CSCI 1515 Applied Cryptography

This Lecture:

- Sigma Protocol and Examples (Continued)
- Proving AND/OR Statements
- Non-Interactive Zero-Knowledge (NIZK) Proof
- Fiat-Shamir Heuristic
- Putting it Together: Anonymous Online Voting
- ElGamal Encryption: Homomorphism and Threshold Decryption

Zero-Knowledge Proof of Knowledge

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

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

- **Proof of Knowledge:** $\exists$ PPT E s.t. $\forall P^*$, $\forall x$,

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

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

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

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

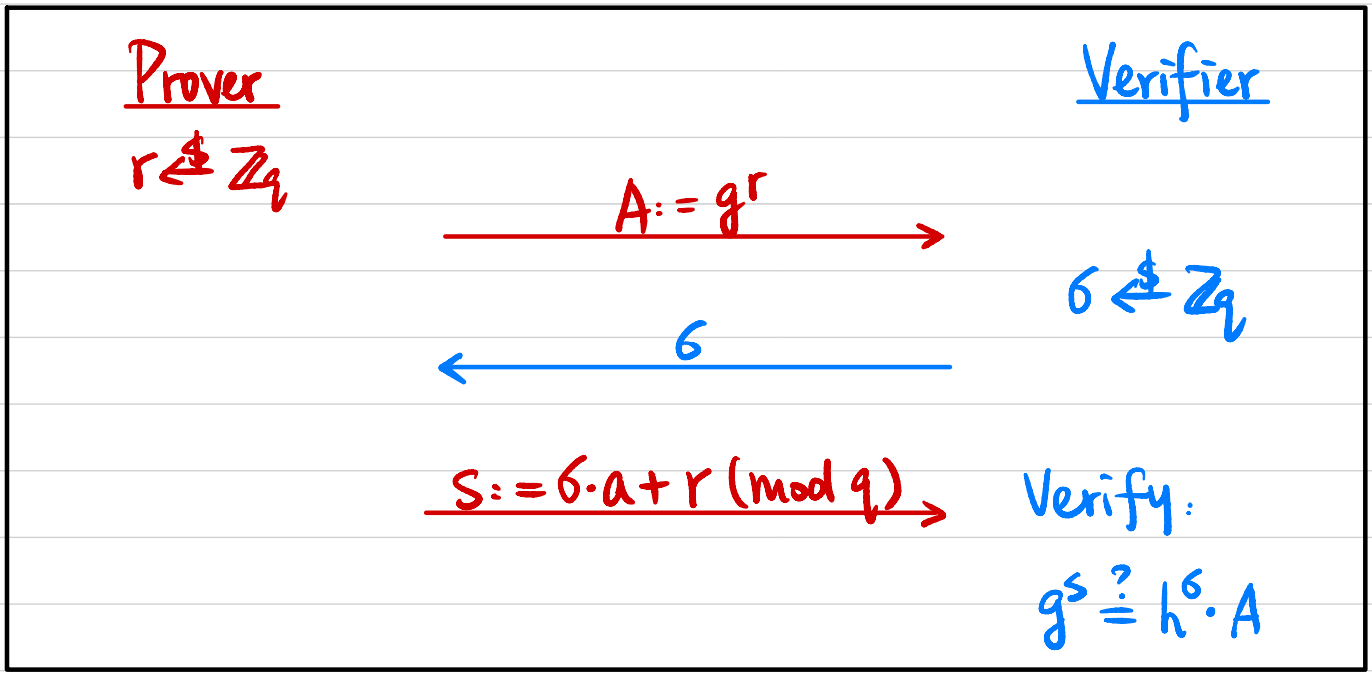
$$\text{Output}_{V^*}[P(x,w) \longleftrightarrow V^*(x)] \simeq S(x)$$

Example 1: Schnorr's Identification Protocol

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

Witness: a

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



Completeness?

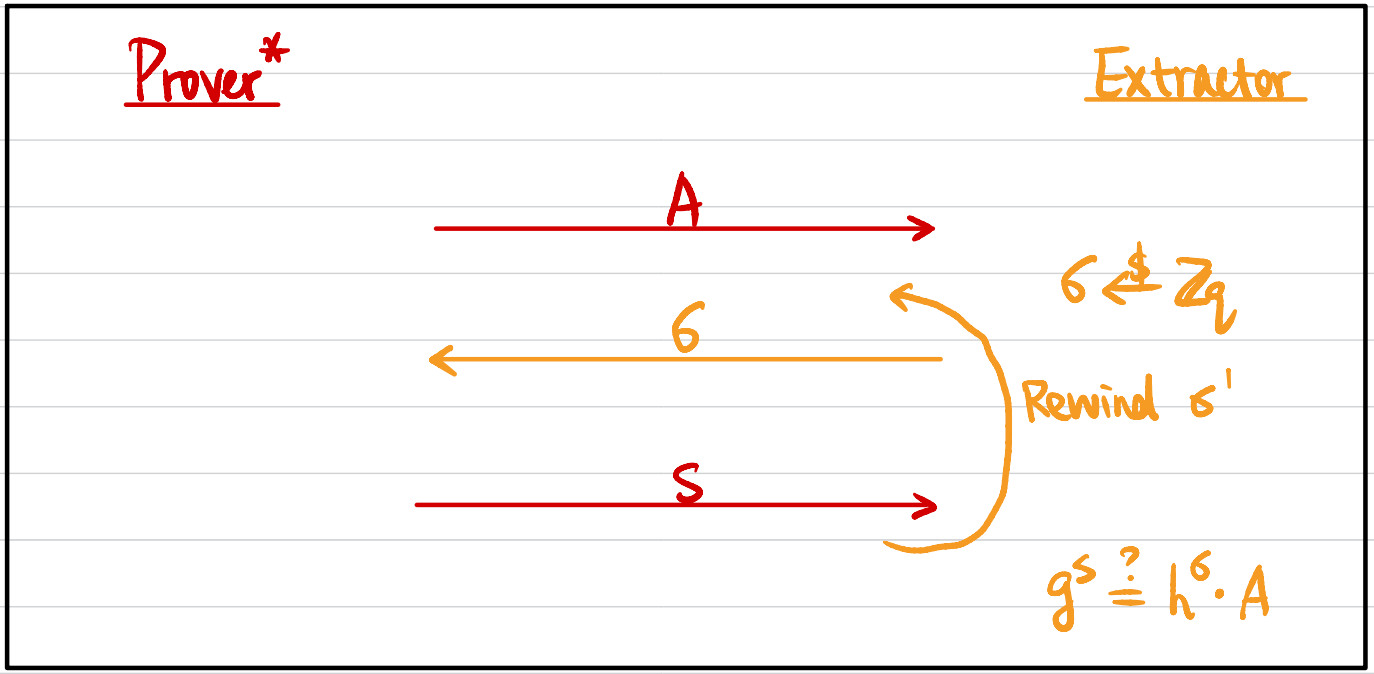
$g^s = g^{\sigma \cdot a + r}$

$h^\sigma \cdot A = (g^a)^\sigma \cdot g^r = g^{\sigma \cdot a + r}$

$\Rightarrow$ Verifier always outputs 1

Proof of Knowledge?

Extract a s.t. $h = g^a$?



$\sigma \Rightarrow s$ s.t. $g^s = h^\sigma \cdot A$

$\sigma' \Rightarrow s'$ s.t. $g^{s'} = h^{\sigma'} \cdot A$

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

$\Downarrow$

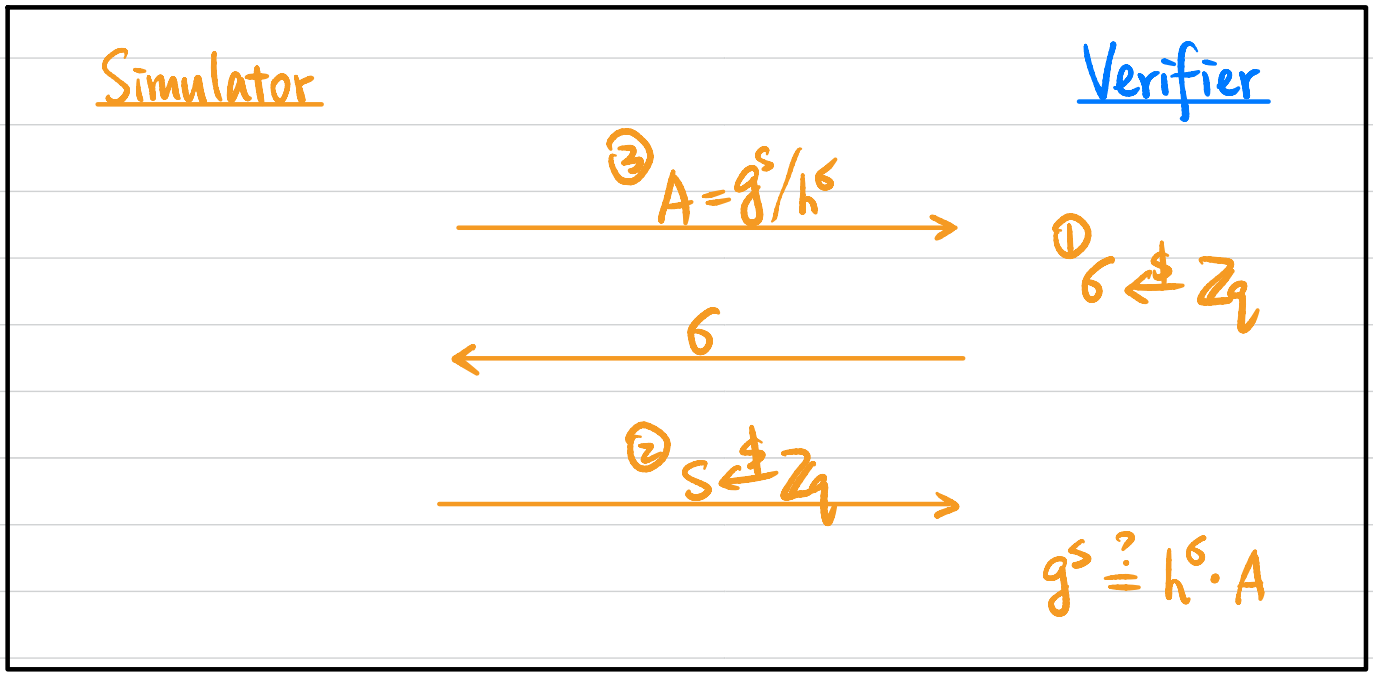
$g^{(s-s')(\sigma-\sigma')^{-1}} = h$

$\Downarrow$

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

Honest Verifier Zero Knowledge (HVZK)

