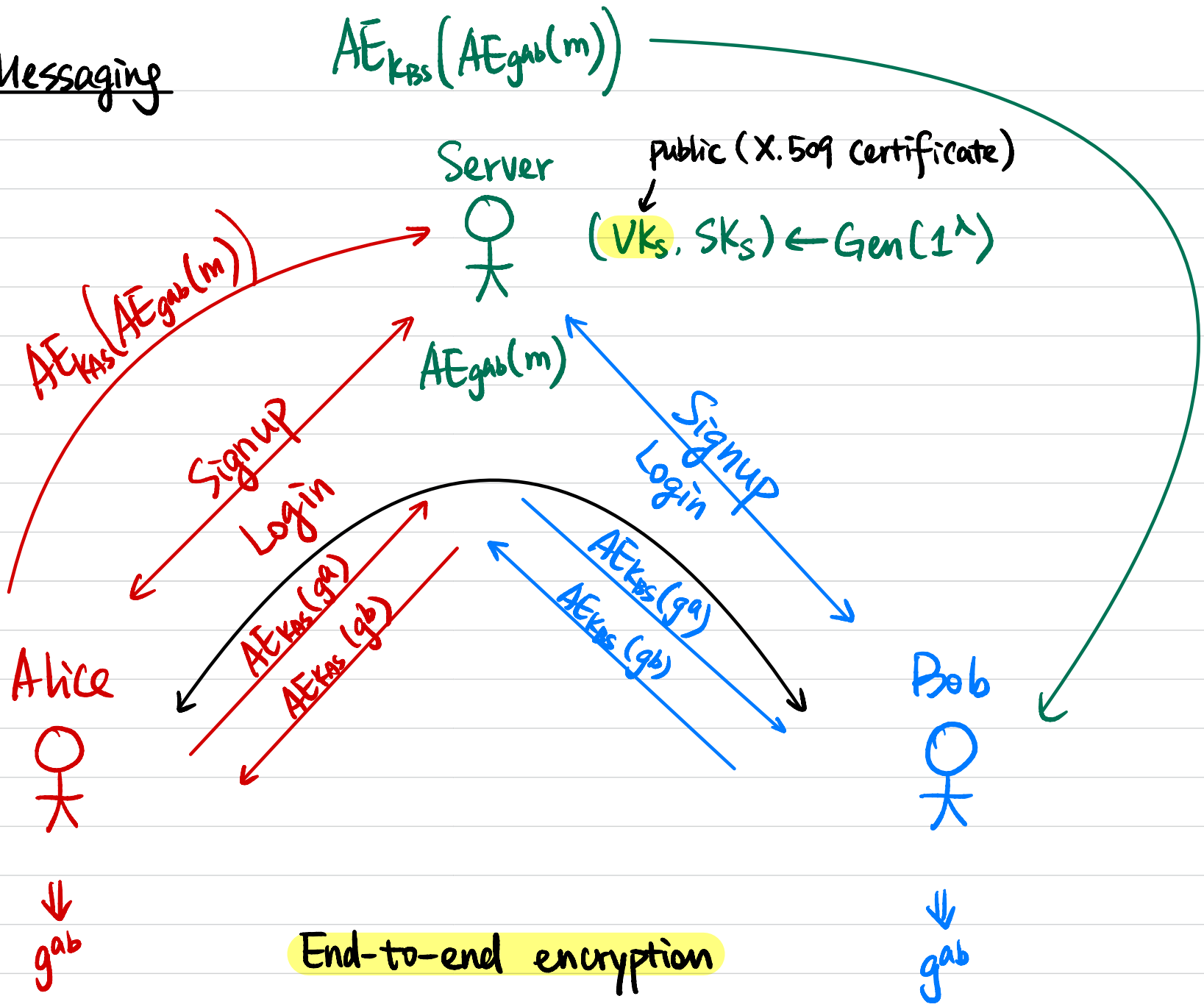
CSCI 1515 Applied Cryptography

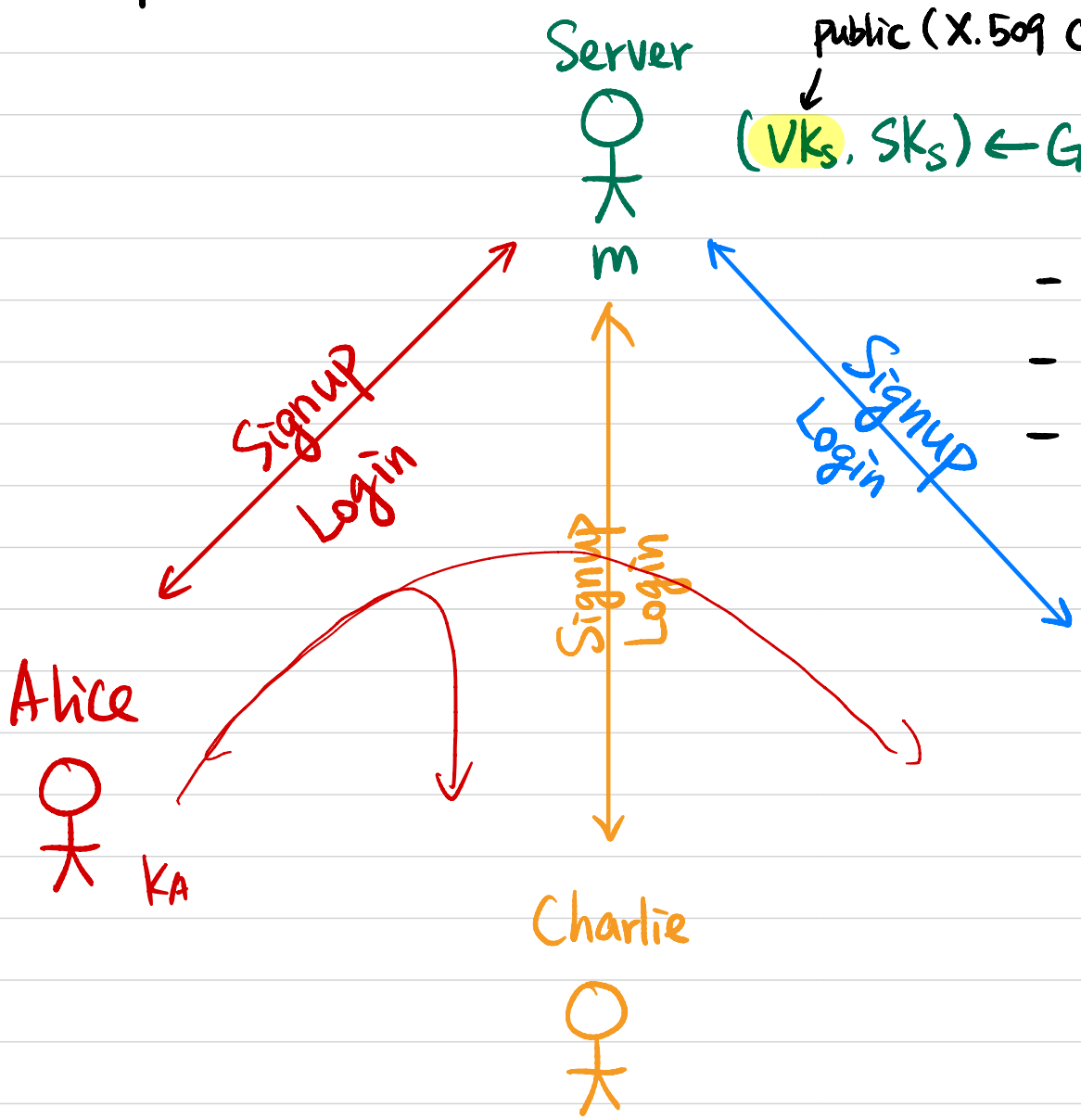
This Lecture:

- Case Study: Group Chat (Continued)
- Single Sign-On (SSO) Authentication
- Zero-Knowledge Proof
- Example: Diffie-Hellman Tuple

Secure Messaging



Group Chat ?



