CSCI 1515 Applied Cryptography

This Lecture:

- Anonymous Online Voting: An Overview
- Example: Diffie-Hellman Tuple (continued)
- Non-Interactive Zero-Knowledge (NIZK) Proof
- Fiat-Shamir Heuristic
- Homomorphism of ElGamal Encryption

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

Zero-Knowledge Proof of Knowledge

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

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

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

  $\forall x \notin L$, 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^* doesn't know w, then P^* cannot prove it.

- **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. $\forall (x,w) \in R_L,$

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

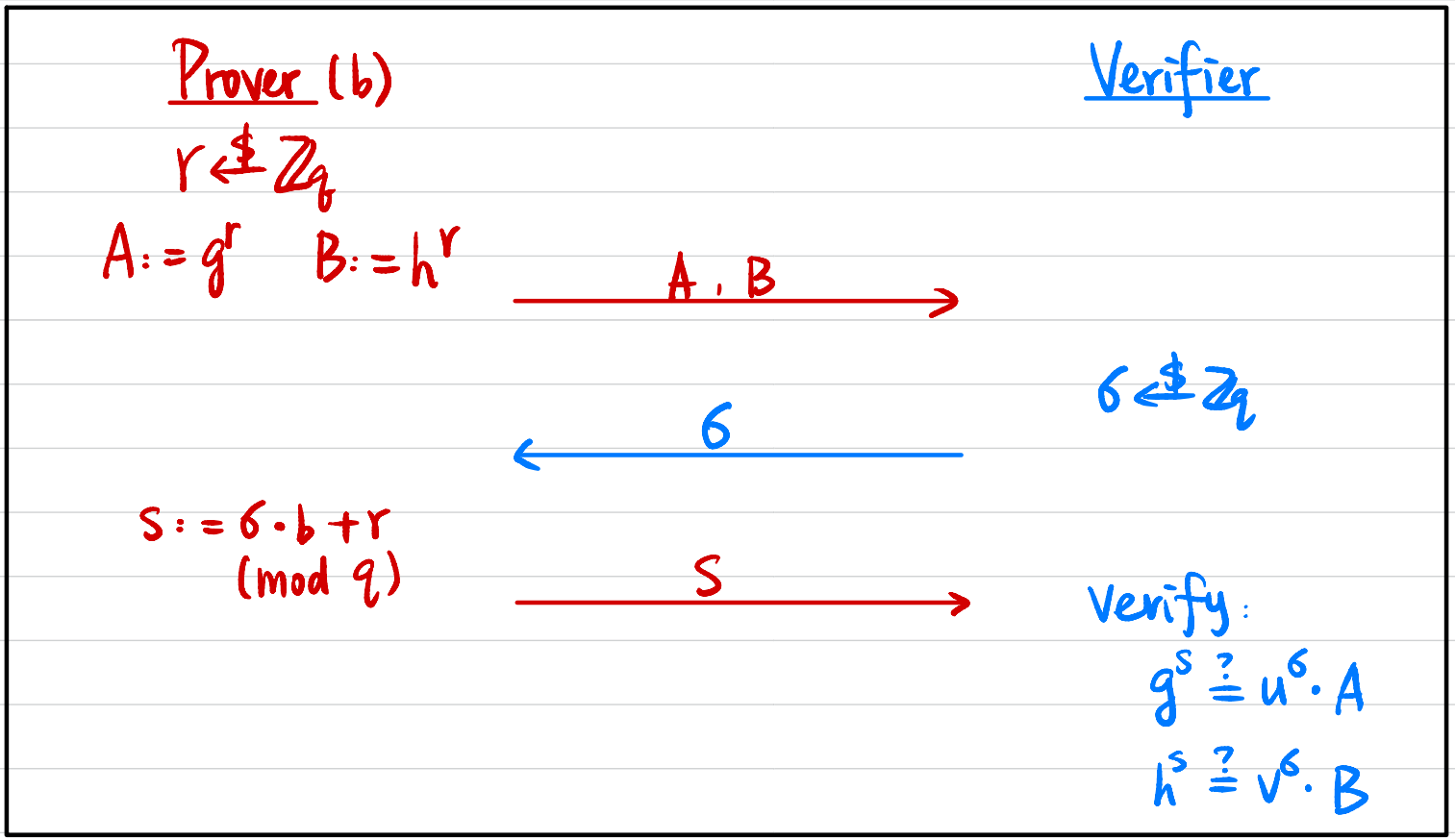
  A malicious V^* doesn't learn anything about w.