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

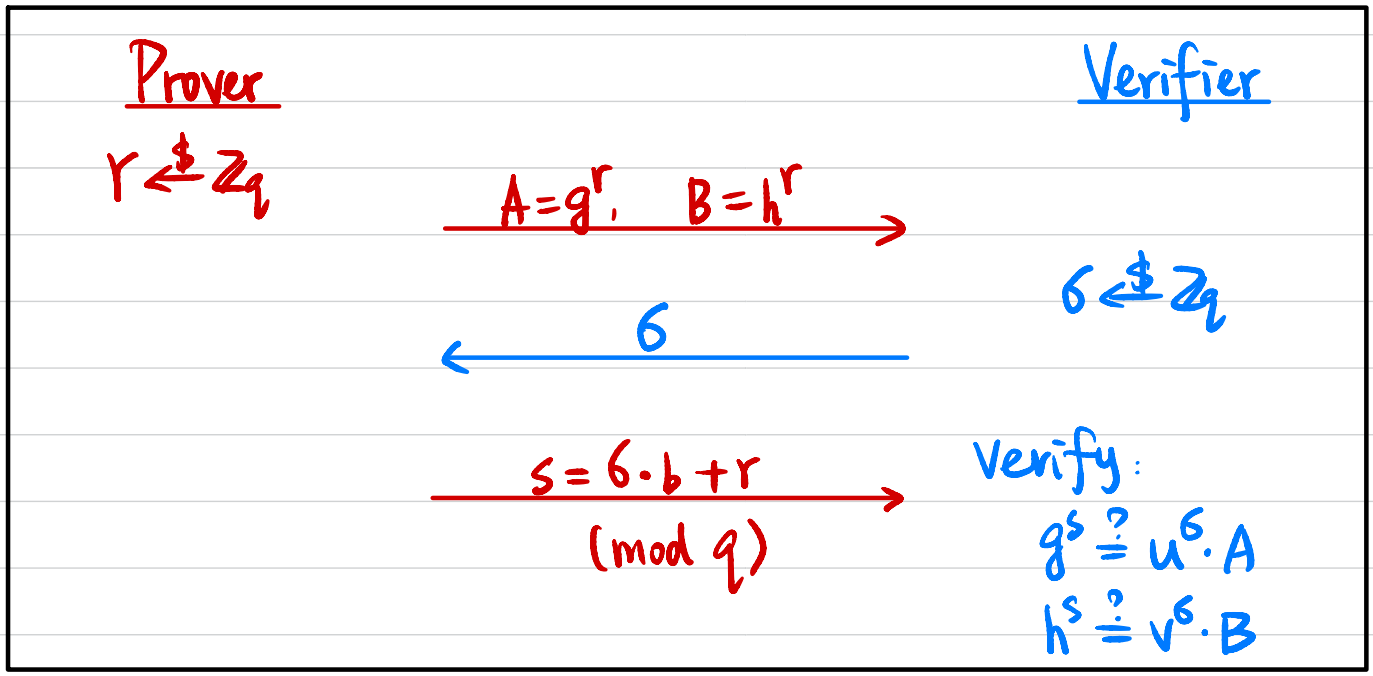


Example 2: Chaum-Pedersen Protocol for DH Tuple

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

Witness: b

Statement: $\exists\, b \in \mathbb{Z}_q$ s.t. $u = g^b \wedge v = h^b$

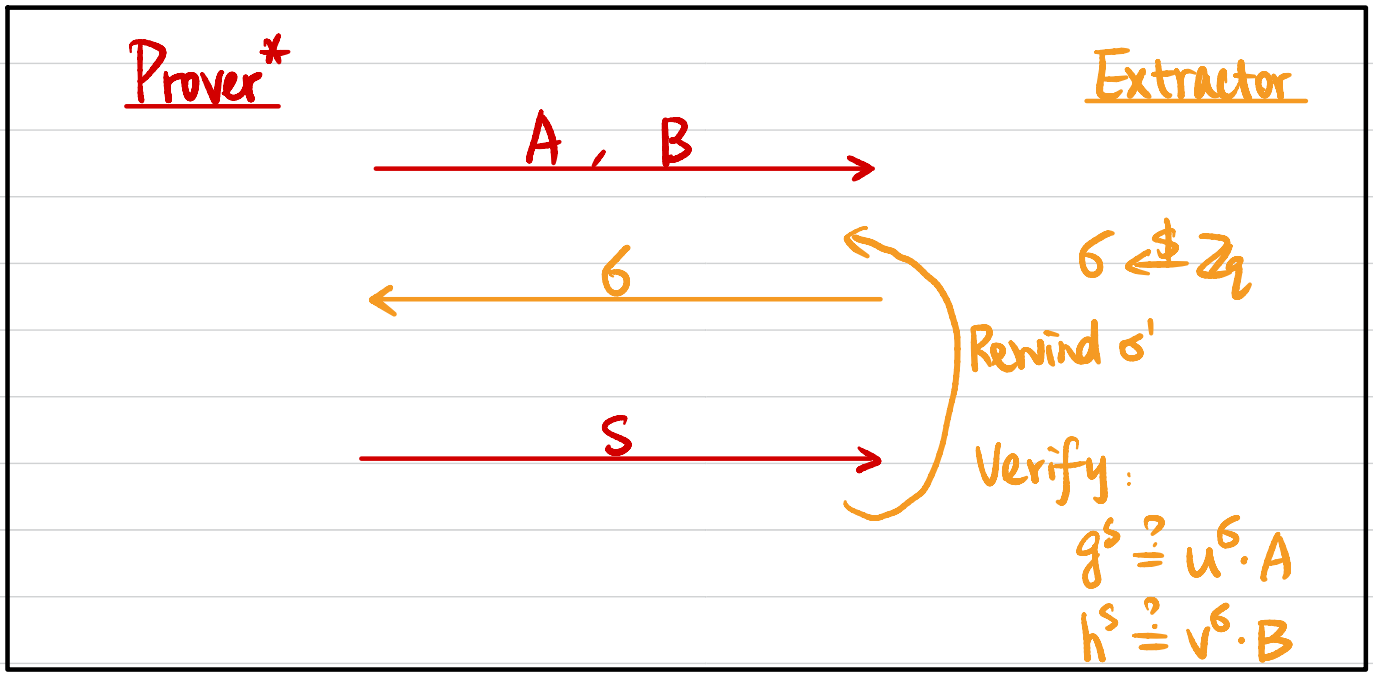


Completeness? $g^s = g^{c \cdot b + r}$ $\quad h^s = h^{c \cdot b + r}$ $\Rightarrow$ Verifier always outputs 1

$u^c \cdot A = (g^b)^c \cdot g^r = g^{c \cdot b + r}$ $\quad v^c \cdot B = (h^b)^c \cdot h^r = h^{c \cdot b + r}$

Proof of Knowledge?

Extract b s.t. $u = g^b \wedge v = h^b$?



$\sigma \Rightarrow s$ s.t. $g^s = u^\sigma \cdot A$, $h^s = v^\sigma \cdot B$

$\sigma' \Rightarrow s'$ s.t. $g^{s'} = u^{\sigma'} \cdot A$, $h^{s'} = v^{\sigma'} \cdot B$

