

CSCI 1515 Applied Cryptography

This Lecture:

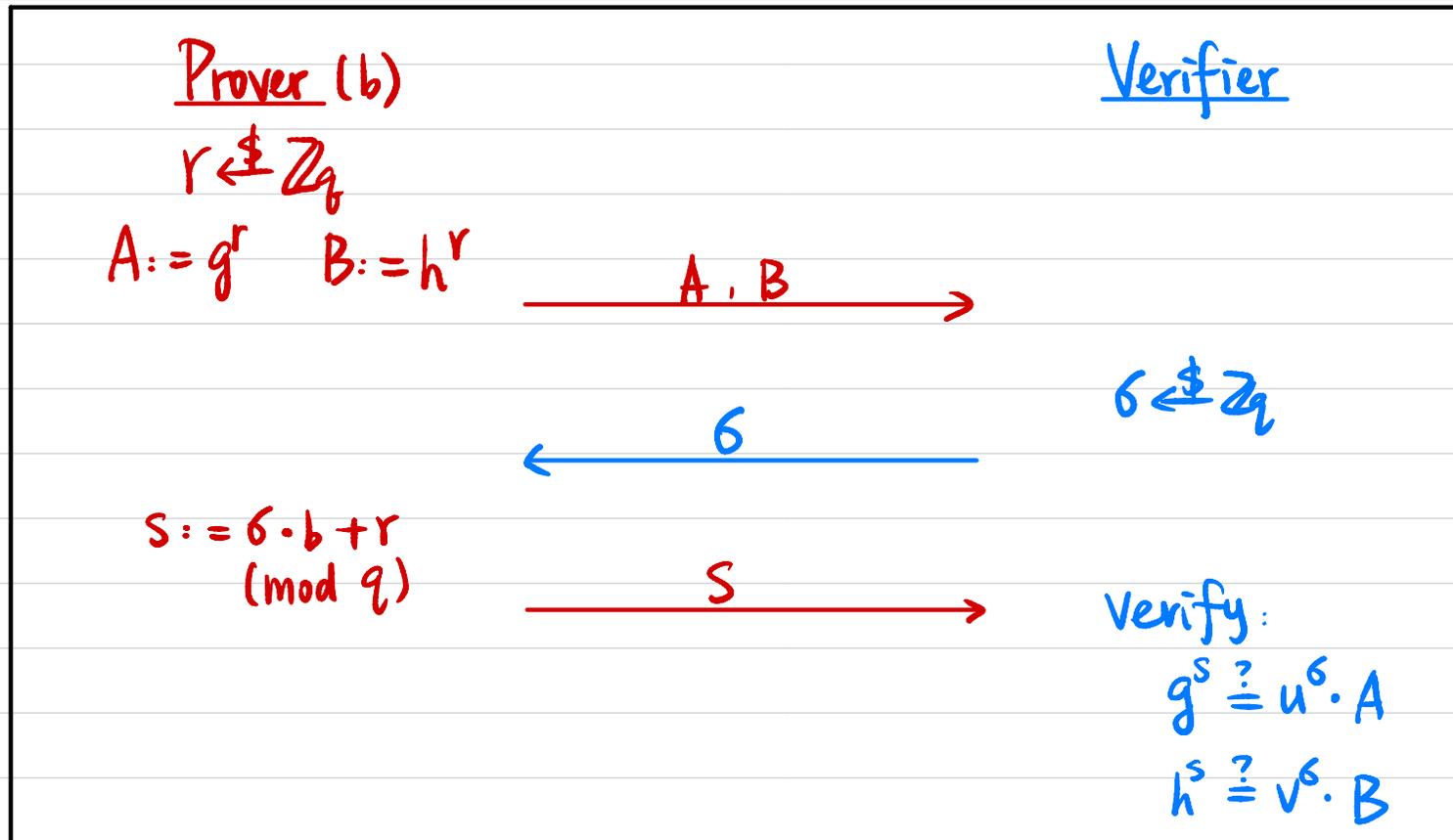
- Example: Diffie-Hellman Tuple
- Non-Interactive Zero-Knowledge (NIZK) Proof
- Fiat-Shamir Heuristic
- Anonymous Online Voting: More Details
- Homomorphism of ElGamal Encryption
- ZKP for OR Statements

