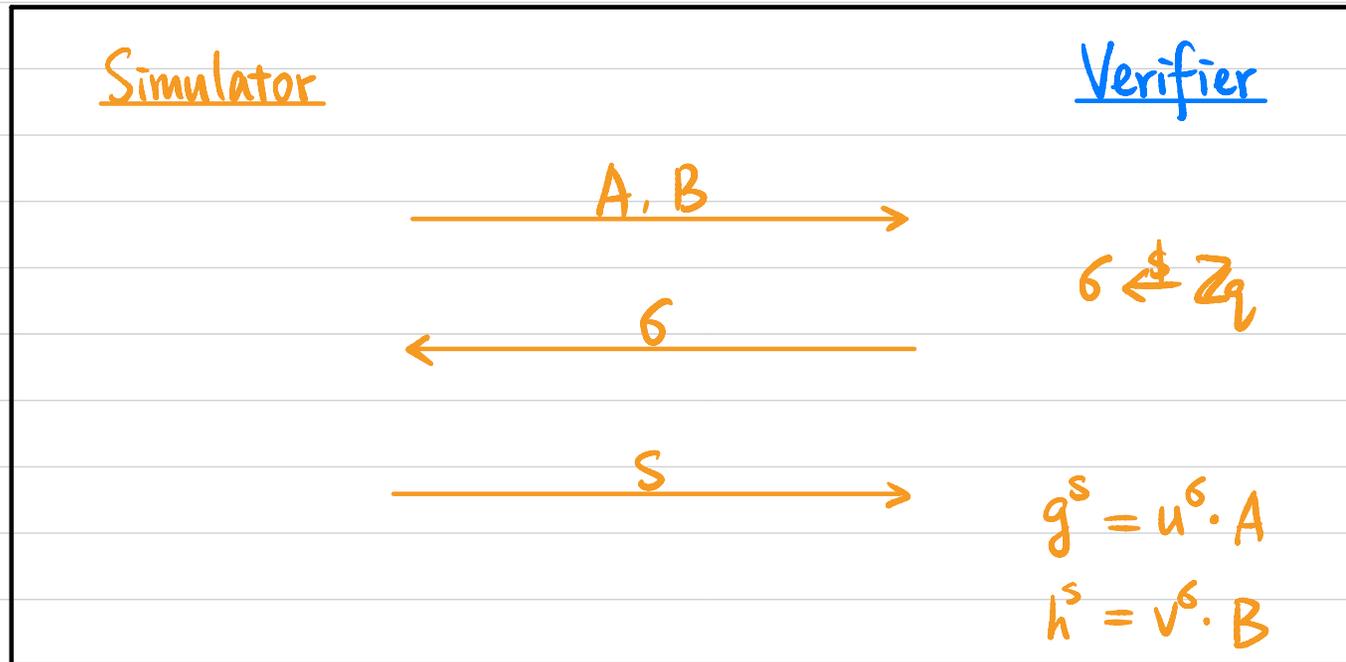
How to extract b s.t. $u = g^b \wedge v = h^b$?

Example: Diffie-Hellman Tuple

Honest-Verifier Zero-Knowledge (HVZK)?

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

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



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

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

- $\text{Gen}(1^\lambda)$:
 $(G, q, g) \leftarrow G(1^\lambda)$
 $x \xleftarrow{\$} \mathbb{Z}_q$, compute $h = g^x$
 $\text{PK} = (G, q, g, h)$ $\text{SK} = x$
- $\text{Enc}_{\text{PK}}(m)$: $m \in G$
 $y \xleftarrow{\$} \mathbb{Z}_q$
 $C = \langle g^y, h^y \cdot m \rangle$
- $\text{Dec}_{\text{SK}}(C)$:
 $C = \langle C_1, C_2 \rangle$
 $m = C_2 \cdot (C_1^{\text{SK}})^{-1}$

Correctness?

CPA Security?