$\Downarrow$

$g^{s-s'} = u^{\sigma-\sigma'}$, $h^{s-s'} = v^{\sigma-\sigma'}$

$\Downarrow$

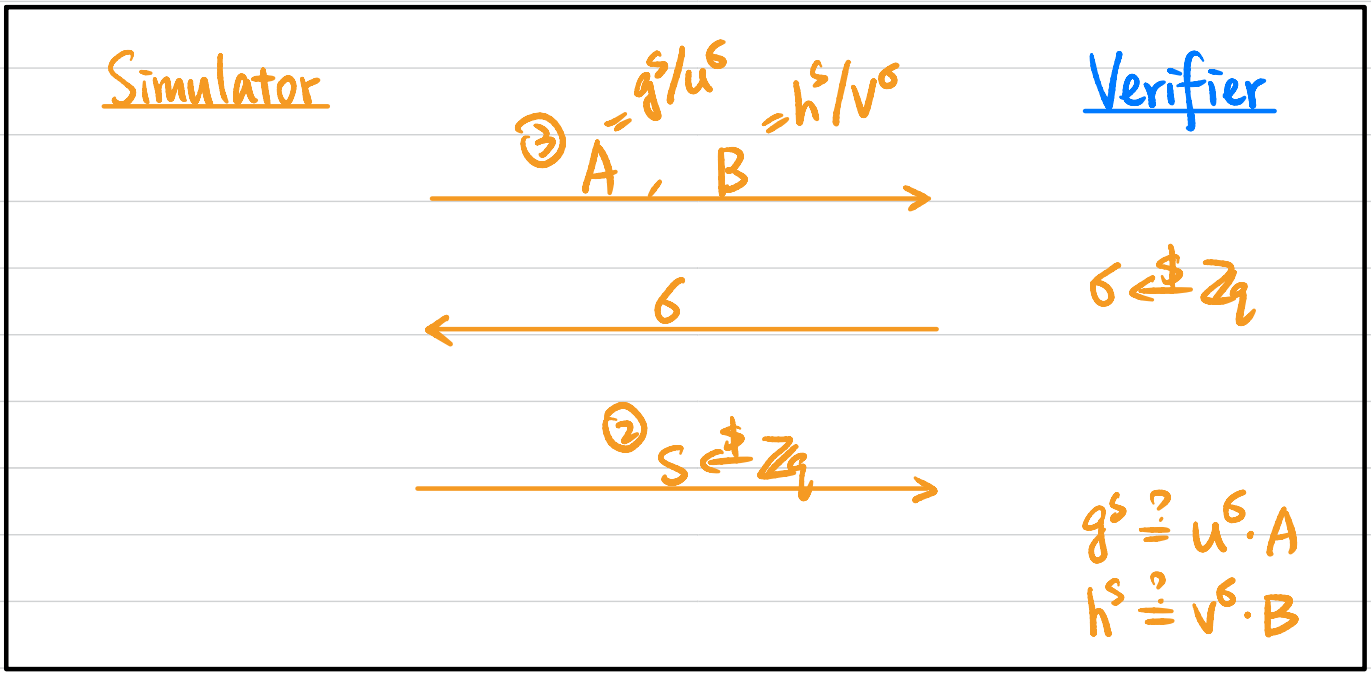
$g^{(s-s')(\sigma-\sigma')^{-1}} = u$, $h^{(s-s')(\sigma-\sigma')^{-1}} = v$

$\Downarrow$

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

Honest Verifier Zero Knowledge ?

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

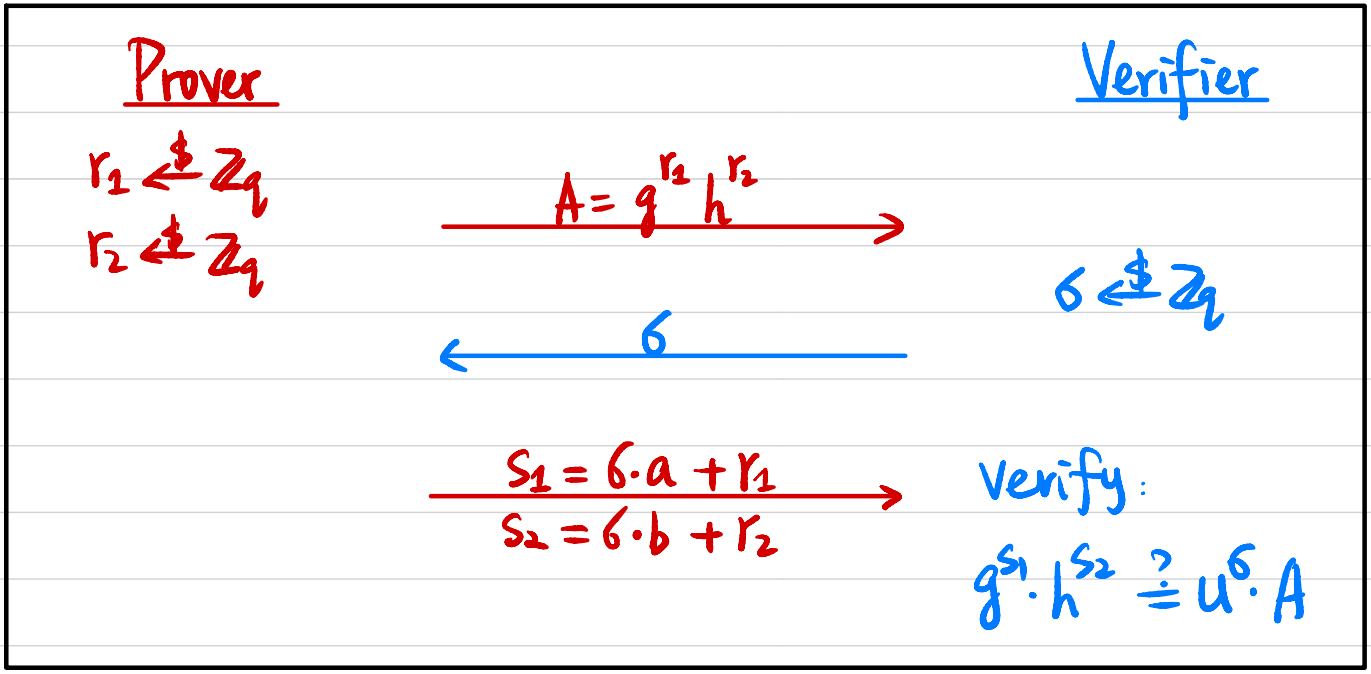


Example 3: Okamoto's Protocol for Representation

Input: Cyclic group G of order q, generator g, h, $u = g^a h^b$

Witness: (a, b)

$R_L = \{ (u = g^a h^b, \ (a, b)) \}$

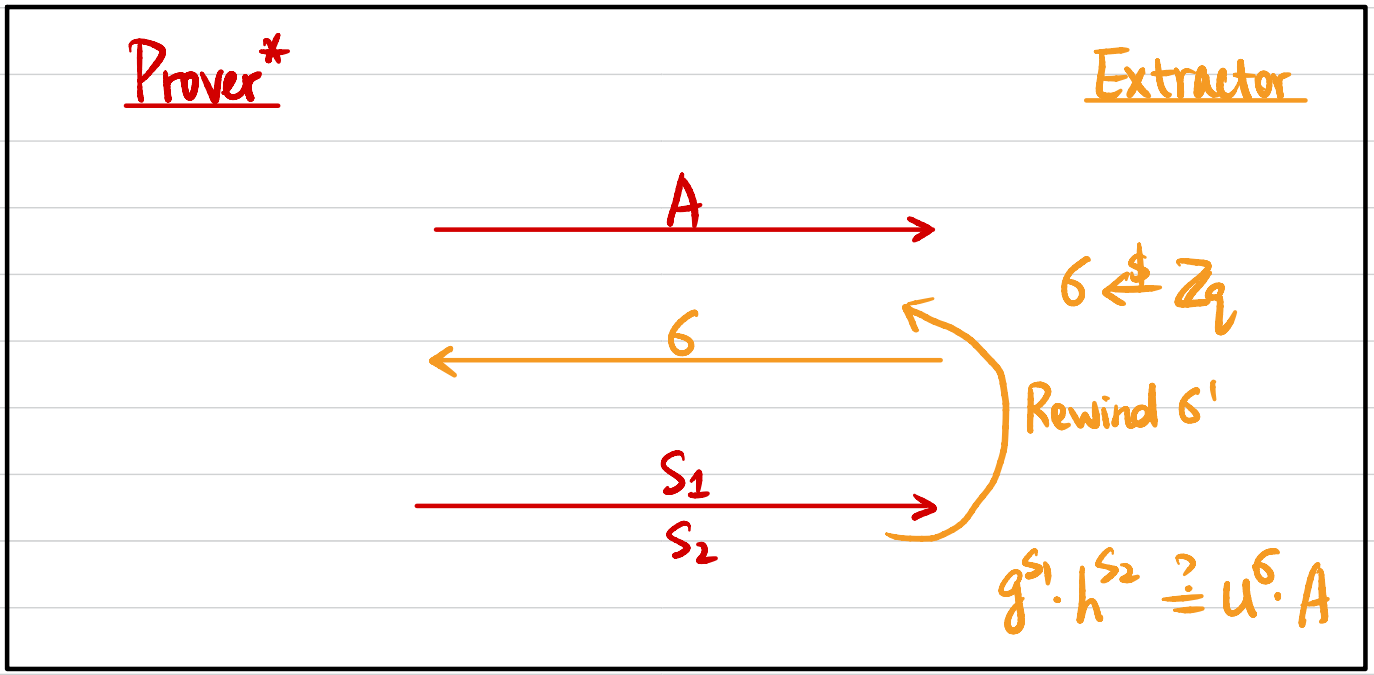


Completeness? $g^{s_1} \cdot h^{s_2} = g^{\sigma \cdot a + r_1} \cdot h^{\sigma \cdot b + r_2}$ $\Rightarrow$ Verifier always outputs 1

$u^{\sigma} \cdot A = (g^a h^b)^{\sigma} \cdot g^{r_1} h^{r_2} = g^{\sigma a + r_1} \cdot h^{\sigma b + r_2}$

Proof of Knowledge ?

Extract (a,b) s.t. $u = g^a h^b$?



