

# 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

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

ElGamal

ZKP

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

⋮  
⋮  
⋮

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



$\text{Enc}(\sum v_i)$



Decrypt to  $\sum v_i$

# 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:**  $\exists \text{PPT } E \text{ s.t. } \forall P^*, \forall x,$

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

If  $P^*$  doesn't know  $w$ , then  $P^*$  cannot prove it.

• **Honest-Verifier Zero-Knowledge (HVZK):**  $\exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$

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

An honest  $V$  doesn't learn anything about  $w$ .

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

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

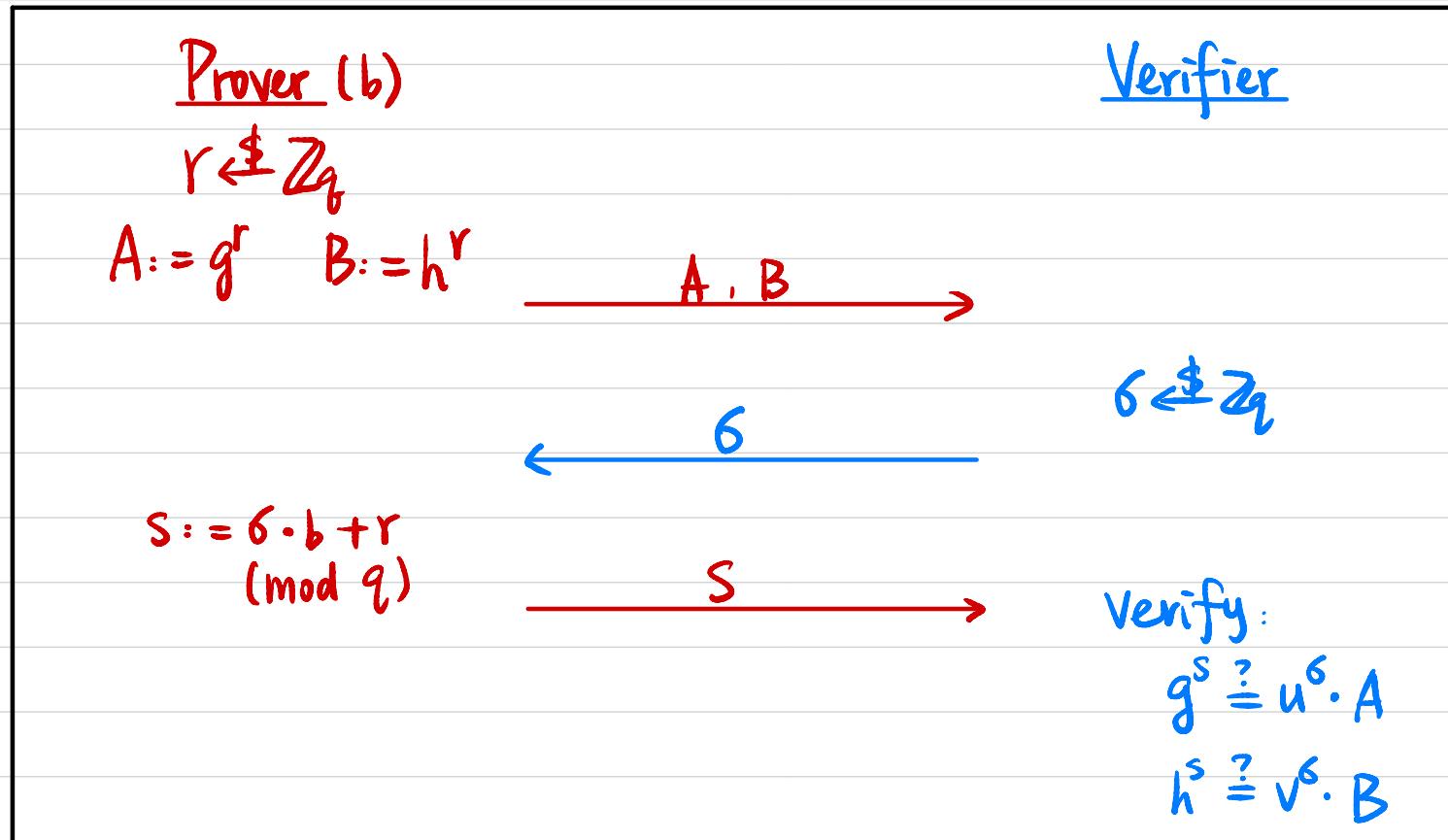
A malicious  $V^*$  doesn't learn anything about  $w$ .