Example: Diffie-Hellman Tuple

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

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

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



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

Zero-Knowledge Proof of Knowledge

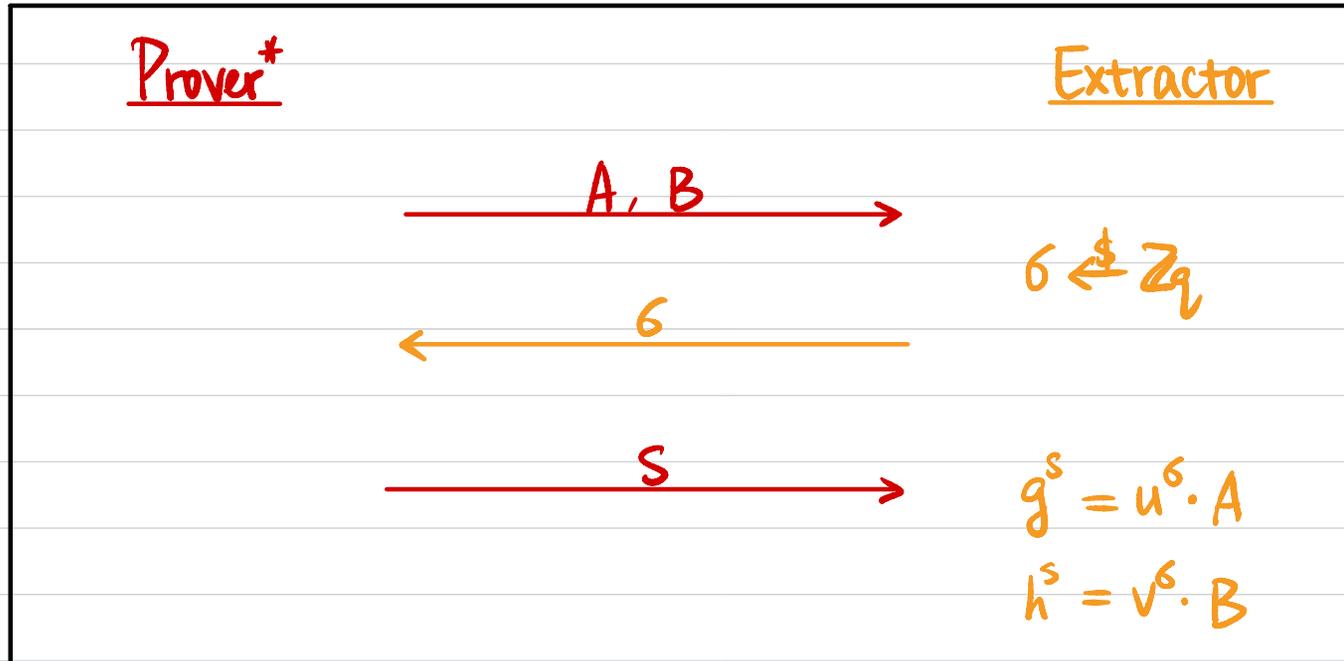
- Completeness: $\forall (x, w) \in R_L, \Pr [P(x, w) \leftrightarrow V(x) \text{ outputs } 1] = 1.$
 $\forall (x, w) \in R_L, P \text{ can prove it.}$
- Soundness: $\forall x \notin L, \forall P^*, \Pr [P^*(x) \leftrightarrow V(x) \text{ outputs } 1] \approx 0.$
If P^* can prove it, $x \in L.$
- Proof 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^* can prove it, P^* must know $w.$
- Honest-Verifier Zero-Knowledge (HVZK): $\exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$
 $\text{View}_V [P(x, w) \leftrightarrow V(x)] \approx S(x)$
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

Proof of Knowledge?

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

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



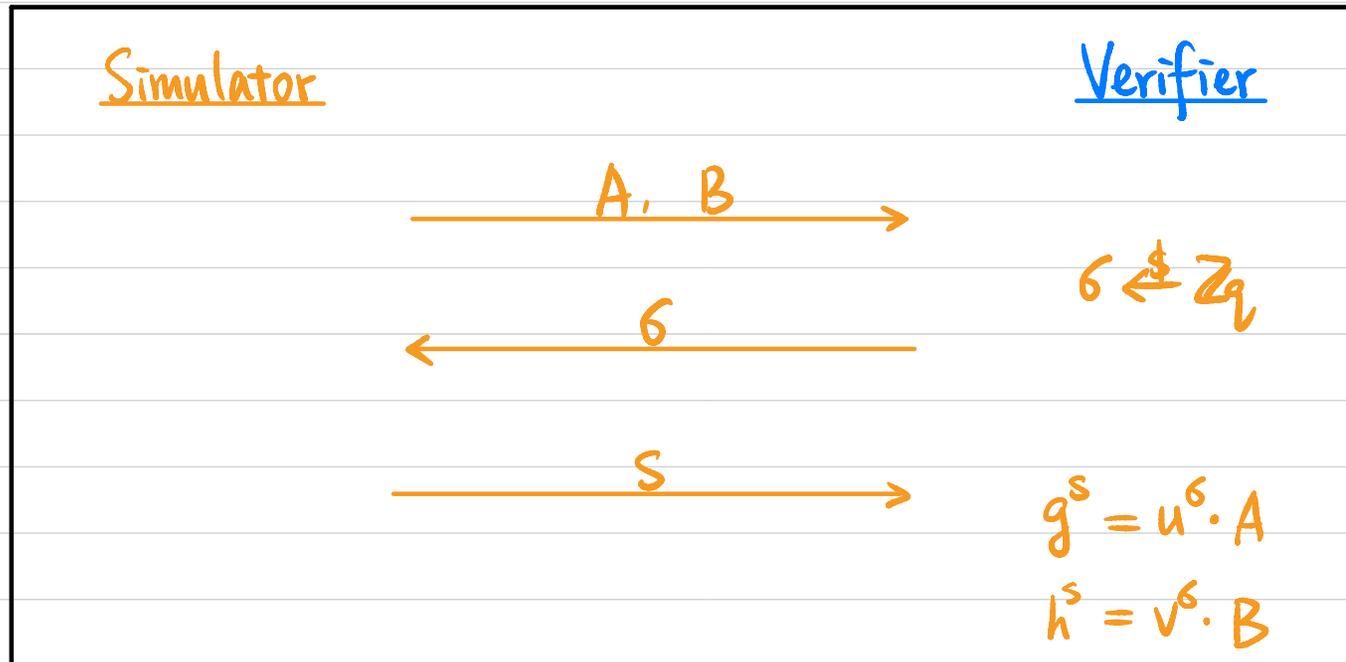
How to extract b s.t. $u = g^b \wedge v = h^b$?

Example: Diffie-Hellman Tuple

Honest-Verifier Zero-Knowledge (HVZK)?