$\sigma \Rightarrow s_1, s_2$ s.t. $g^{s_1} \cdot h^{s_2} = u^{\sigma} \cdot A$

$\sigma' \Rightarrow s_1', s_2'$ s.t. $g^{s_1'} \cdot h^{s_2'} = u^{\sigma'} \cdot A$

$\Downarrow$

$$g^{s_1 - s_1'} \cdot h^{s_2 - s_2'} = u^{\sigma - \sigma'}$$

$\Downarrow$

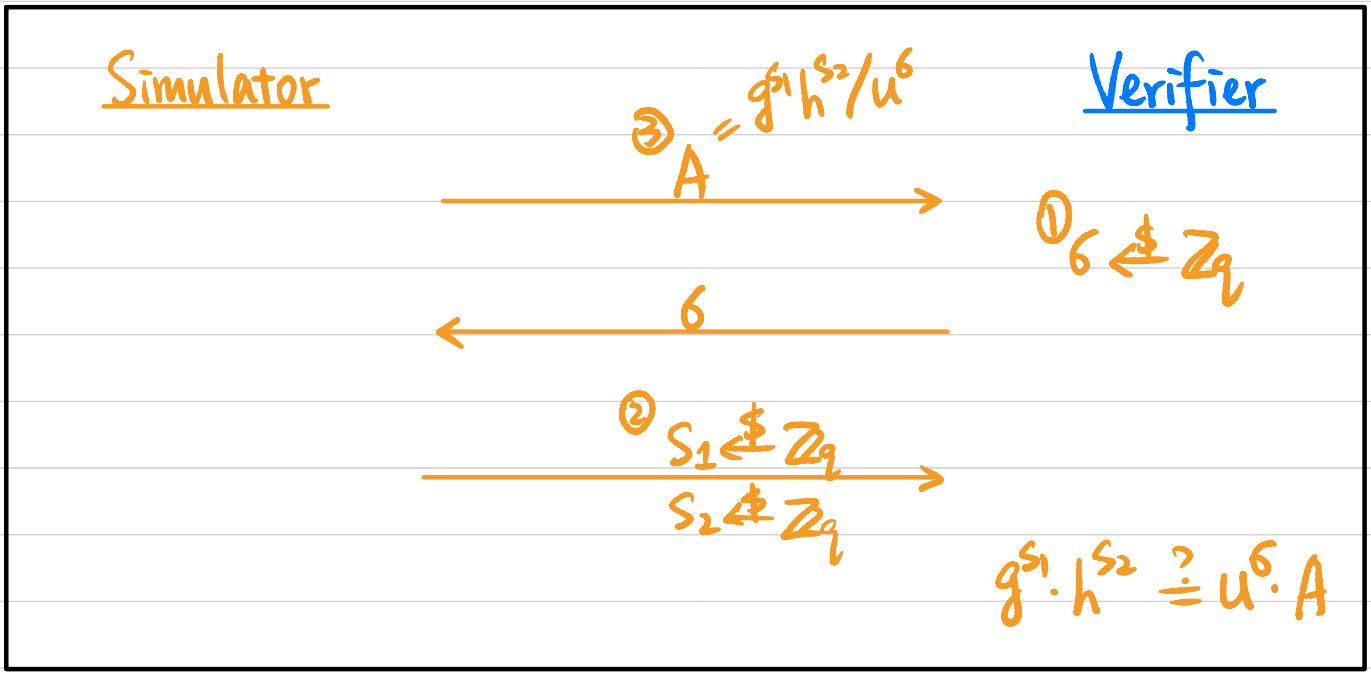
$$g^{(s_1 - s_1')(\sigma - \sigma')^{-1}} \cdot h^{(s_2 - s_2')(\sigma - \sigma')^{-1}} = u$$

$\Downarrow$

$$a = (s_1 - s_1')(\sigma - \sigma')^{-1}, \quad b = (s_2 - s_2')(\sigma - \sigma')^{-1} \pmod q$$

Honest Verifier Zero Knowledge ?

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



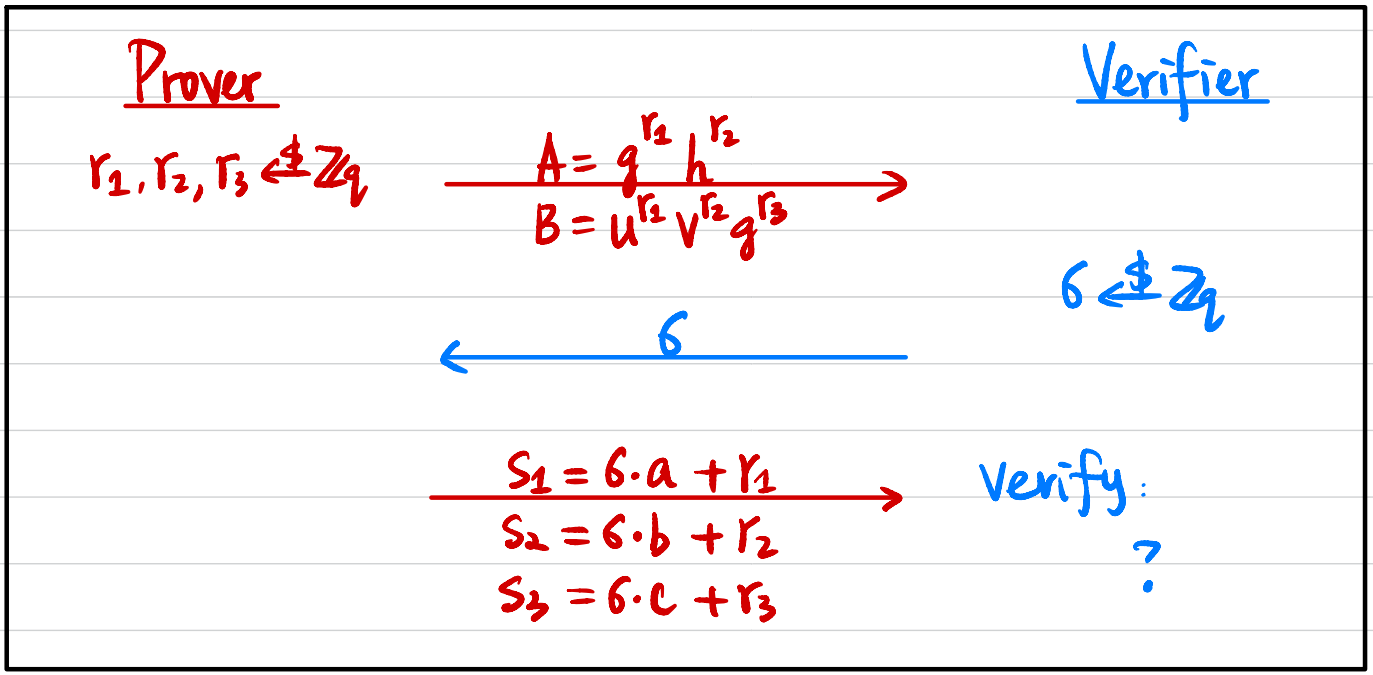
Example 4: Arbitrary Linear Equations

Input: Cyclic group G of order q, generator g, h, u, v

Witness: (a, b, c)

$u = g^a h^b$

$h = u^a v^b g^c$



Completeness ?

PoK ?

HVZK ?

Proving AND/OR Statements ?

Statements: x_1, x_2

Witnesses: w_1, w_2

AND: $R_{AND} = \{ ((x_1, x_2), (w_1, w_2)) :$

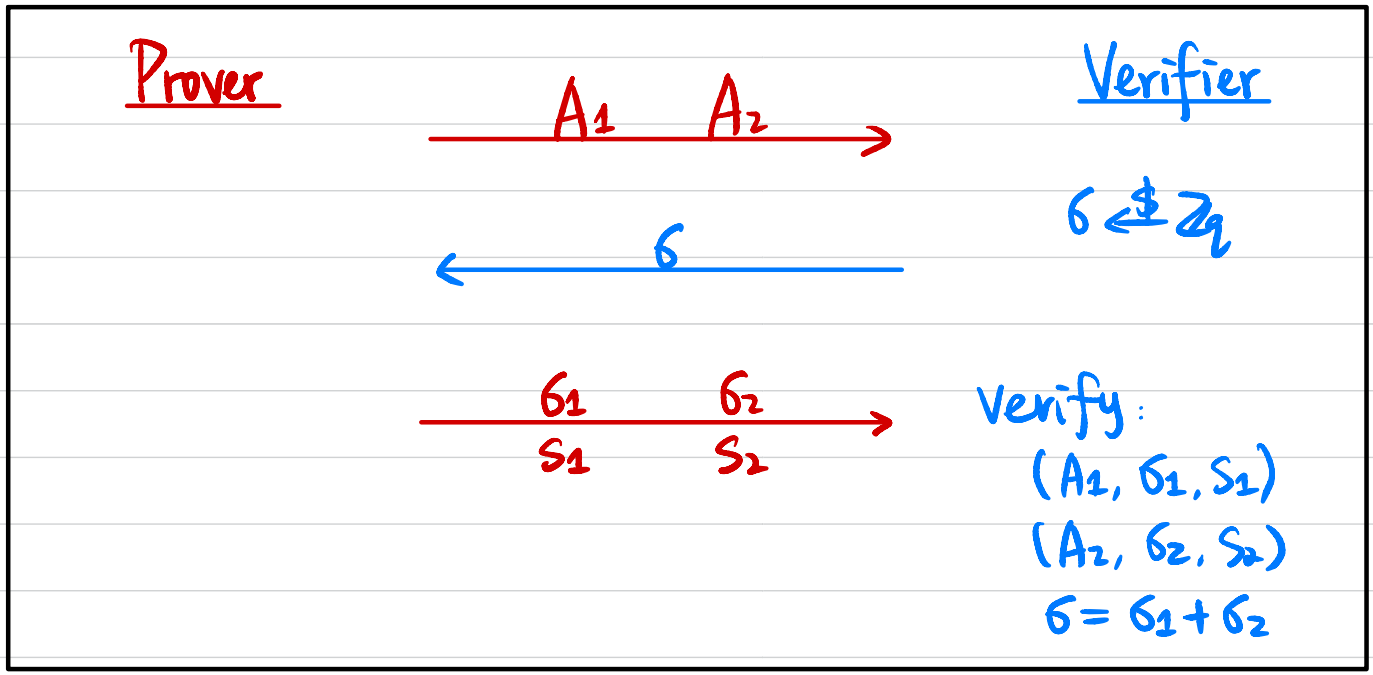
$(x_1, w_1) \in R_{L_1}$ AND $(x_2, w_2) \in R_{L_2} \}$