Example: 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. $u = g^b \wedge v = h^b$

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



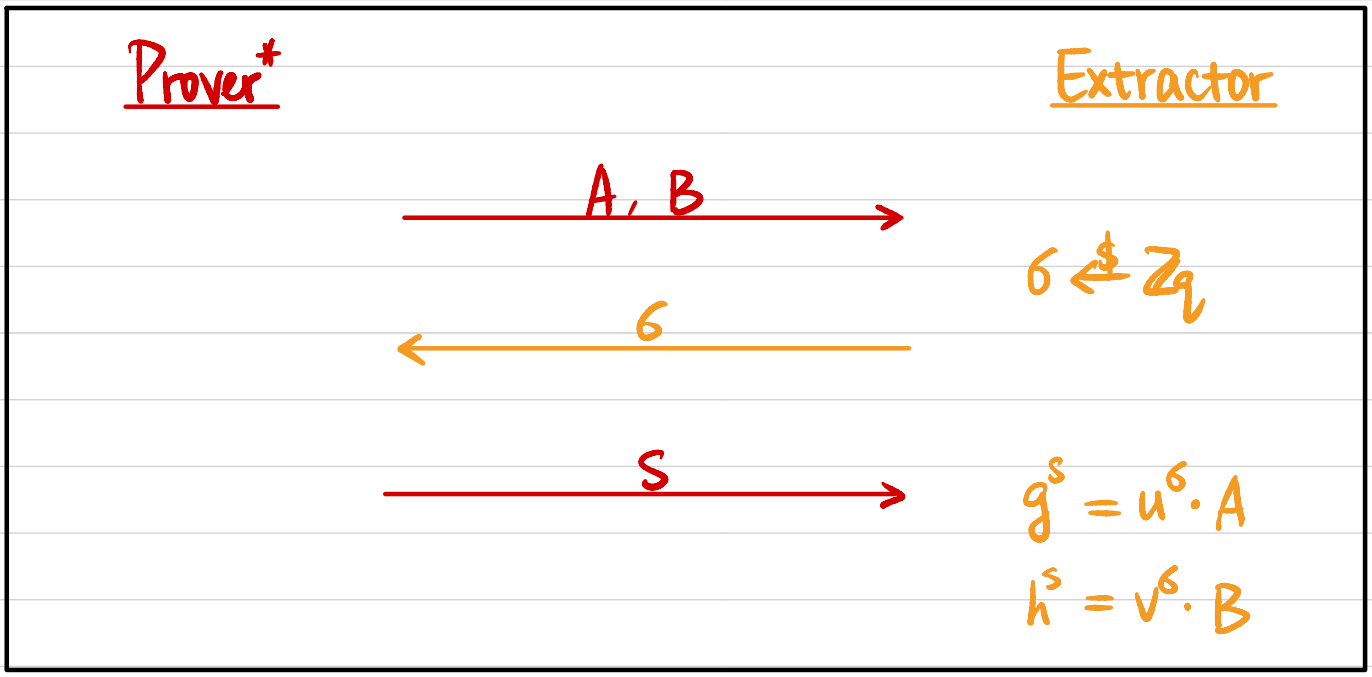
Completeness? $\forall (x,w) \in R_L \quad \Pr[P(x,w) \longleftrightarrow V(x) \text{ outputs } 1] = 1.$

Example: Diffie-Hellman Tuple

Proof of Knowledge?