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

$$g^s = g^{\delta \cdot b + r} = (g^b)^\delta \cdot g^r = u^\delta \cdot A$$

$$h^s = h^{\delta \cdot b + r} = (h^b)^\delta \cdot h^r = v^\delta \cdot B$$

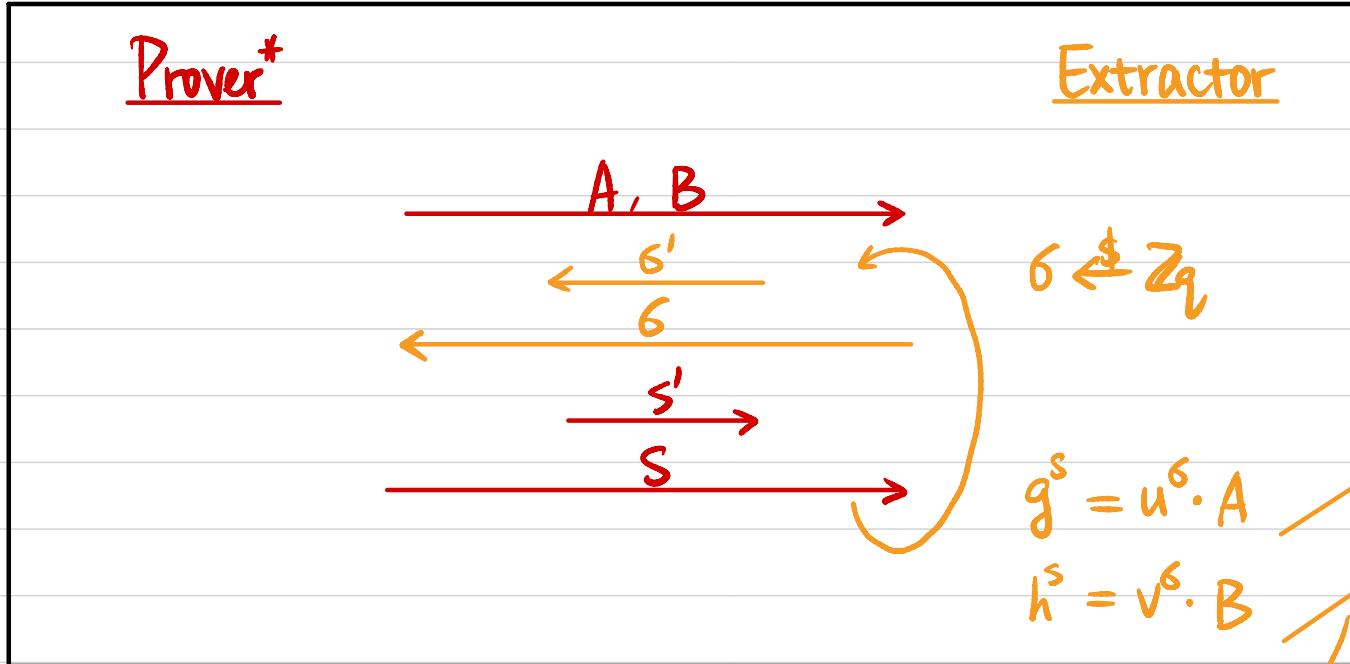
## Example: Diffie-Hellman Tuple

Proof of Knowledge?