OR: $R_{OR} = \{ ((x_1, x_2), (w_1, w_2)) :$

$(x_1, w_1) \in R_{L_1}$ OR $(x_2, w_2) \in R_{L_2} \}$

Proving OR Statement

$$R_{OR} = \{ ((x_1, x_2), (w_1, w_2)) : (x_1, w_1) \in R_{L_1} \quad \text{OR} \quad (x_2, w_2) \in R_{L_2} \}$$

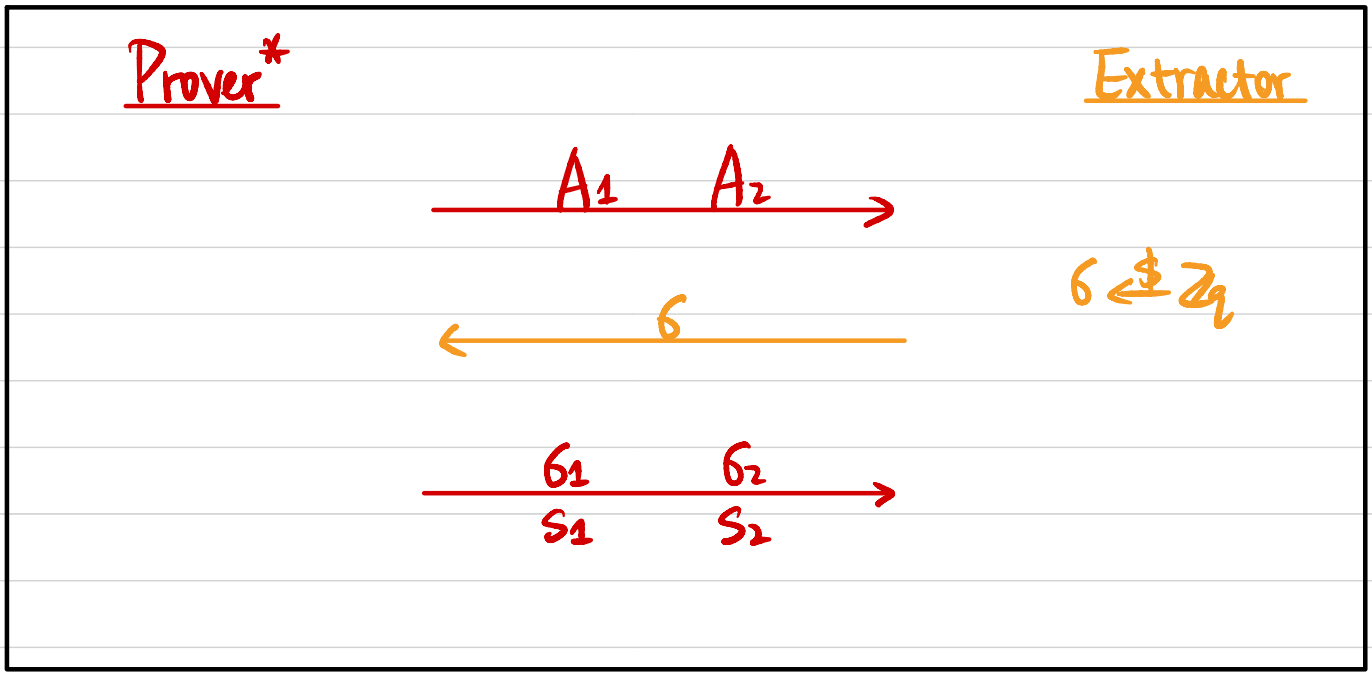


Say Prover only has w_1, how to generate response?

Completeness?

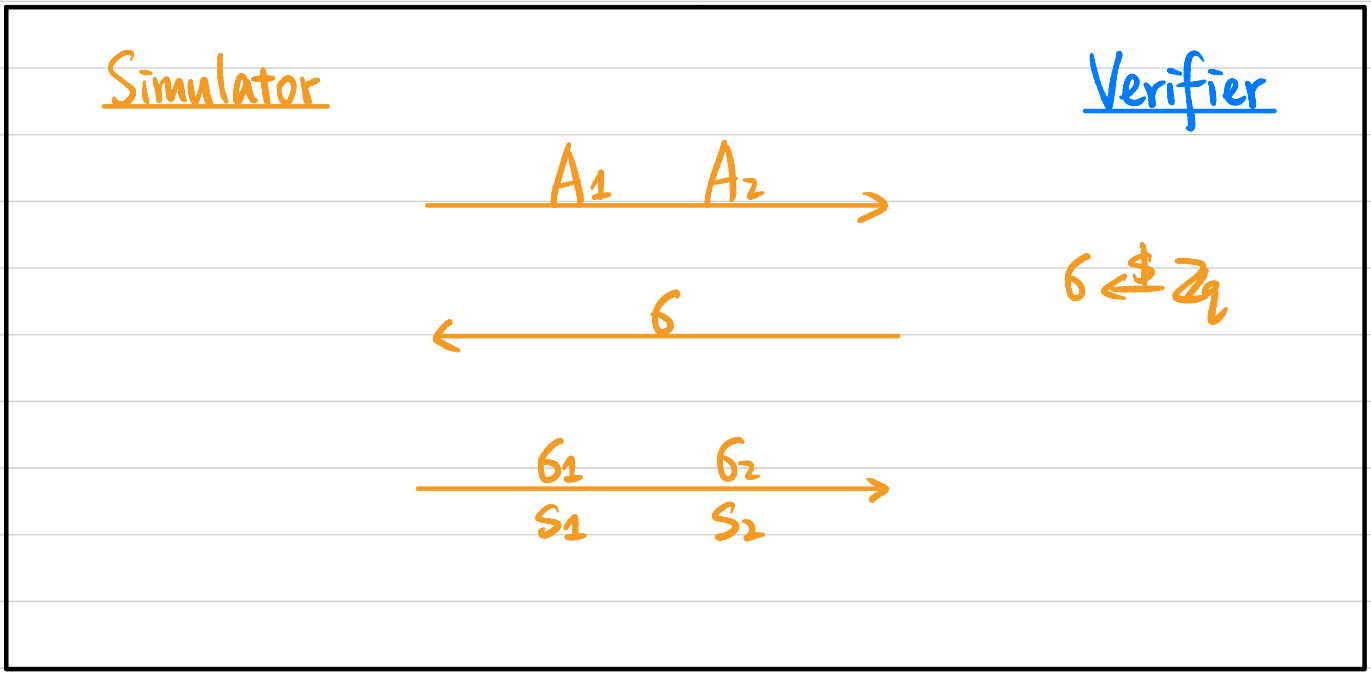
Proof of Knowledge?

Extract (w_1, w_2) s.t. $(x_1, w_1) \in R_{L_1}$ OR $(x_2, w_2) \in R_{L_2}$?

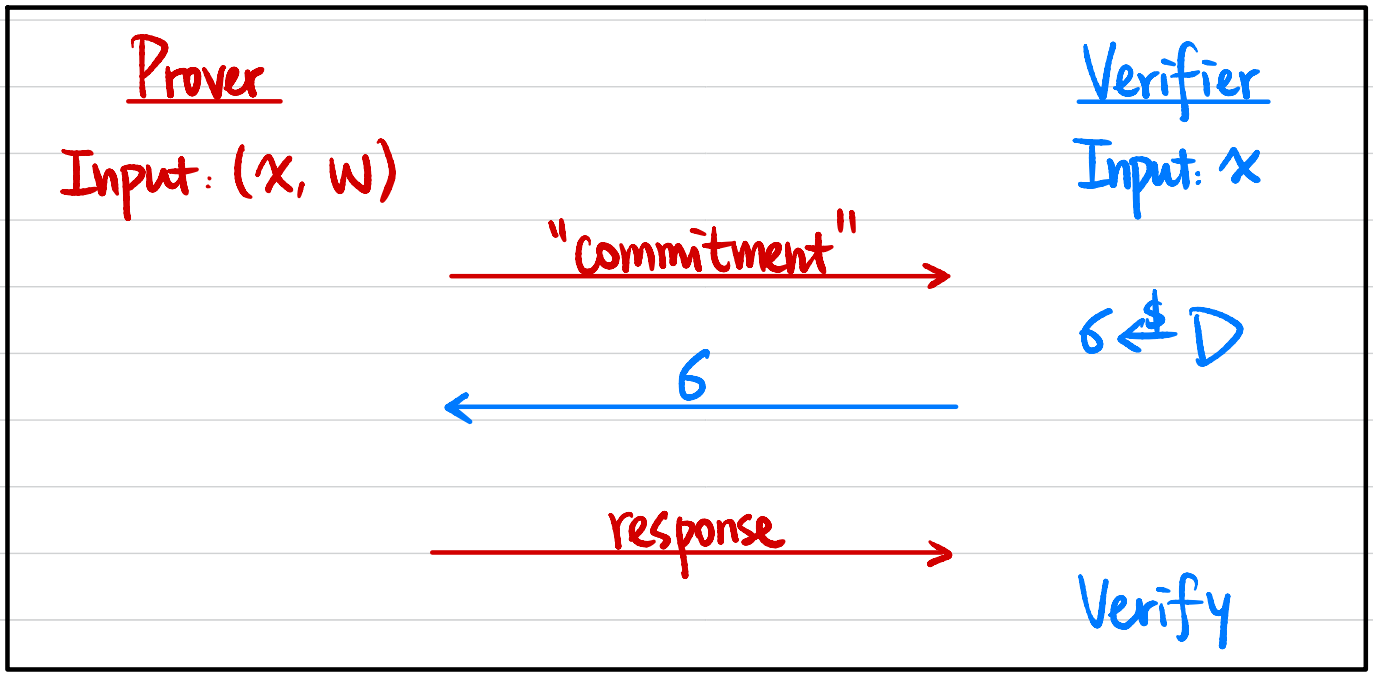


Honest Verifier Zero Knowledge ?