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

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



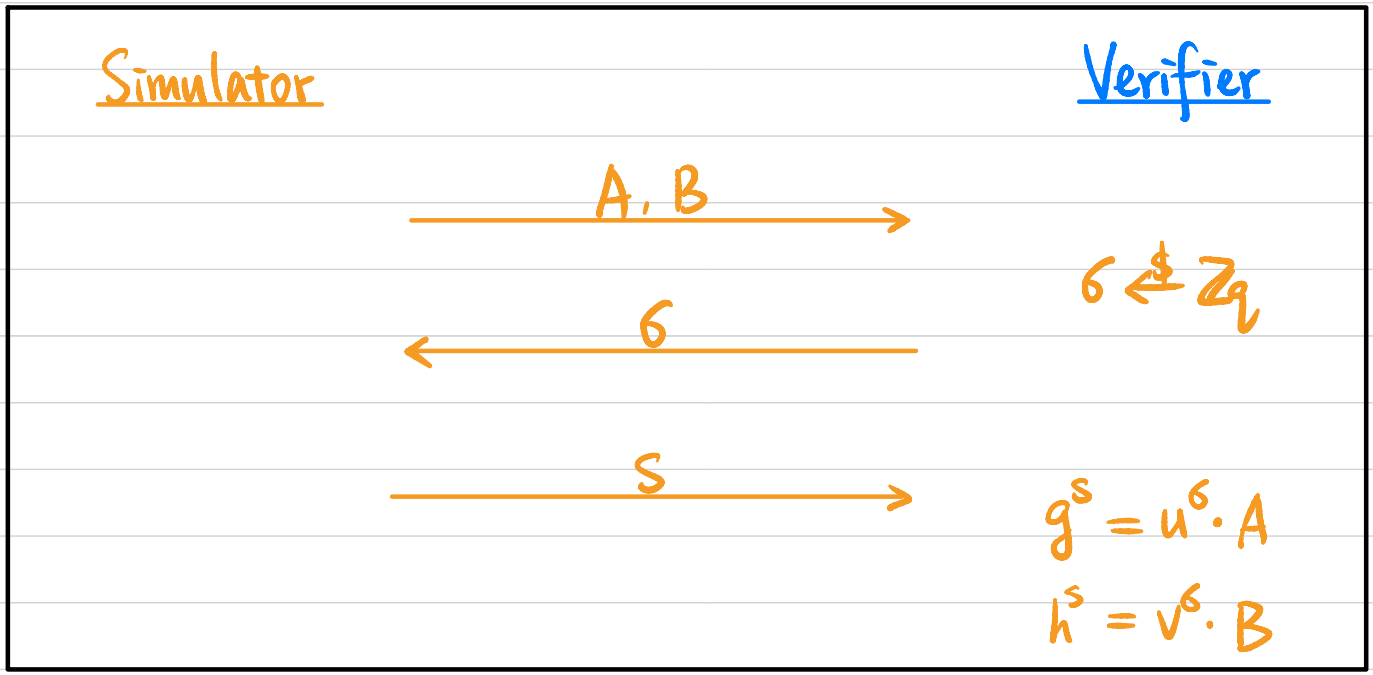
How to extract b s.t. $u = g^b \wedge v = h^b$?

Example: Diffie-Hellman Tuple

Honest-Verifier Zero-Knowledge (HVZK)?

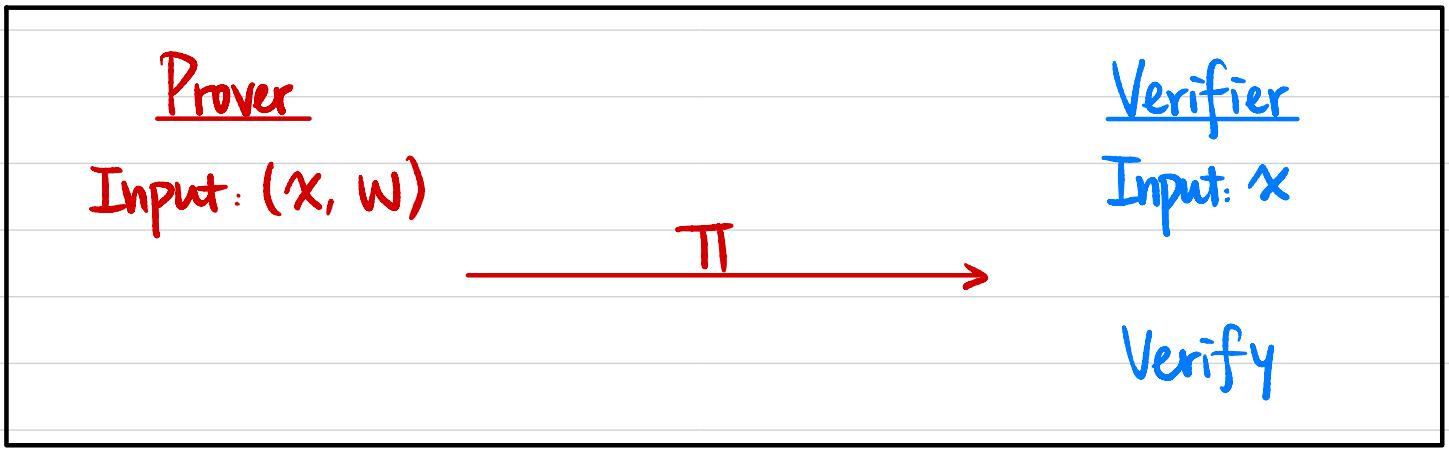
$\exists$ PPT S s.t. $\forall (x, w) \in R_L$,