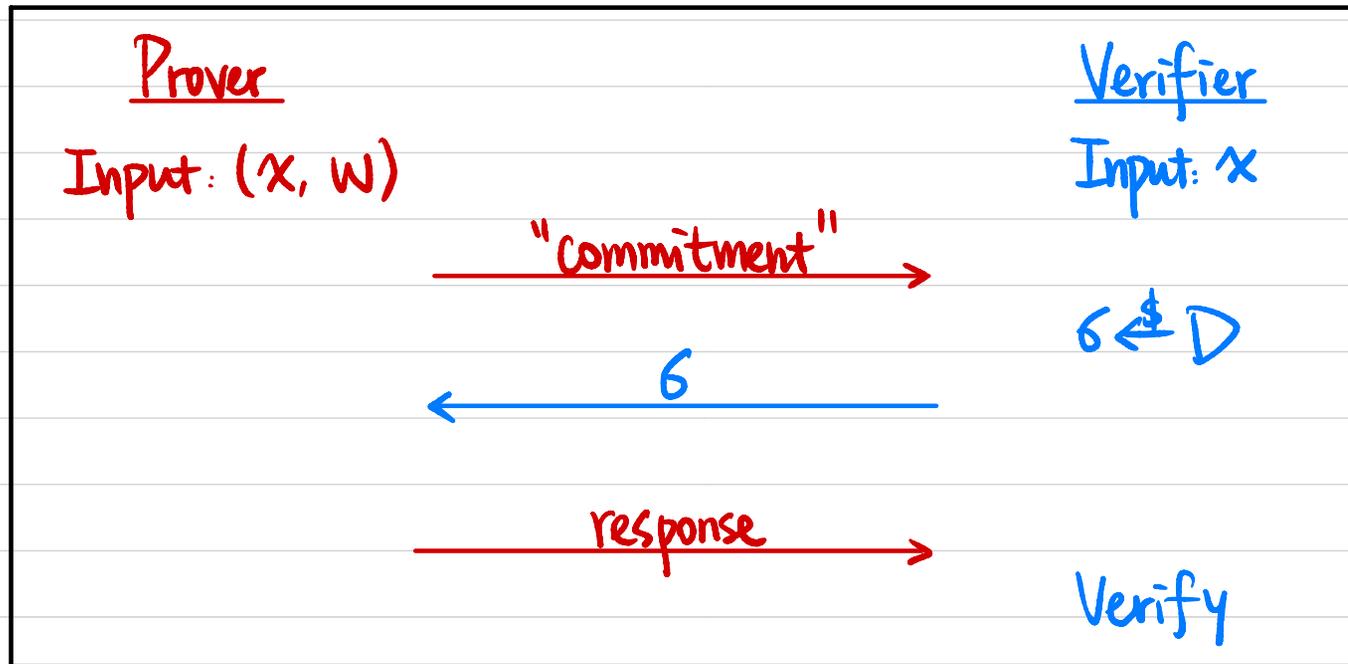
\exists PPT S s.t. $\forall (x, w) \in R,$

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

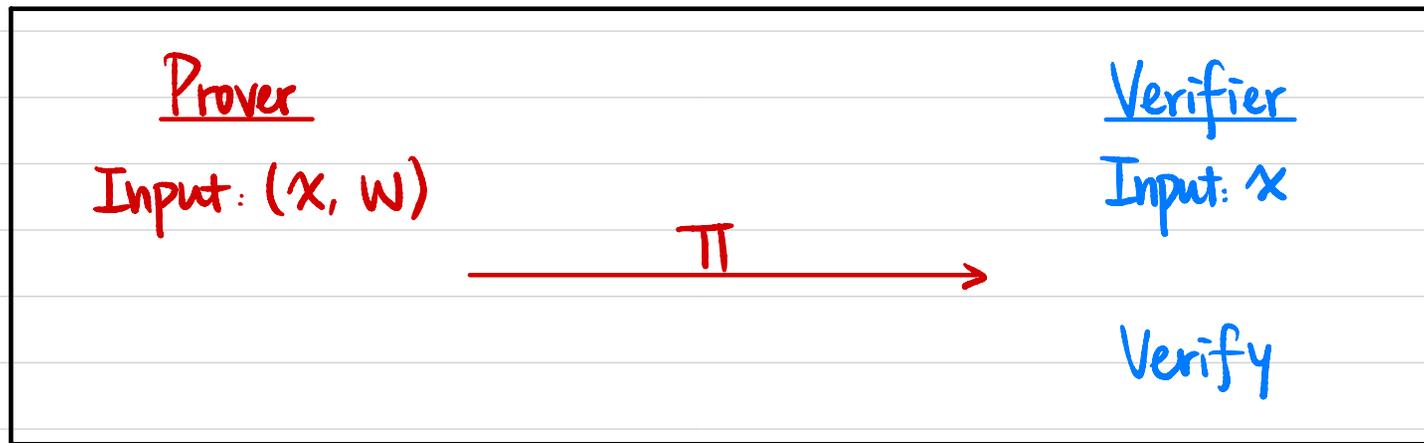


How to generate (A, B, s) s.t. $g^s = h^\sigma \cdot A \wedge h^s = v^\sigma \cdot B$?