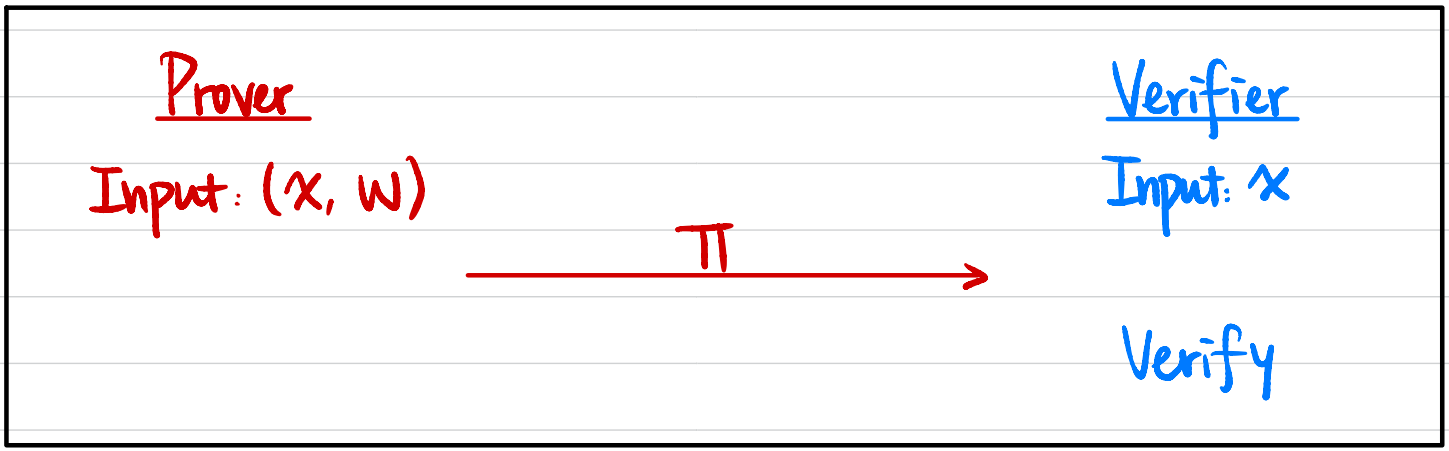
$$\forall (x, w) \in R_L, \quad \text{View}_V[P(x,w) \longleftrightarrow V(x)] \simeq S(x)$$



Sigma Protocols Σ



Non-Interactive Zero-Knowledge (NIZK) Proof

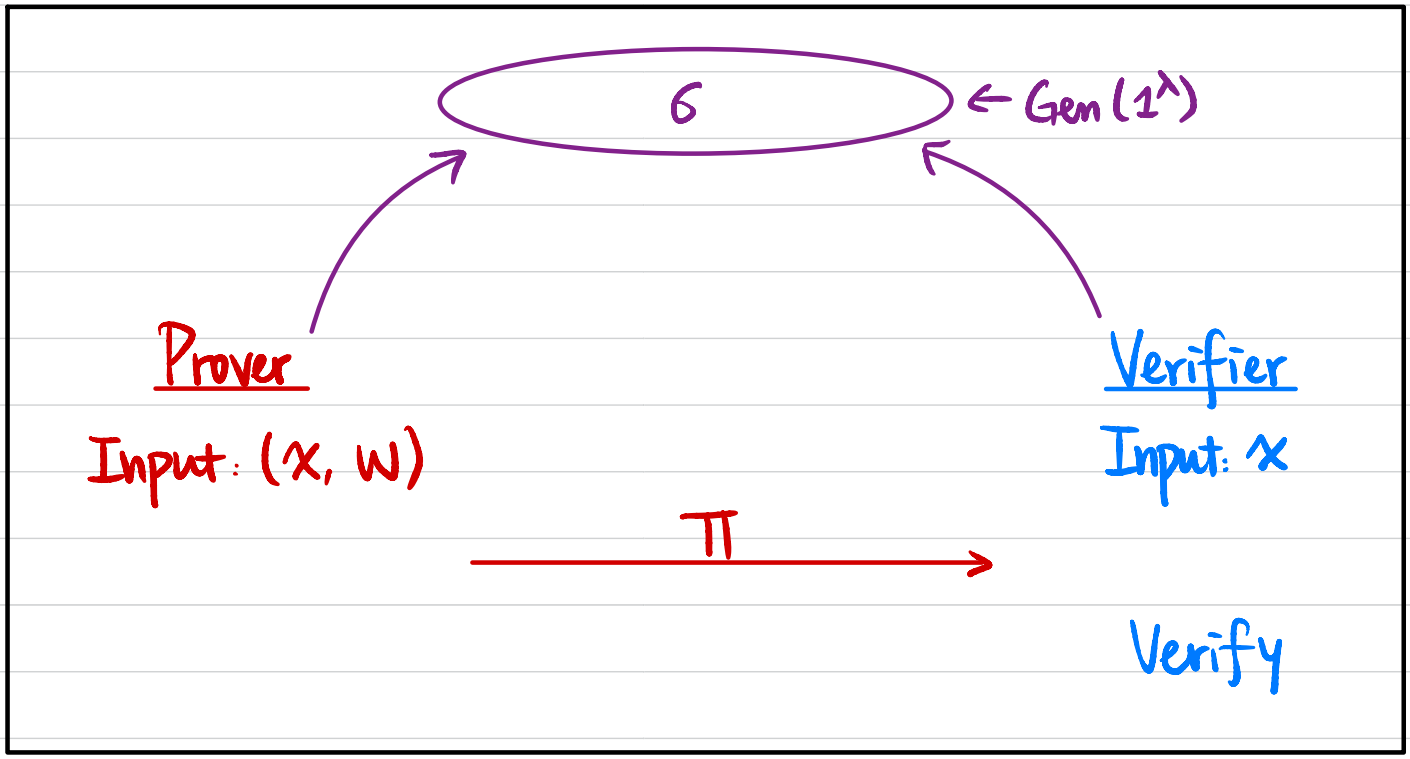


- **Completeness:** $\forall (x,w) \in R_L, \quad \Pr[P(x,w) \longrightarrow V(x) \text{ outputs } 1] = 1.$
- **Soundness:** $\forall x \notin L, \ \forall P^*, \quad \Pr[P^*(x) \longrightarrow V(x) \text{ outputs } 1] \simeq 0.$
- **Zero-Knowledge:** $\forall PPT\ V^*, \ \exists PPT\ S \quad s.t. \quad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

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

Is it possible ?

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



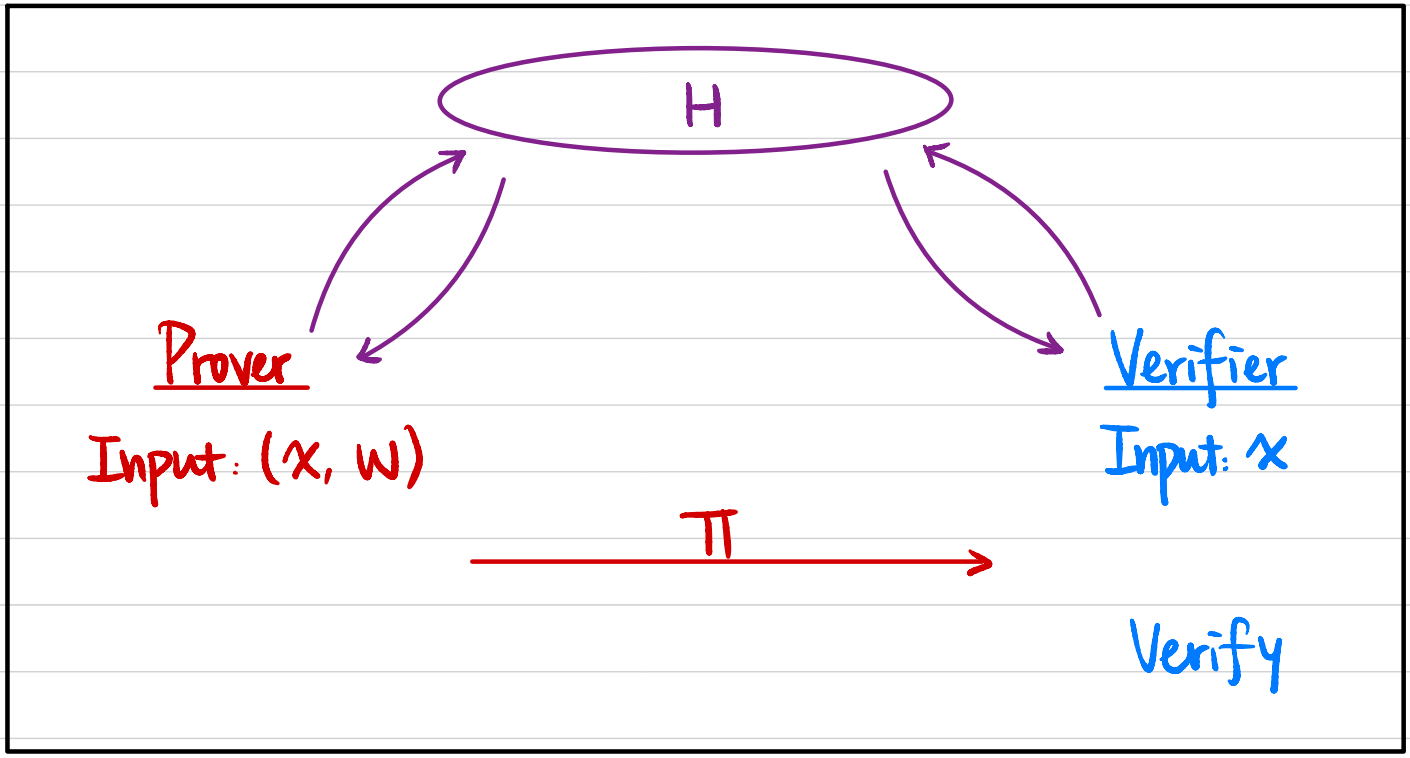
S(x) generates both (σ, π)