$$b \text{ s.t. } u = g^b \wedge v = h^b$$

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

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



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

$$b = (s - s')(s - s')^{-1} \pmod q$$

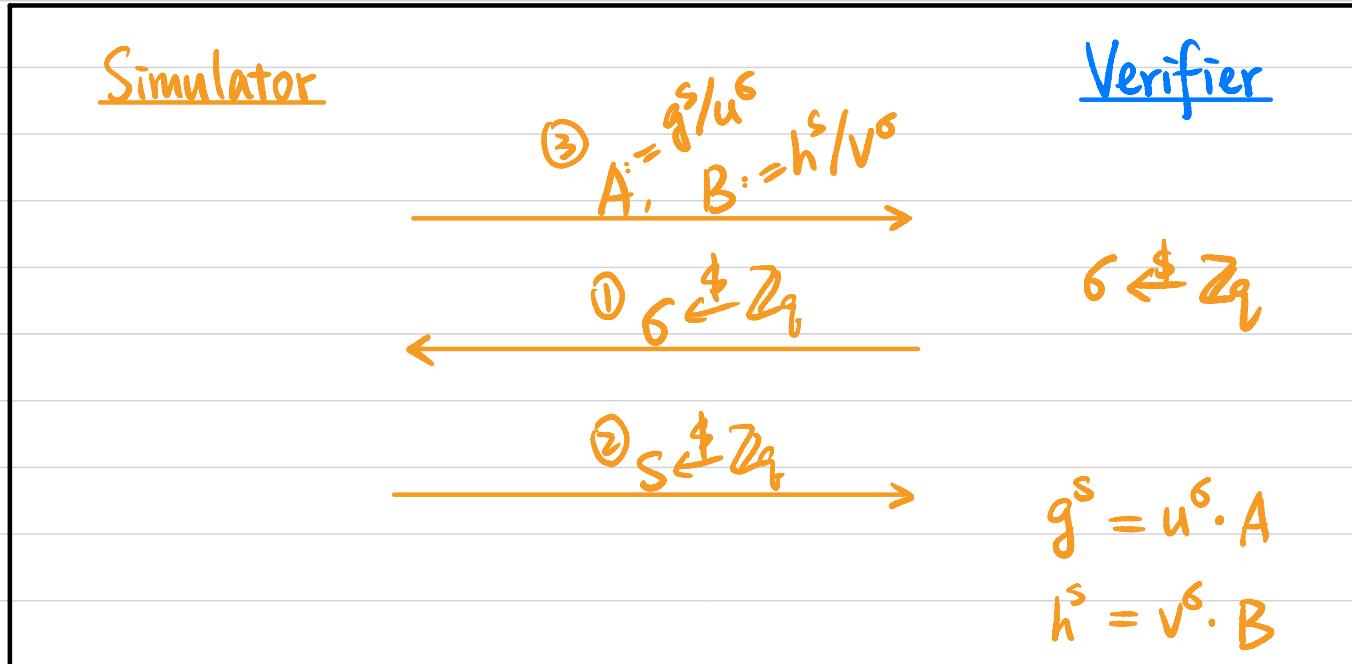
$$\begin{aligned}
 & g^{s-s'} = u^{s-s'} \\
 & h^{s-s'} = v^{s-s'} \\
 & h^{(s-s')(s-s')^{-1}} = v \\
 & g^{(s-s')(s-s')^{-1}} = u
 \end{aligned}$$

## Example: Diffie-Hellman Tuple

Honest-Verifier Zero-Knowledge (HVZK) ?

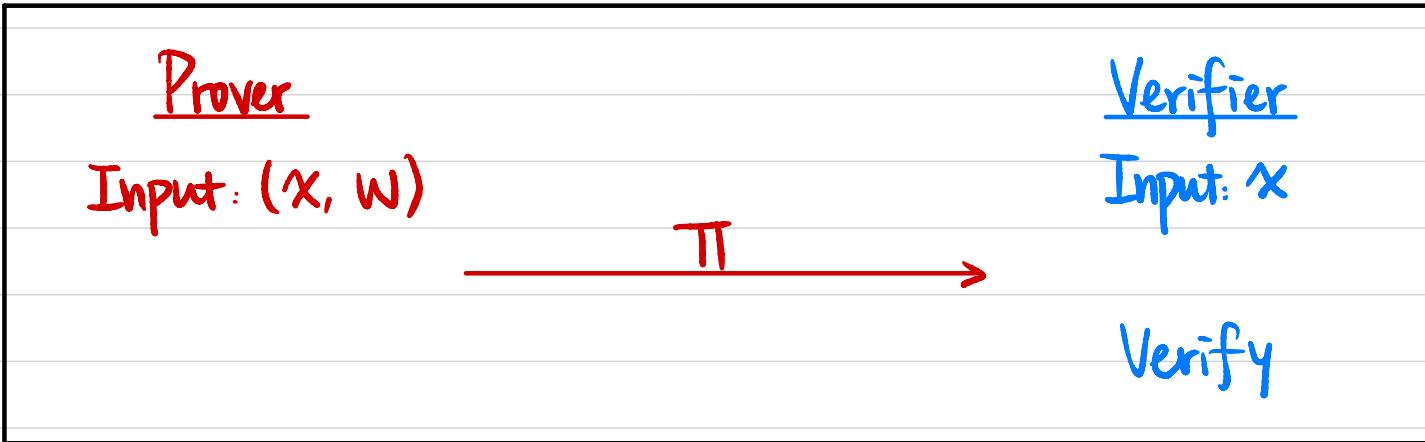
$\exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$