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



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

Non-Interactive Zero-Knowledge (NIZK) Proof



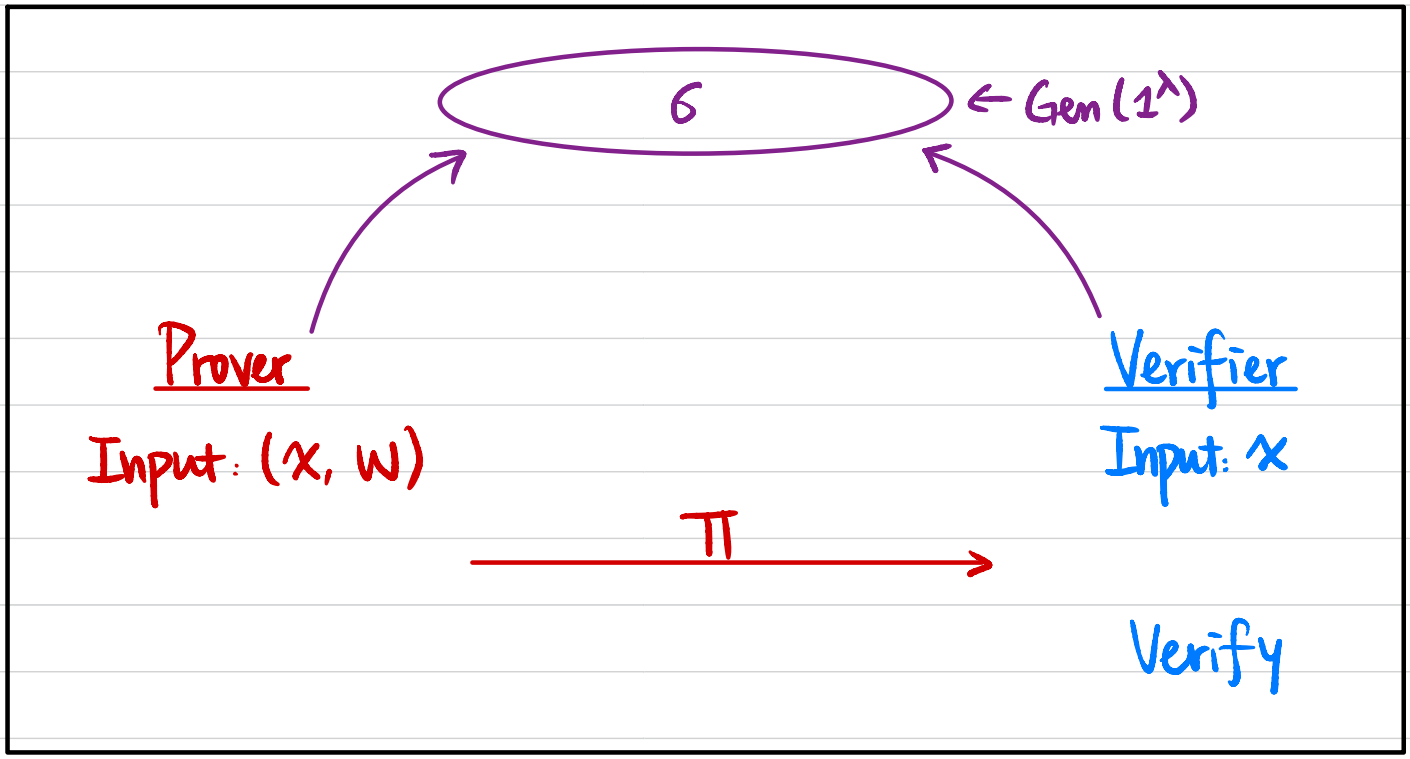
- **Completeness:** $\forall (x,w) \in R_L, \quad Pr[\, P(x,w) \longrightarrow V(x) \text{ outputs } 1\,] = 1.$
- **Soundness:** $\forall x \notin L, \ \forall P^*, \quad Pr[\, P^*(x) \longrightarrow V(x) \text{ outputs } 1\,] \simeq 0.$
- **Zero-Knowledge:** $\forall PPT\ V^*, \ \exists PPT\ S \quad s.t. \quad \forall (x,w) \in R_L,$

$$Output_{V^*}[P(x,w) \longrightarrow V^*(x)] \simeq S(x)$$

Is it possible?

Example: Diffie-Hellman Tuple (h, u, v)

Model 1: Common Random String / Common Reference String (CRS)