Sigma Protocols Σ



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:** $\forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L, \text{Output}_{V^*} [P(x, w) \rightarrow V^*(x)] \approx S(x)$

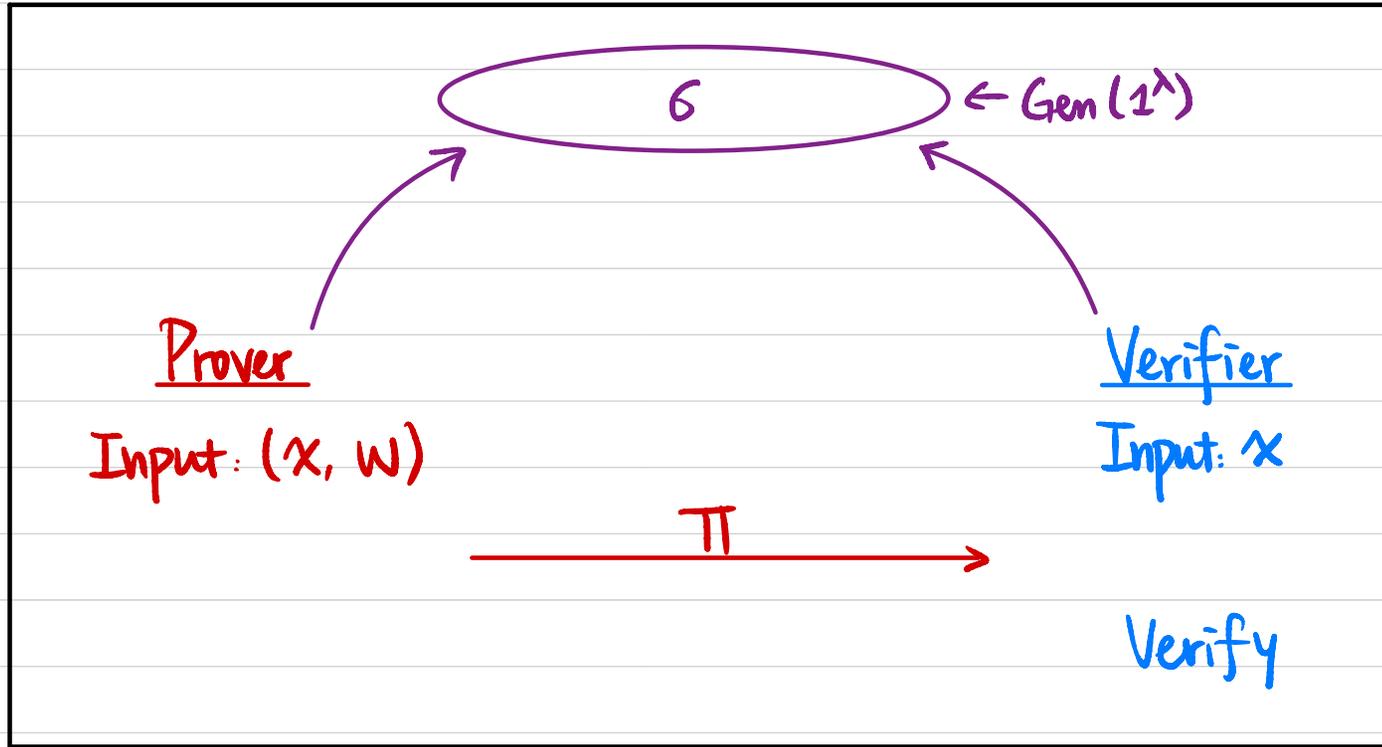
Is it possible?

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

If (h, u, v) is a DH tuple, then $S(h, u, v)$ outputs?

If (h, u, v) is not a DH tuple, then $S(h, u, v)$ outputs?

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



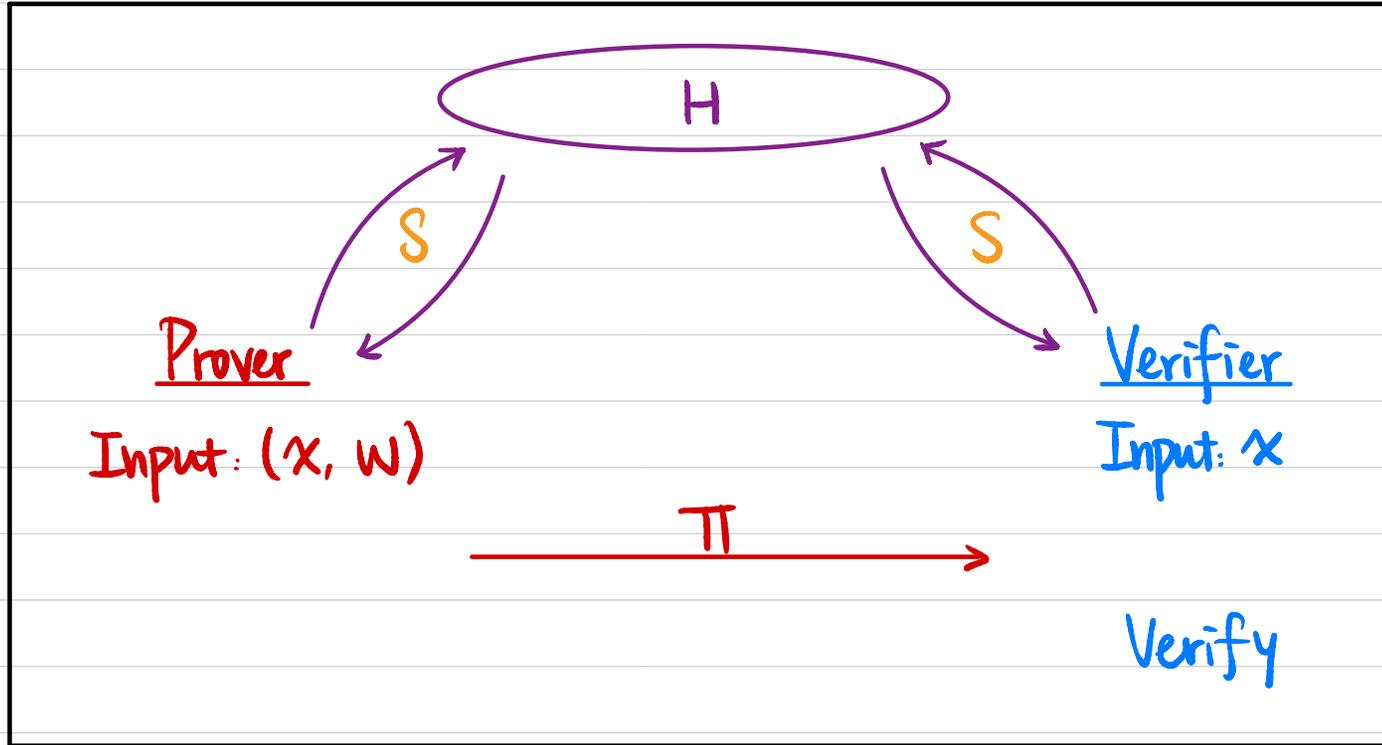
• **Soundness:** $\forall x \notin L, \forall P^*, \Pr[G \leftarrow \text{Gen}(1^\lambda), P^*(G, x) \rightarrow V(G, x) \text{ outputs } 1] \approx 0.$