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

$$\text{Output}_{V^*}\left[\sigma \leftarrow \text{Gen}(1^\lambda),\ P(x, w, \sigma) \longrightarrow V^*(x, \sigma)\right] \simeq S(x)$$

Alternatively: $$\left(\sigma \leftarrow \text{Gen}(1^\lambda),\ P(x, w, \sigma)\right) \simeq S(x)$$

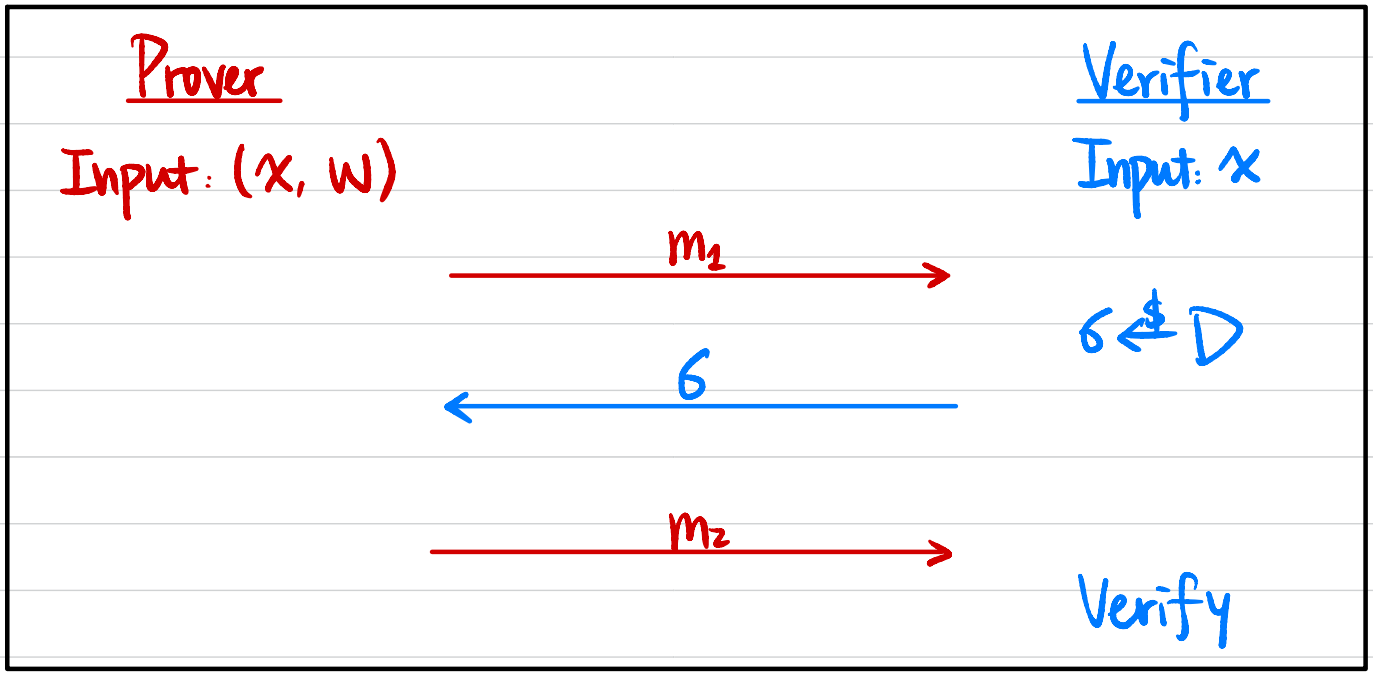
Model 2: Random Oracle Model



S controls input/output behavior of RO

Fiat-Shamir Heuristic

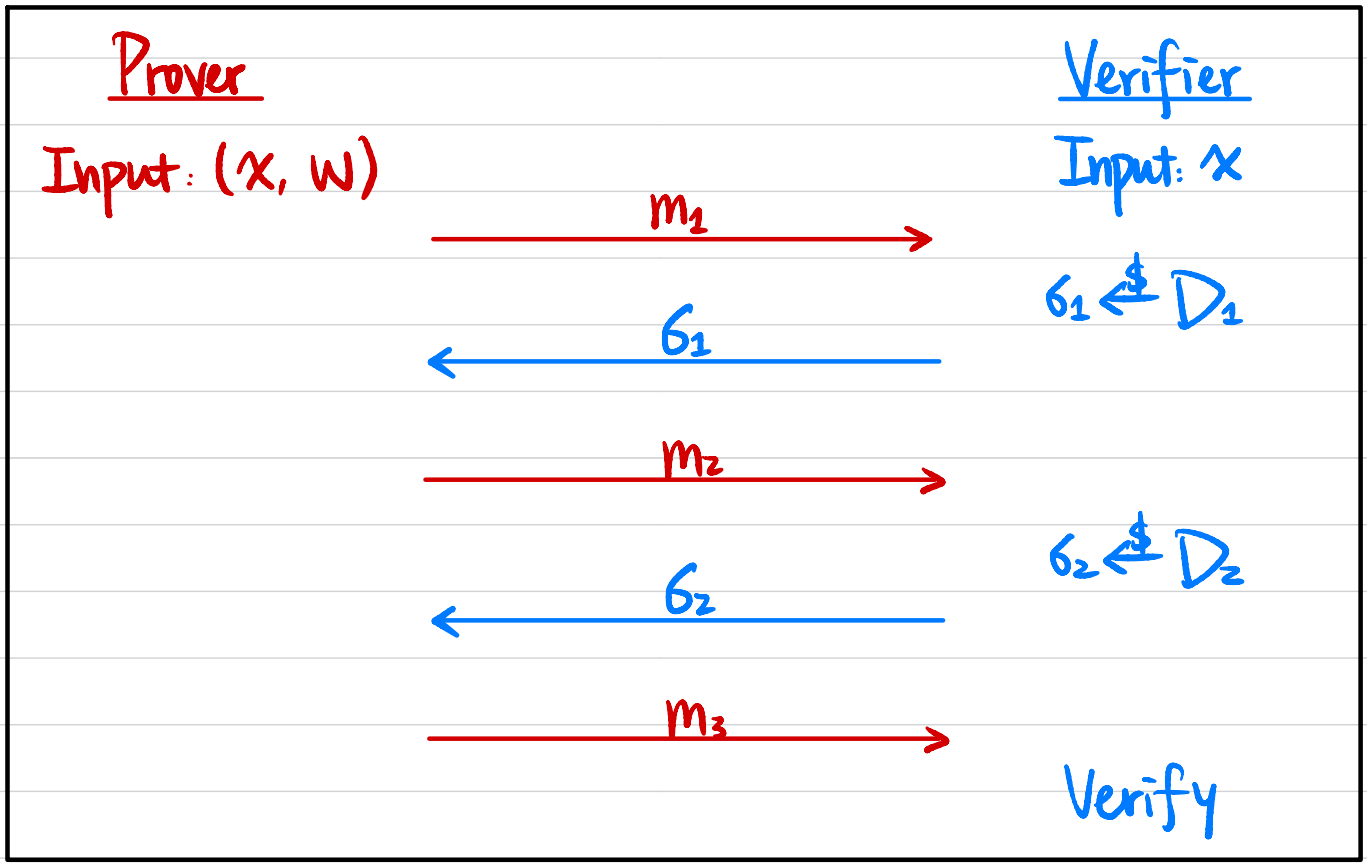
Sigma Protocol $\Rightarrow$ NIZK in the RO model