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

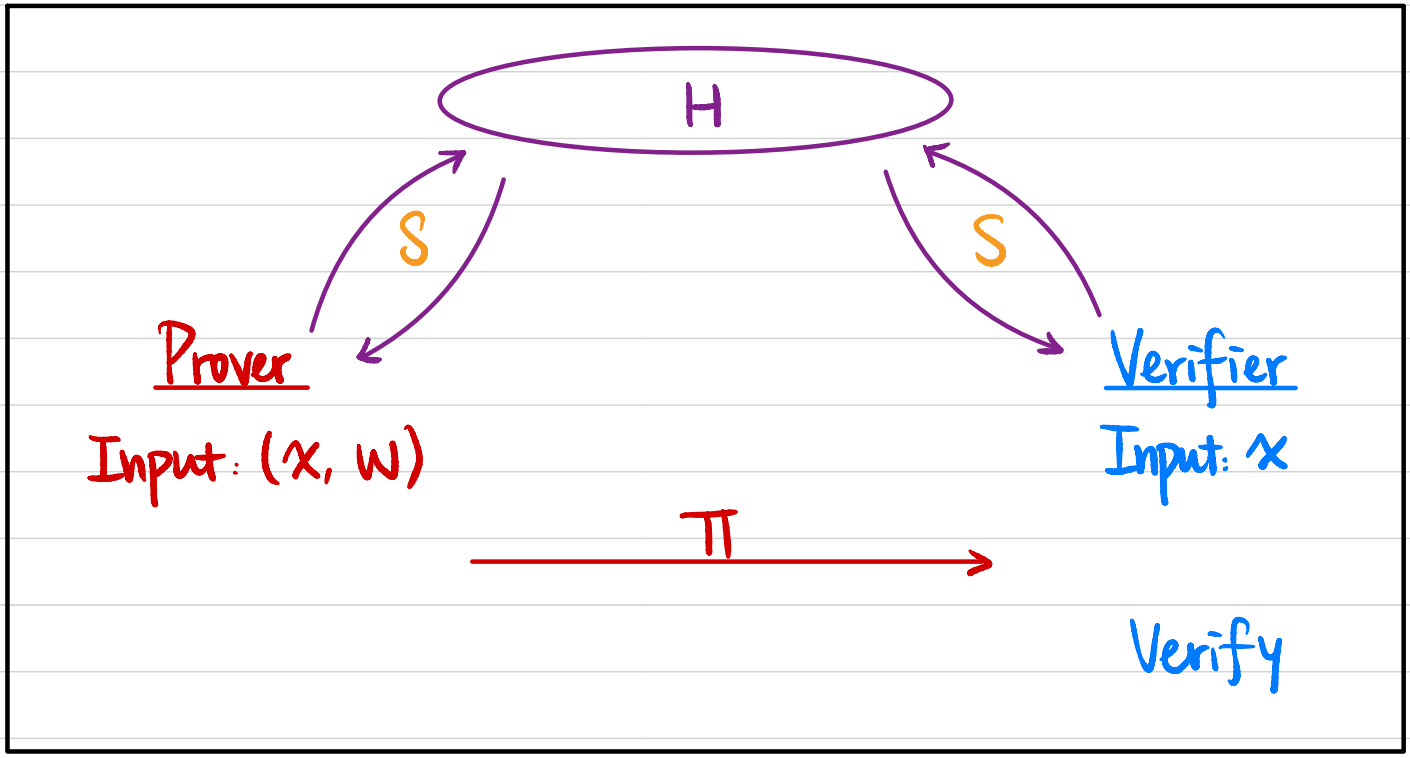
- **Zero-Knowledge:** $\forall PPT\ V^*,\ \exists PPT\ S$ s.t. $\forall (x, w) \in R_L,$

$$Output_{V^*}[\sigma \leftarrow