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



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

# Non-Interactive Zero-Knowledge (NIZK) Proof



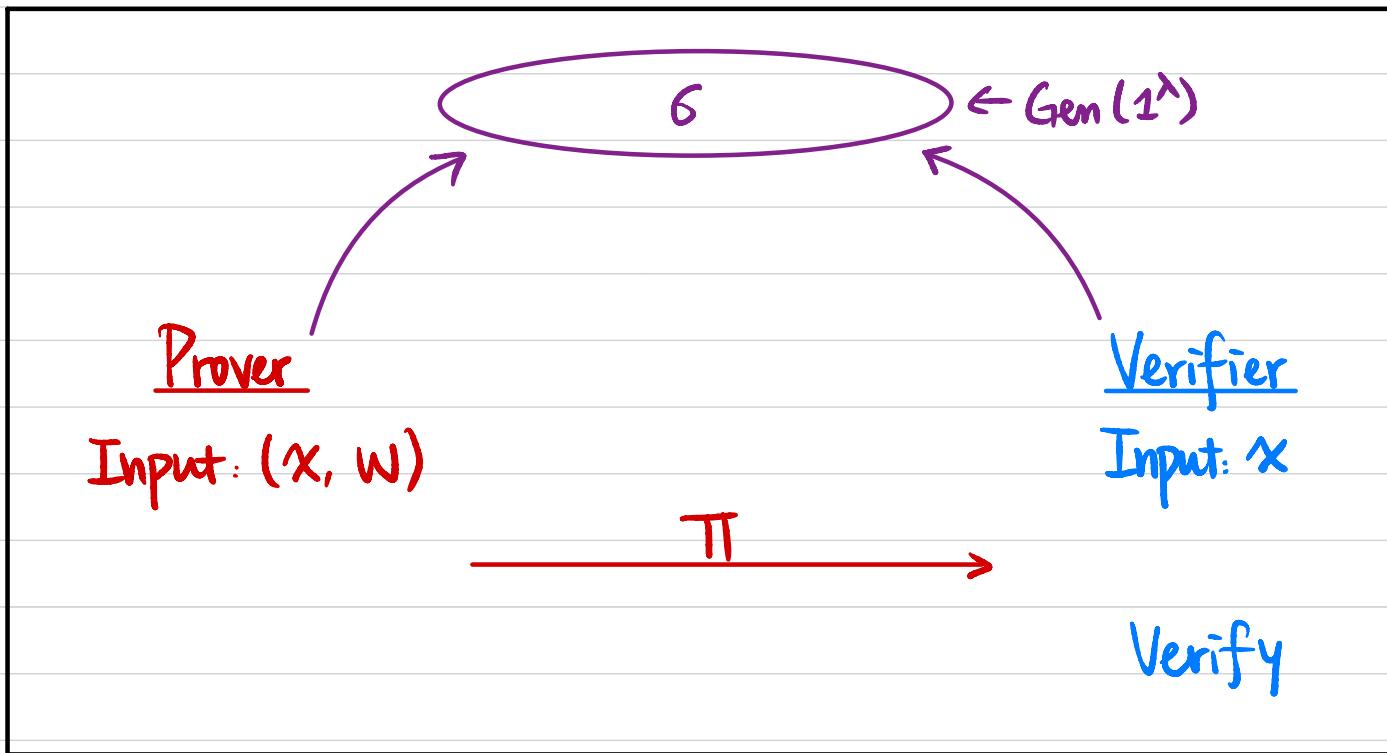
- Completeness:  $\forall (x, w) \in R_L, \Pr[P(x, w) \rightarrow V(x) \text{ outputs } 1] = 1$ .
- Soundness:  $\forall x \notin L, \forall P^*, \Pr[P^*(x) \rightarrow V(x) \text{ outputs } 1] \approx 0$ .
- Zero-Knowledge:  $\forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (x, w)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