$\sigma := H(x \| m_1)$

Fiat-Shamir Heuritic

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



$\sigma_1 := H(x \| m_1)$

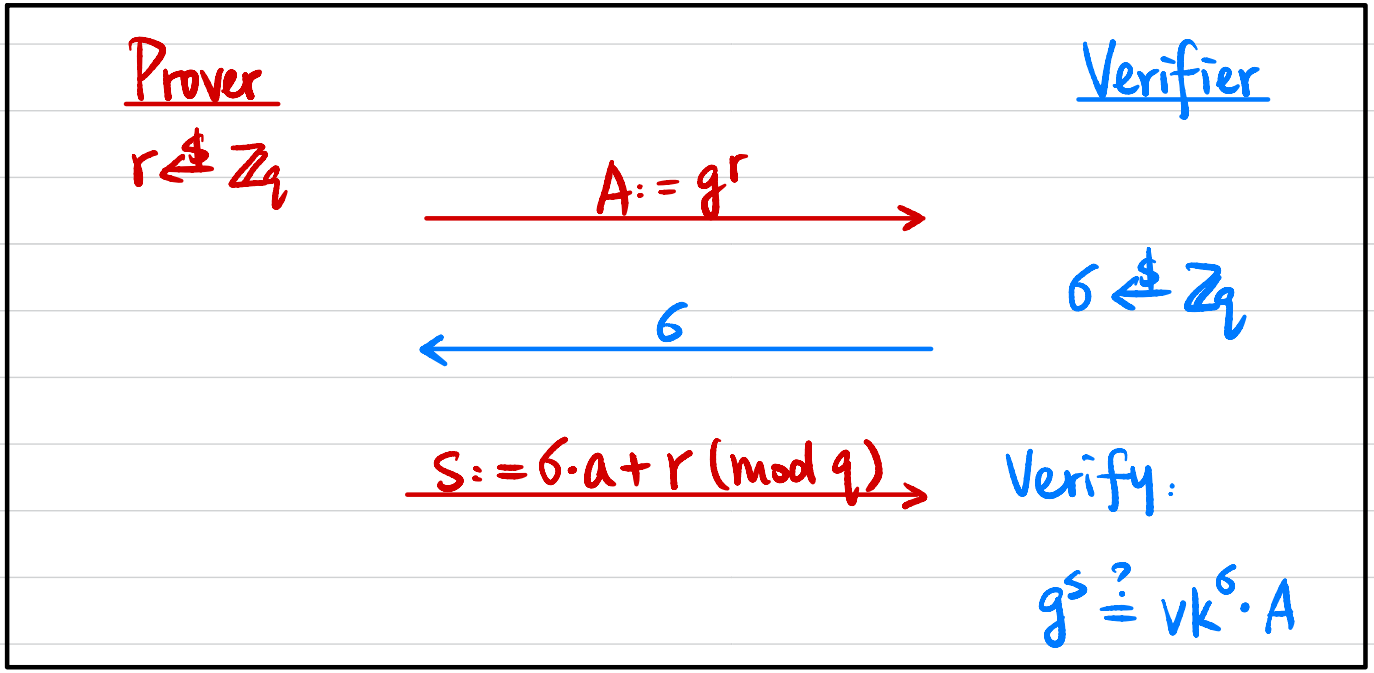
$\sigma_2 := H(x \| m_1 \| m_2)$

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:

Anonymous Online Voting

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

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

⋮

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

$\Downarrow$

$Enc(\Sigma v_i)$

$\Downarrow$

Decrypt to Σv_i

Additively Homomorphic Encryption

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

Additively Homomorphic

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

Multiplicatively Homomorphic

ElGamal Encryption: Cyclic group G with generator g, public key pk.

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

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

Exponential ElGamal:

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

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

Threshold Encryption

P_1: $(pk_1, sk_1) \leftarrow \text{PartialGen}(1^\lambda) \longrightarrow pk_1$

P_2: $(pk_2, sk_2) \leftarrow \text{PartialGen}(1^\lambda) \longrightarrow pk_2$

$\vdots$

P_t: $(pk_t, sk_t) \leftarrow \text{PartialGen}(1^\lambda) \longrightarrow pk_t$

$\Rightarrow pk$

$ct \leftarrow \text{Enc}_{pk}(m)$

P_1: $\alpha_1 \leftarrow \text{PartialDec}(sk_1, ct) \longrightarrow \alpha_1$

P_2: $\alpha_2 \leftarrow \text{PartialDec}(sk_2, ct) \longrightarrow \alpha_2$

$\vdots$

P_t: $\alpha_t \leftarrow \text{PartialDec}(sk_t, ct) \longrightarrow \alpha_t$

$\Rightarrow m$

Threshold Encryption : ElGamal

P_1 : $sk_1 \xleftarrow{\$} \mathbb{Z}_q$ $\quad pk_1 = g^{sk_1}$ $\longrightarrow$ pk_1

P_2 : $sk_2 \xleftarrow{\$} \mathbb{Z}_q$ $\quad pk_2 = g^{sk_2}$ $\longrightarrow$ pk_2

$\vdots$

P_t : $sk_t \xleftarrow{\$} \mathbb{Z}_q$ $\quad pk_t = g^{sk_t}$ $\longrightarrow$ pk_t

$\Rightarrow pk = \prod pk_i$

$sk = ?$

$ct \leftarrow Enc_{pk}(m)$

$ct = (c_1, c_2)$

$= (g^r, pk^r \cdot g^m)$

P_1 : $\alpha_1 = c_1^{sk_1}$ $\longrightarrow$ α_1

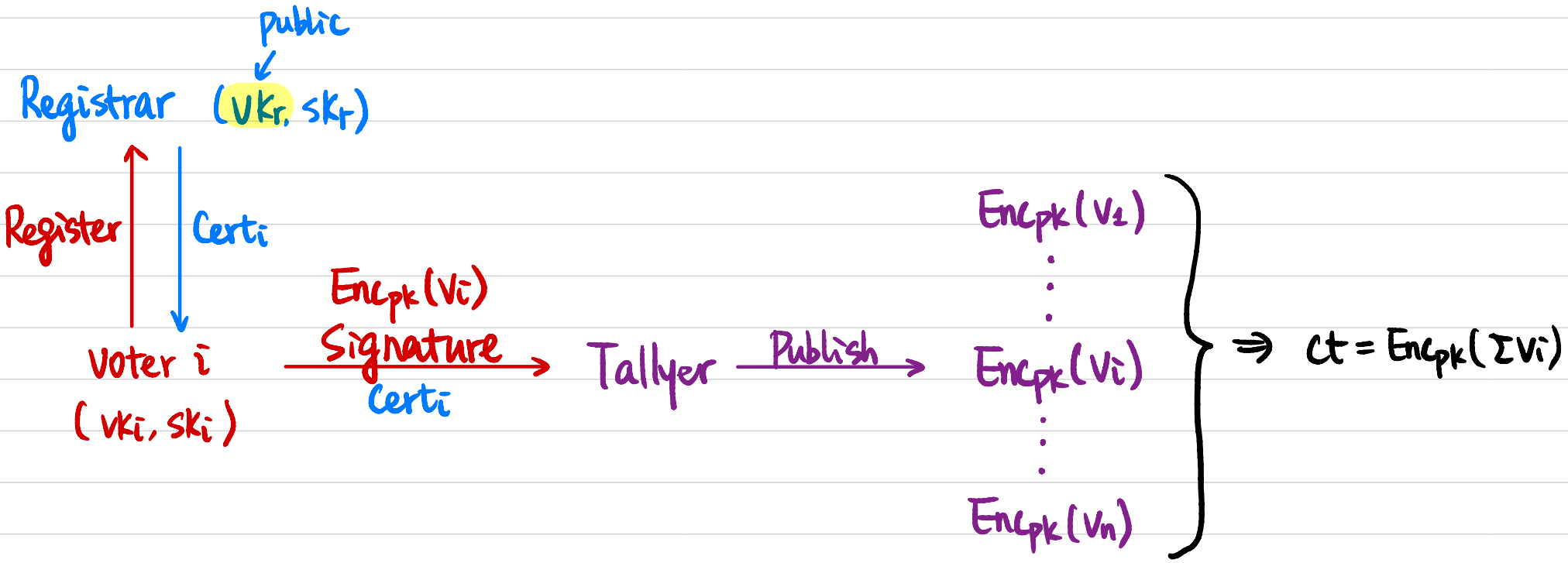
P_2 : $\alpha_2 = c_1^{sk_2}$ $\longrightarrow$ α_2

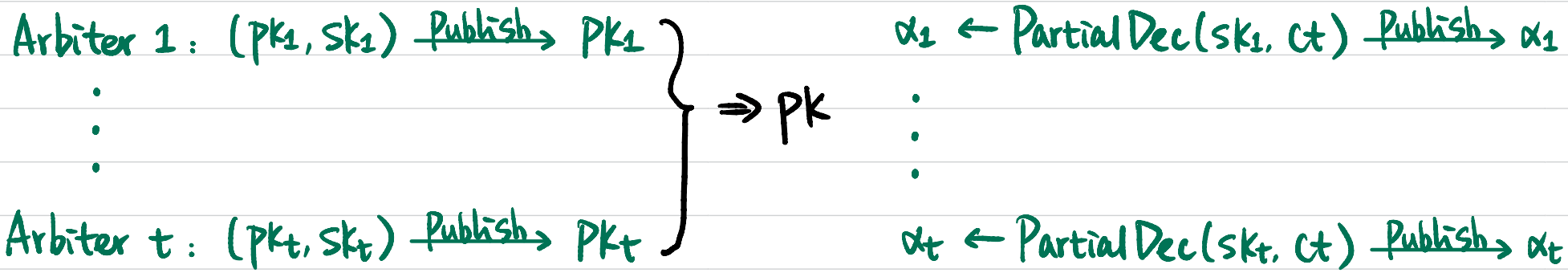
$\vdots$

P_t : $\alpha_t = c_1^{sk_t}$ $\longrightarrow$ α_t

$\Rightarrow m = ?$

Anonymous Online Voting





Correctness of Encryption

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

Public key $pk \in G$.

Ciphertext $C = (c_1, c_2)$

ZKP for an OR statement:

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

Witness: randomness r used in encryption

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

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

Correctness of Partial Decryption

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

Partial public key $pk_i \in G$.

Ciphertext $C = (C_1, C_2)$.

Partial decryption α_i

Witness: partial secret key sk_i

ZKP for partial decryption:

$$R_L = \{ ((pk_i, C_1, \alpha_i), sk_i) : pk_i = g^{sk_i} \wedge \alpha_i = C_1^{sk_i} \}$$

Multiple Candidates ?

k candidates

Voter 1 $\longrightarrow$ Enc (V_1) $\quad V_1 \in \{0, 1, \cdots, k-1\}$

Voter 2 $\longrightarrow$ Enc (V_2) $\quad V_2 \in \{0, 1, \cdots, k-1\}$

$\vdots$

Voter n $\longrightarrow$ Enc (V_n) $\quad V_n \in \{0, 1, \cdots, k-1\}$

$\Downarrow$

Enc (ΣV_i)

$\Downarrow$

Decrypt to ΣV_i