Gen(1^\lambda),\ P(x, w, \sigma) \longrightarrow V^*(x, \sigma)] \simeq S(x)$$

Alternatively: $(\sigma \leftarrow Gen(1^\lambda),\ P(x, w, \sigma)) \simeq S(x)$

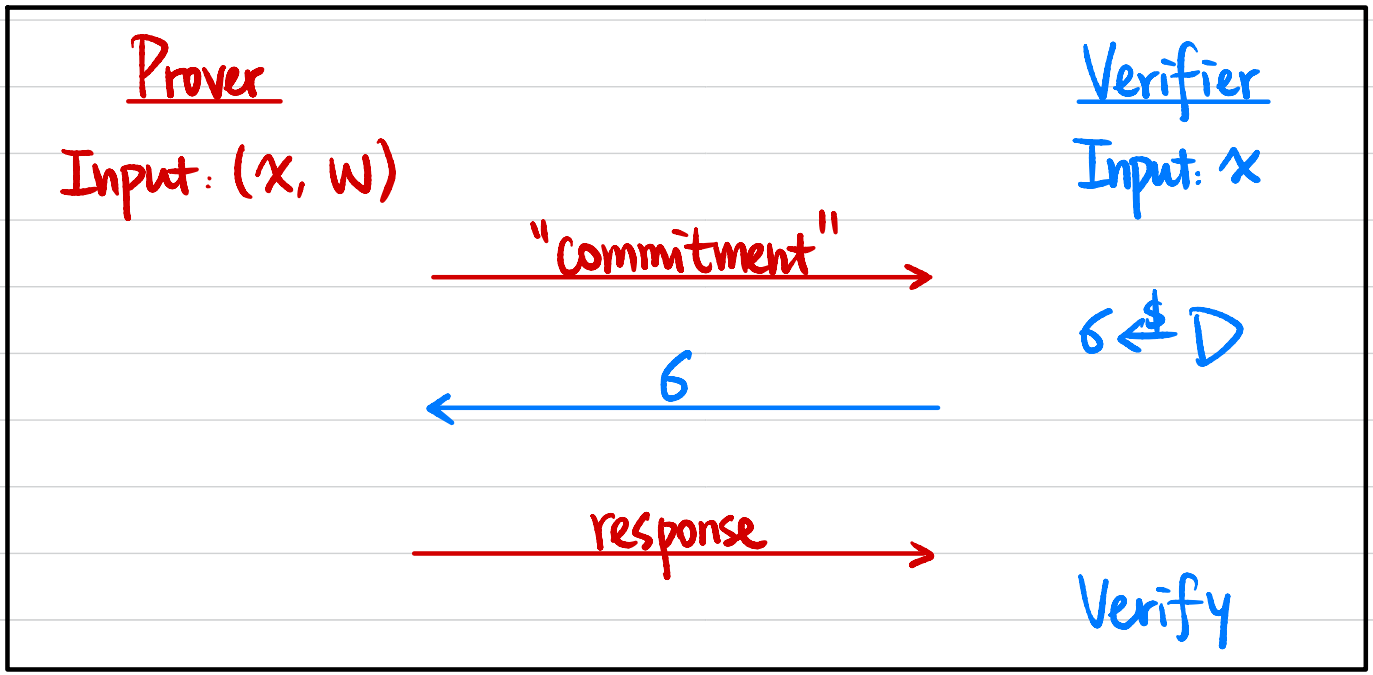
$S(x)$ generates both (σ, π)