Is it possible? NOT in the "plain" model

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

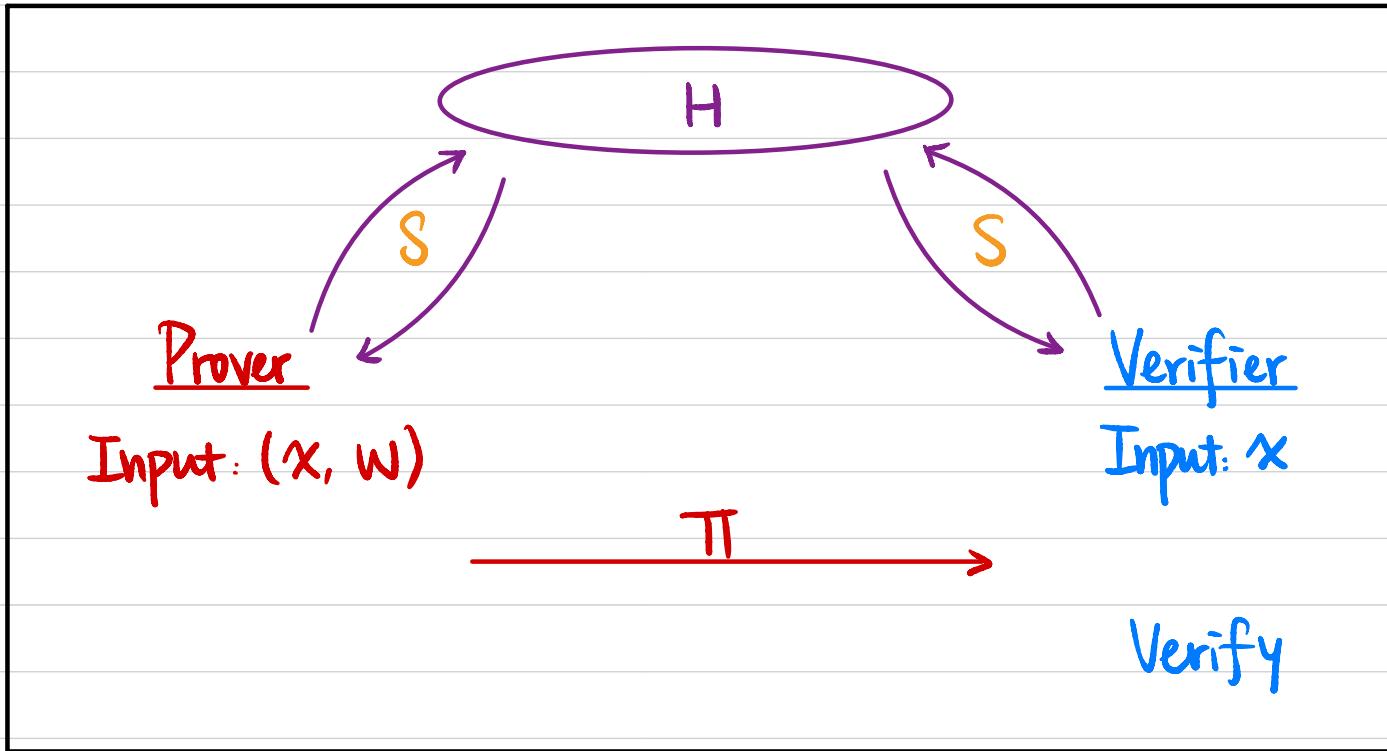
If  $(h, u, v)$  is a DH tuple, then  $S(h, u, v)$  outputs a valid proof.  $\Rightarrow$  break DDH  
If  $(h, u, v)$  is not a DH tuple, then  $S(h, u, v)$  cannot find a valid proof.

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



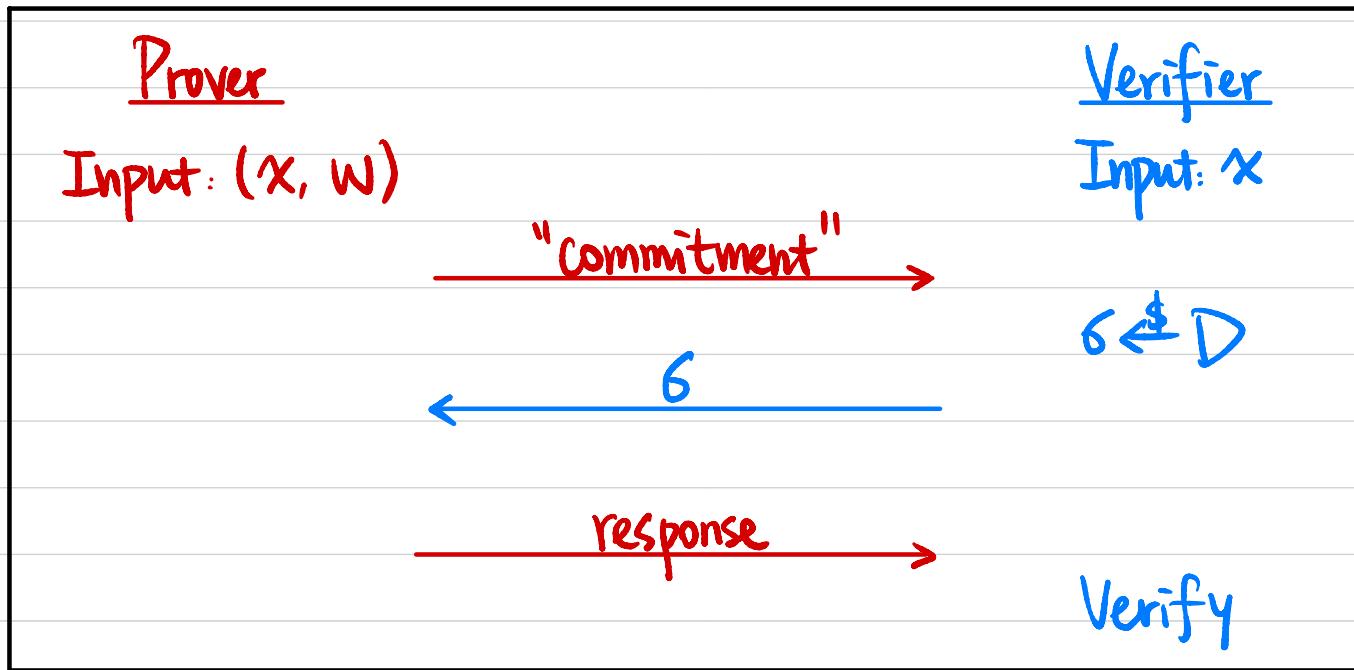
- **Soundness:**  $\forall x \in L, \forall P^*, \Pr[\sigma \leftarrow \text{Gen}(1^\lambda), P^*(\sigma, x) \rightarrow V(\sigma, x) \text{ outputs } 1] \approx 0$ .
- **Zero-Knowledge:**  $\forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$   
 $\text{Output}_{V^*}[\sigma \leftarrow \text{Gen}(1^\lambda), P(x, w, \sigma) \rightarrow V^*(x, \sigma)] \approx S(x)$   
Alternatively:  $(\sigma \leftarrow \text{Gen}(1^\lambda), P(x, w, \sigma)) \approx S(x)$   
 $S(x)$  generates both  $(\sigma, \pi)$