• **Zero-Knowledge:** $\forall \text{PPT } V^*, \exists \text{PPT } S \text{ s.t. } \forall (x, w) \in R_L,$
 $\text{Output}_{V^*}[G \leftarrow \text{Gen}(1^\lambda), P(x, w, G) \rightarrow V^*(x, G)] \approx S(x)$

Alternatively: $(G \leftarrow \text{Gen}(1^\lambda), P(x, w, G)) \approx S(x)$

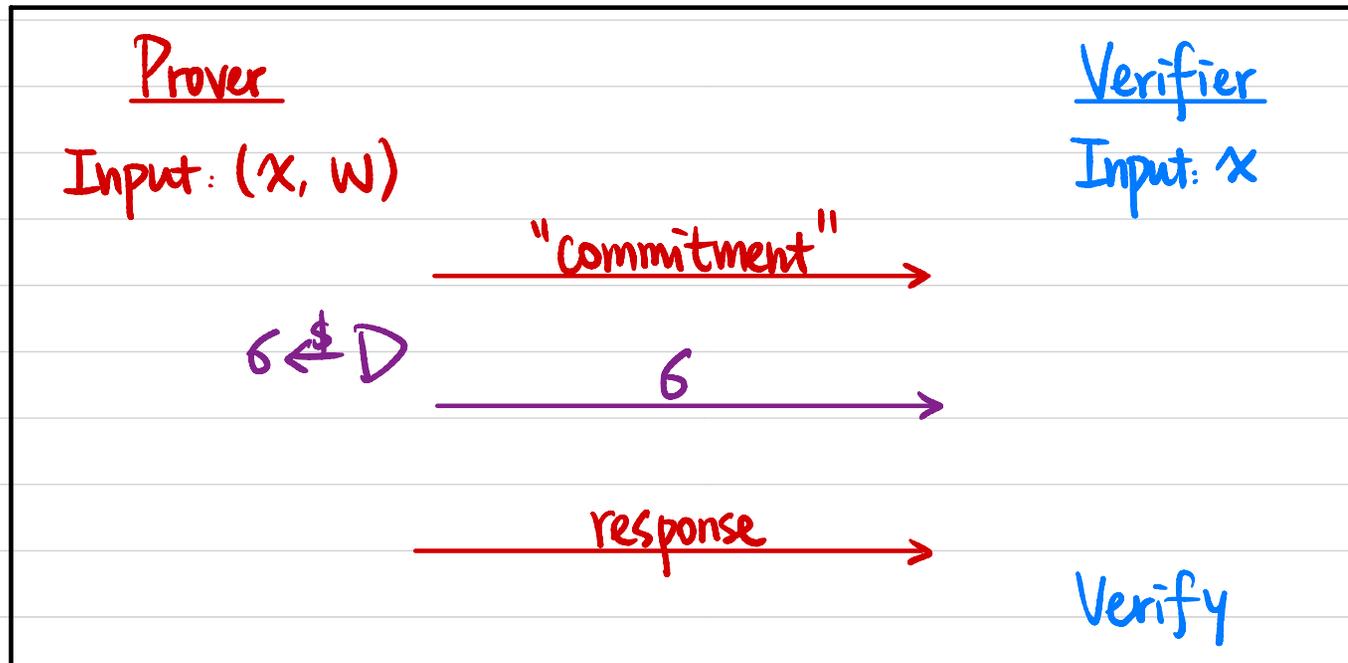
$S(x)$ generates both (G, π)