Model 2: Random Oracle Model



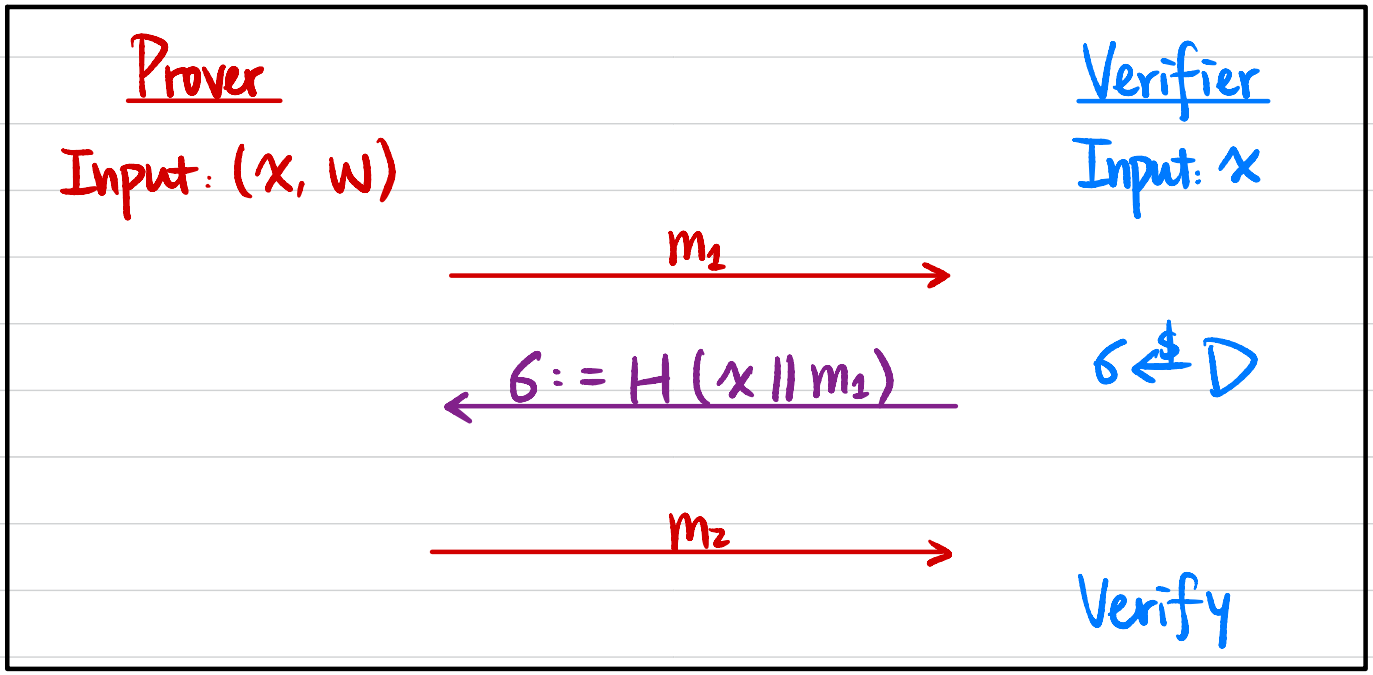
S controls input/output behavior of RO

Sigma Protocols Σ



Fiat-Shamir Heuristic

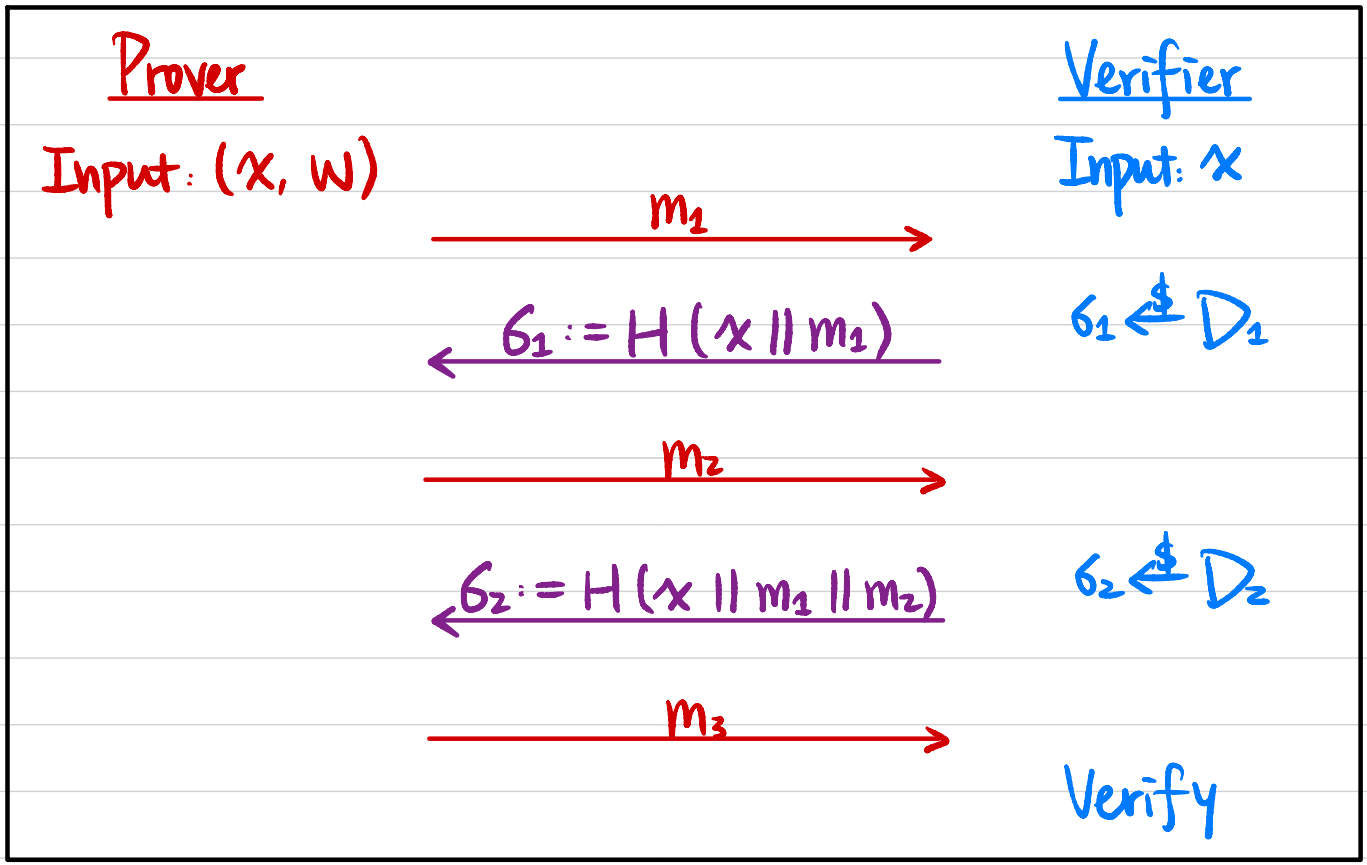
Sigma Protocol $\Rightarrow$ NIZK in the RO model