## Model 2: Random Oracle Model



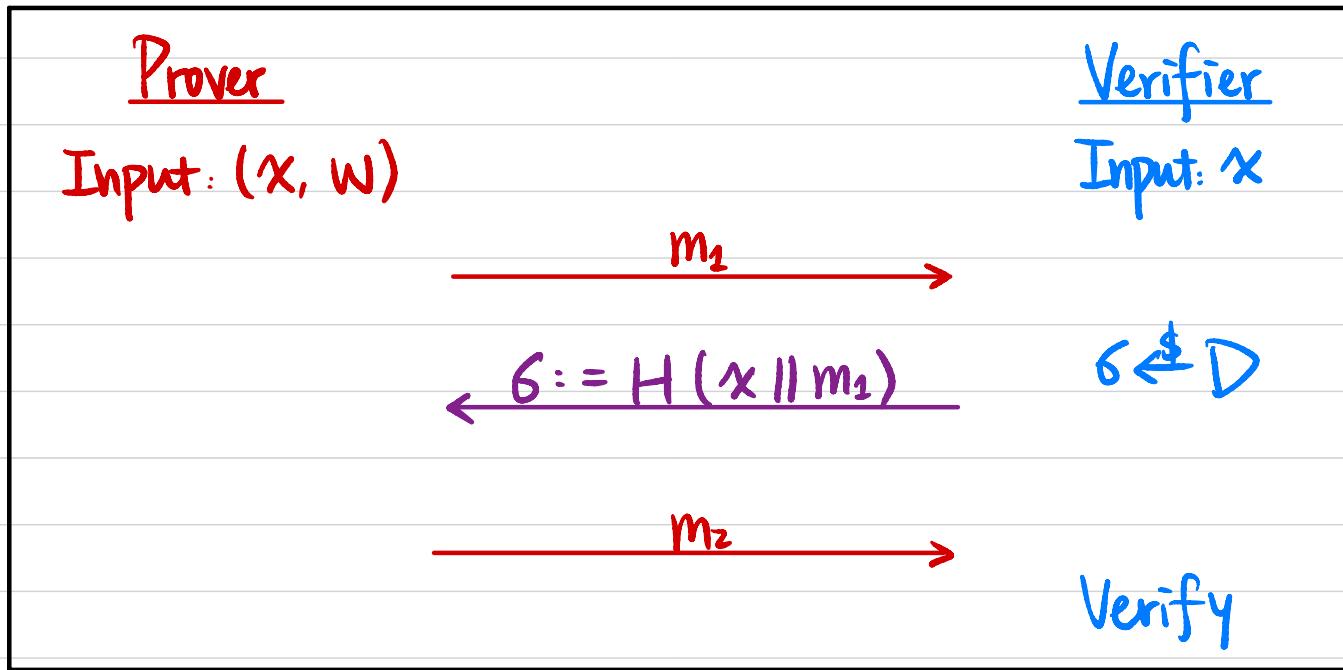
$S$  controls input/output behavior of RO