Model 2: Random Oracle Model



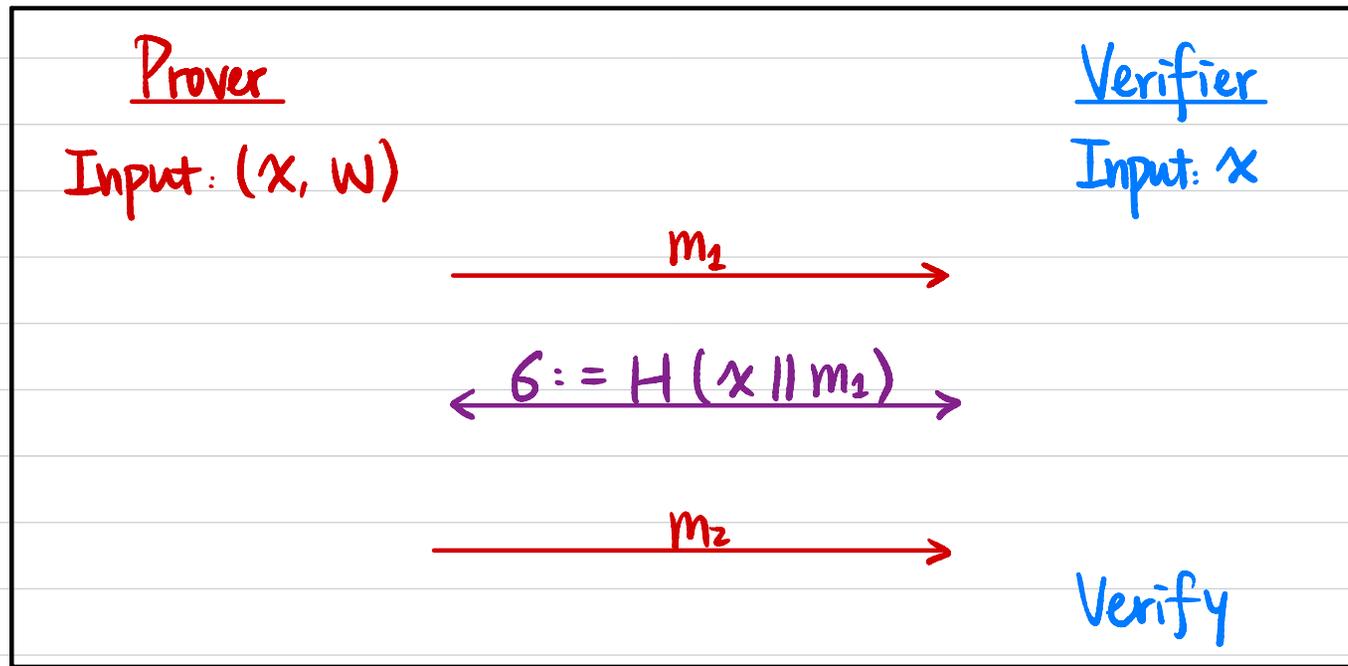
S controls input/output behavior of RO

Sigma Protocol \Rightarrow NIZK?



Fiat-Shamir Heuristic

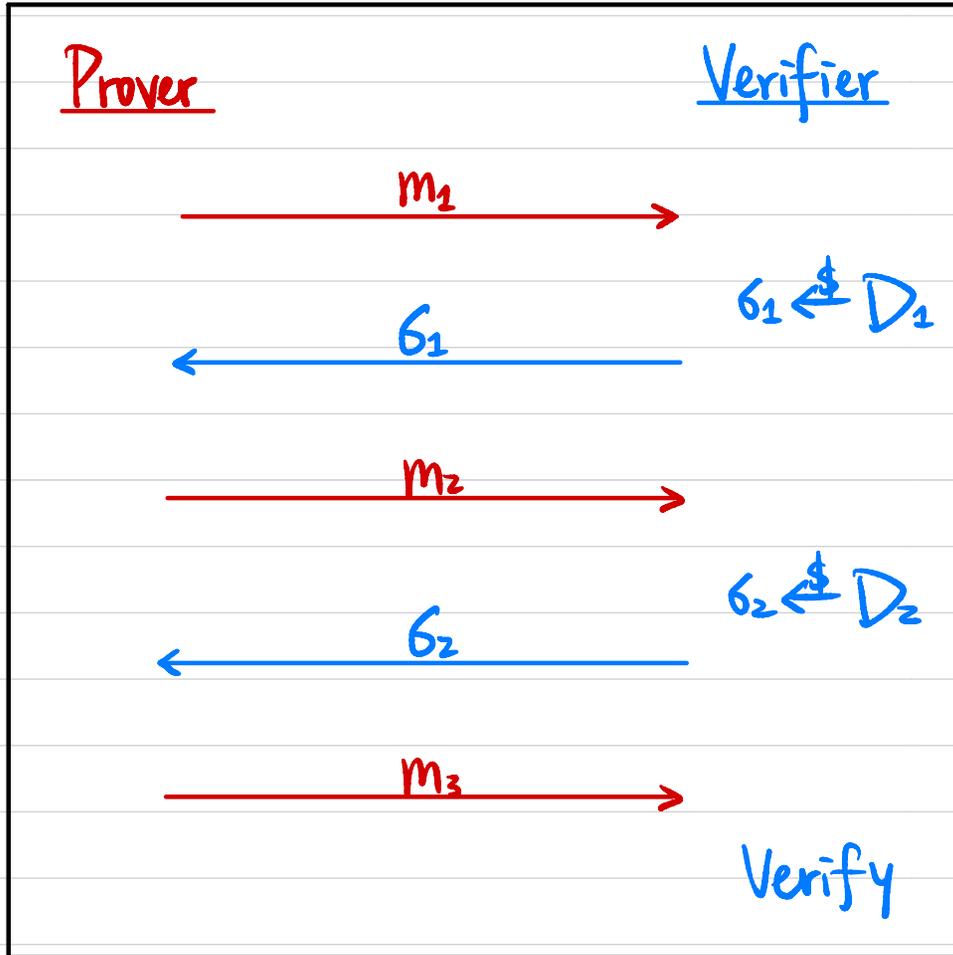
Sigma Protocol \Rightarrow NIZK in the RO model