$\pi = (m_1, m_2)$

Fiat-Shamir Heuristic

Public-Coin HVZK $\Rightarrow$ NIZK in the RO model



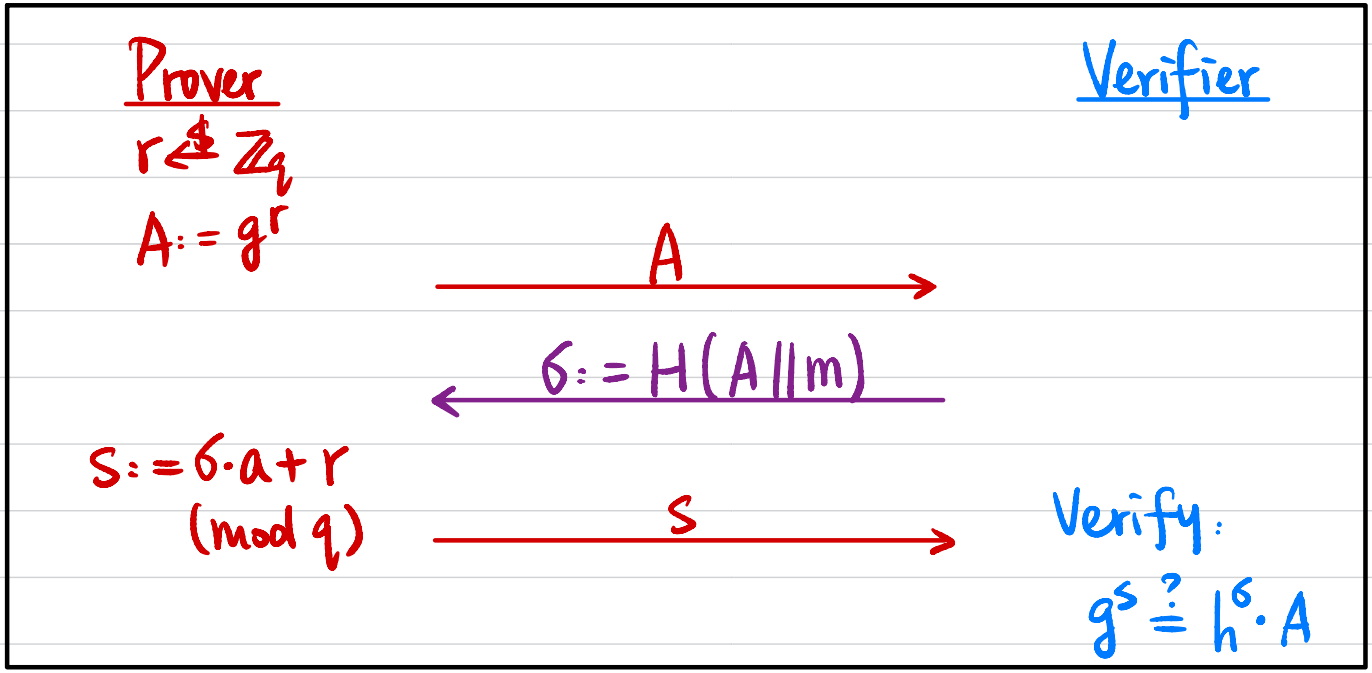
$\pi = (m_1, m_2, m_3)$

Fiat-Shamir Heuritic

Schnorr's Identification Protocol $\Rightarrow$ Schnorr's Signature in the RO model

Cyclic group G of order q, generator g

Public verification key $vk = g^a$; Secret signing key $sk = a$



To sign a message m: output (A, s)

Additively Homomorphic Encryption

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 + m_2)$

Additively Homomorphic

$Enc(m_1)$, $Enc(m_2)$ → $Enc(m_1 \cdot m_2)$

Multiplicatively Homomorphic

ElGamal Encryption: Cyclic group G with generator g, public key $pk = g^{sk}$.

$Enc_{pk}(m_1) = (g^{r_1}, pk^{r_1} \cdot m_1)$

$Enc_{pk}(m_2) = (g^{r_2}, pk^{r_2} \cdot m_2)$

→ $Enc(m_1 \cdot m_2)$?

Exponential ElGamal:

$Enc_{pk}(m_1) = (g^{r_1}, pk^{r_1} \cdot g^{m_1})$

$Enc_{pk}(m_2) = (g^{r_2}, pk^{r_2} \cdot g^{m_2})$

→ $Enc(m_1 + m_2)$?

Correctness of Encryption

Given a cyclic group G of order q with generator g.

Public key $pk \in G$. ← public

Ciphertext $C = (c_1, c_2)$ ← public

ZKP for an OR statement:

C is an encryption of 0 OR C is an encryption of 1

Witness: randomness r used in encryption

↑ secret

$$R_L = \{ ((pk, c_1, c_2), r) : (c_1 = g^r \wedge c_2 = pk^r) \vee (c_1 = g^r \wedge c_2 = pk^r \cdot g) \}$$

↑ (public) statement — ↑ (secret) witness

Correctness of Encryption

C is an encryption of 0

Witness: randomness r used in encryption

$$R_{L_0} = \{ ((pk, c_1, c_2), r) : c_1 = g^r \wedge c_2 = pk^r \}$$

(public) statement

(secret) witness

C is an encryption of 1

Witness: randomness r used in encryption

$$R_{L_1} = \{ ((pk, c_1, c_2), r) : c_1 = g^r \wedge c_2 = pk^r \cdot g \}$$

(public) statement

(secret) witness