# Sigma Protocols $\Sigma$



## Fiat-Shamir Heuristic

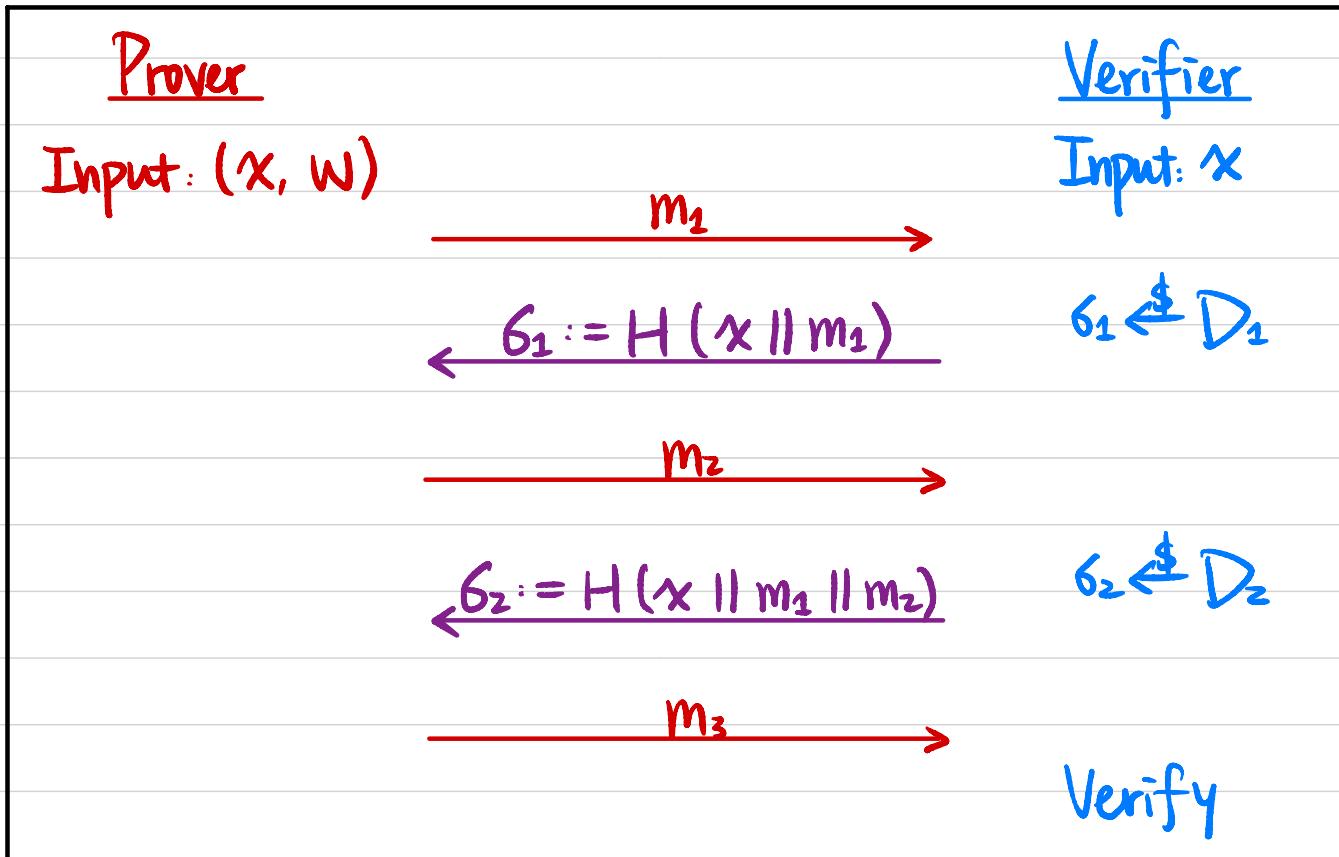
Sigma Protocol  $\Rightarrow$  NIZK in the RO model