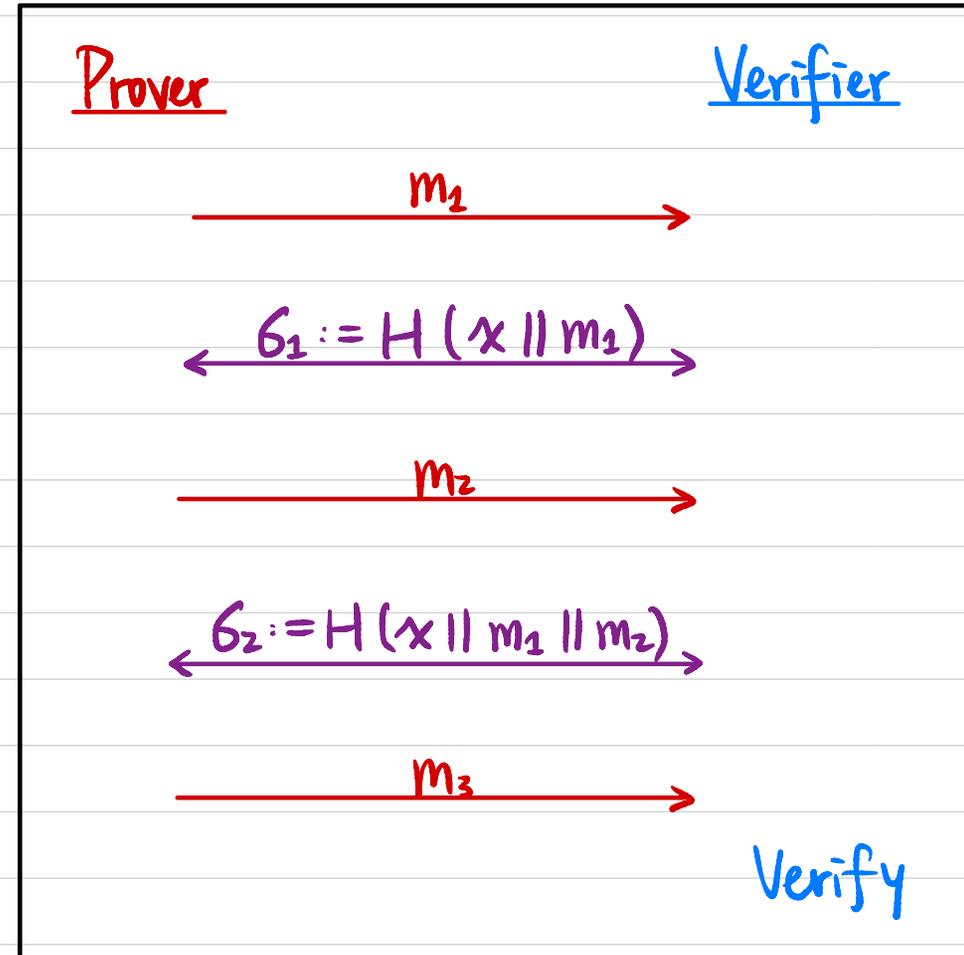
$$\pi = (m_1, m_2)$$

Fiat-Shamir Heuristic

Public-Coin HVZK \Rightarrow NIZK in the RO model



\Rightarrow

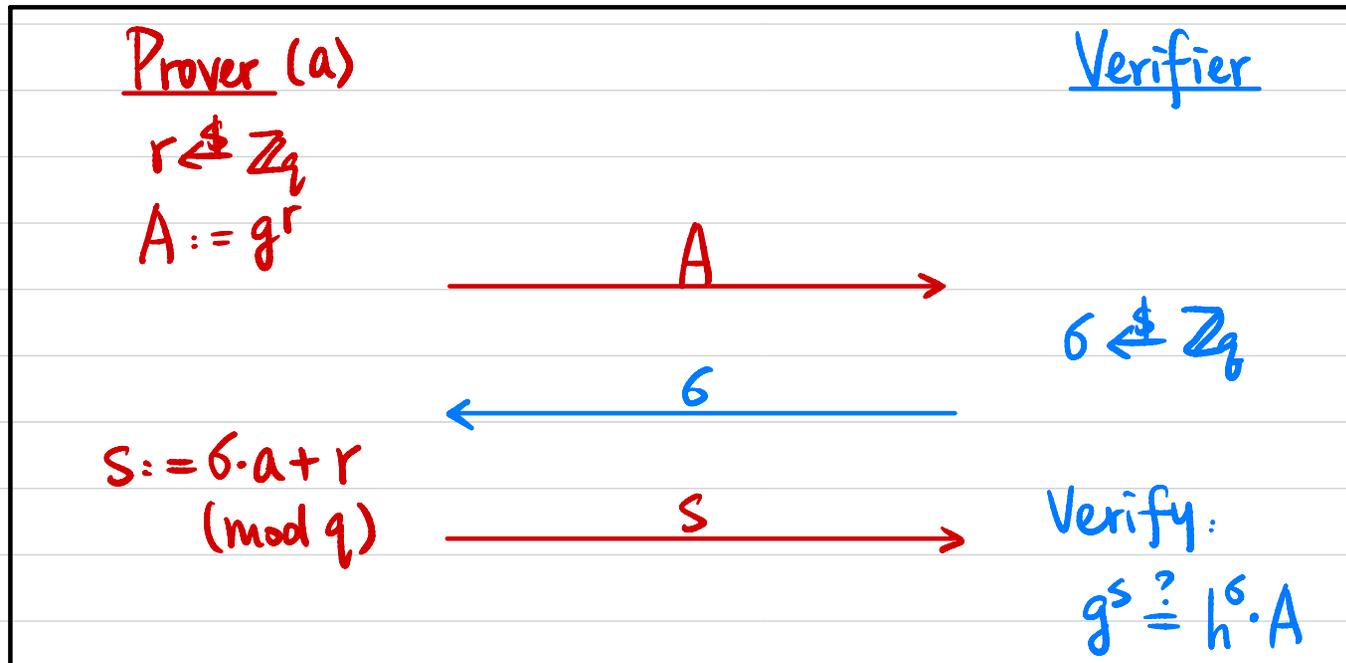


$\Pi = (m_1, m_2, m_3)$

Example: Schnorr's Identification Protocol

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

Prover's secret: a

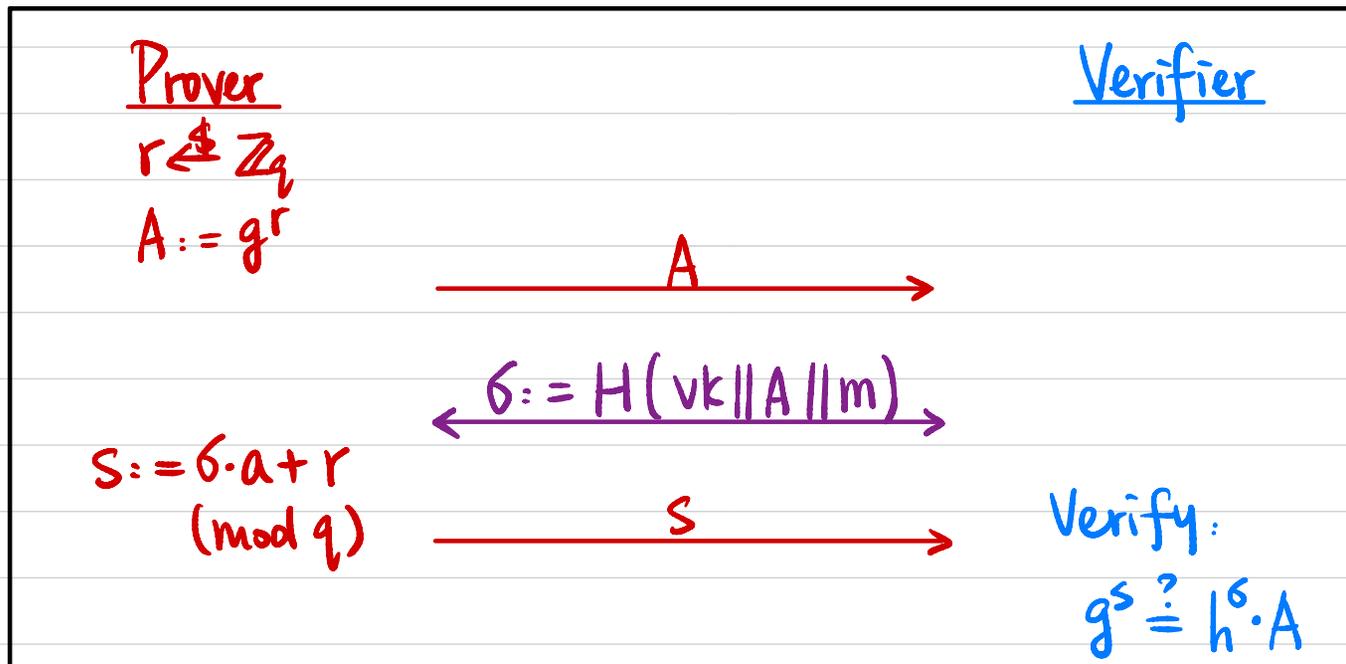


Fiat-Shamir Heuristic

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

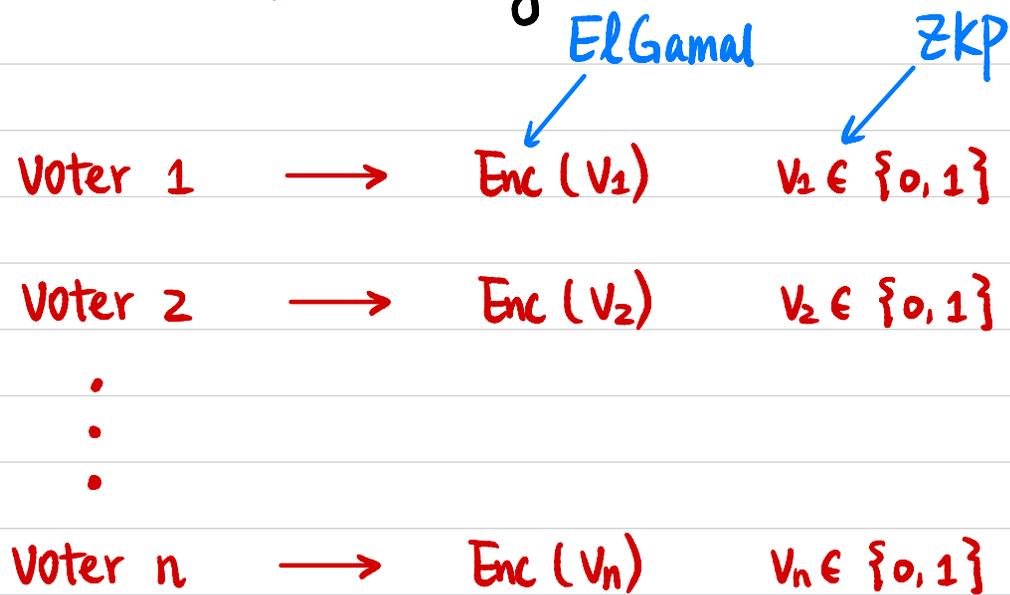
Cyclic group G of order q , generator g

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



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

Anonymous Online Voting



Enc ($\sum V_i$)



Decrypt to $\sum V_i$

Additively Homomorphic Encryption

$$\begin{array}{l} \text{Enc}(m_1) \\ \text{Enc}(m_2) \end{array} \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \text{Enc}(m_1 + m_2)$$

Additively Homomorphic

$$\begin{array}{l} \text{Enc}(m_1) \\ \text{Enc}(m_2) \end{array} \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \text{Enc}(m_1 \cdot m_2)$$

Multiplicatively Homomorphic

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

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

Exponential ElGamal:

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

Correctness of Encryption

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

Public key $pk \in G$. ← public

Ciphertext $C = (c_1, c_2)$ ←

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

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

↑ (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