$$\Pi = (m_1, m_2)$$

## Fiat-Shamir Heuristic

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



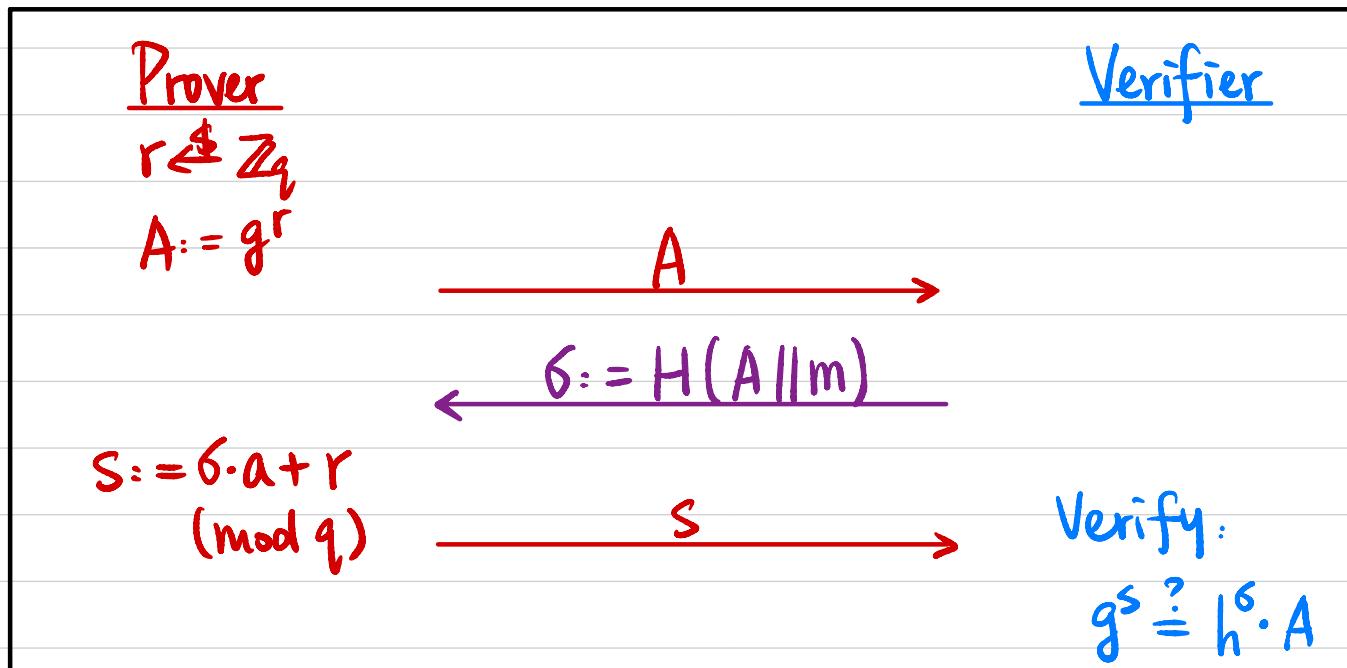
$$\Pi = (m_1, m_2, m_3)$$

## Fiat-Shamir Heuristic

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

$$\begin{array}{ccc} \text{Enc}(m_1) & \xrightarrow{\quad} & \text{Enc}(m_1 + m_2) \\ \text{Enc}(m_2) & \xrightarrow{\quad} & \end{array}$$

Additively Homomorphic

$$\begin{array}{ccc} \text{Enc}(m_1) & \xrightarrow{\quad} & \text{Enc}(m_1 \cdot m_2) \\ \text{Enc}(m_2) & \xrightarrow{\quad} & \end{array}$$

Multiplicatively Homomorphic

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

$$\begin{array}{l} \text{Enc}_{\text{pk}}(m_1) = (g^{r_1}, \text{pk}^{r_1} \cdot m_1) \xrightarrow{\quad} \text{Enc}(m_1 \cdot m_2) ? \Rightarrow (g^{r_1+r_2}, \text{pk}^{r_1+r_2} \cdot (m_1 \cdot m_2)) \\ \text{Enc}_{\text{pk}}(m_2) = (g^{r_2}, \text{pk}^{r_2} \cdot m_2) \xrightarrow{\quad} \end{array}$$

Exponential ElGamal :

$$\begin{array}{l} \text{Enc}_{\text{pk}}(m_1) = (g^{r_1}, \text{pk}^{r_1} \cdot g^{m_1}) \xrightarrow{\quad} \text{Enc}(m_1 + m_2) ? \Rightarrow (g^{r_1+r_2}, \text{pk}^{r_1+r_2} \cdot g^{m_1+m_2}) \\ \text{Enc}_{\text{pk}}(m_2) = (g^{r_2}, \text{pk}^{r_2} \cdot g^{m_2}) \end{array}$$

## Correctness of Encryption

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

Public key  $pk \in G$ .  $\leftarrow$  public

Ciphertext  $c = (c_1, c_2)$   $\leftarrow$

ZKP for an OR statement:

$c$  is an encryption of 0 OR  $c$  is an encryption of 1

Witness: randomness  $r$  used in encryption  
 $\uparrow$   
secret

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

$\uparrow$  (public)  
Statement  $\uparrow$  (secret)  
Witness

## Correctness of Encryption

C is an encryption of 0

$$(h, u, v) = (g^a, g^b, g^{ab})$$

$$b \text{ s.t. } u = g^b \wedge v = h^b$$

Witness: randomness r used in encryption

$$R_{L0} = \{ ((pk, c_1, c_2), r) : c_1 = g^r \wedge c_2 = pk^r \}$$

(public)  
Statement      (secret)  
Witness

Diffie-Hellman Tuple

C is an encryption of 1

Witness: randomness r used in encryption

$$R_{L1} = \{ ((pk, c_1, c_2), r) : c_1 = g^r \wedge c_2 = pk^r \cdot g \}$$

(public)  
Statement      (secret)  
Witness

$$c_2/g = pk^r$$

$((pk, c_1, c_2/g), r) \leftarrow$  Diffie-Hellman Tuple