- m revealed to server ?
- group structure revealed to server?
- same key / different keys ?

How would you design it ?

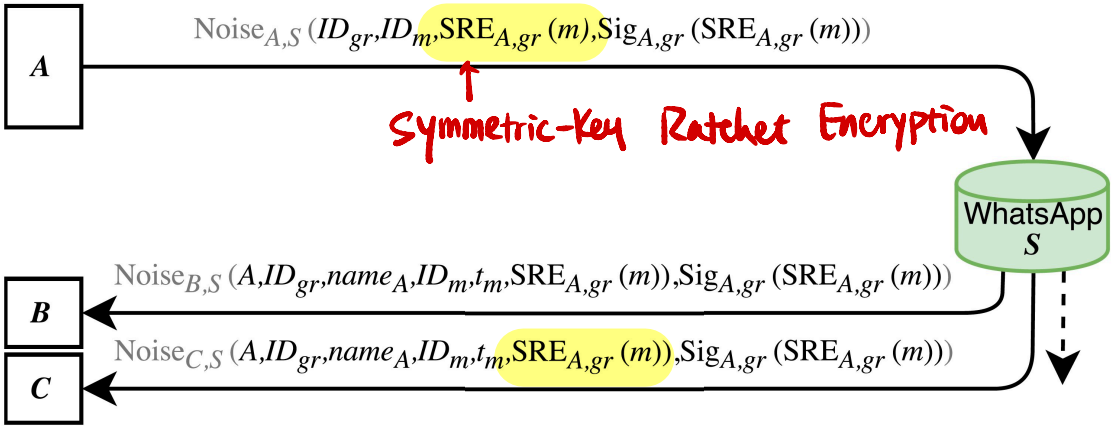


Figure 5. Schematic depiction of traffic, generated for a message m from sender A to receivers B, C in group gr with $\mathcal{G}_{gr} = \{A, B, C\}$ in WhatsApp.

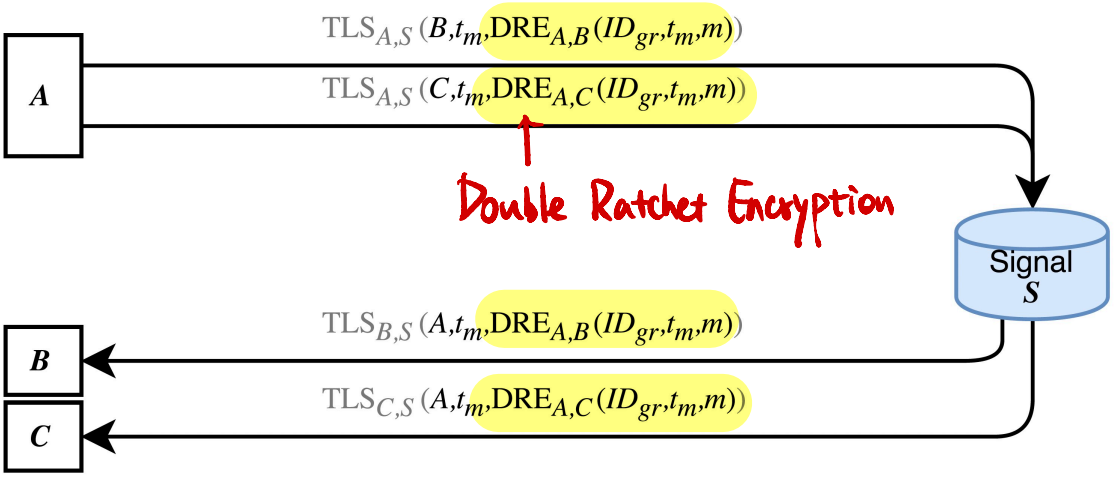
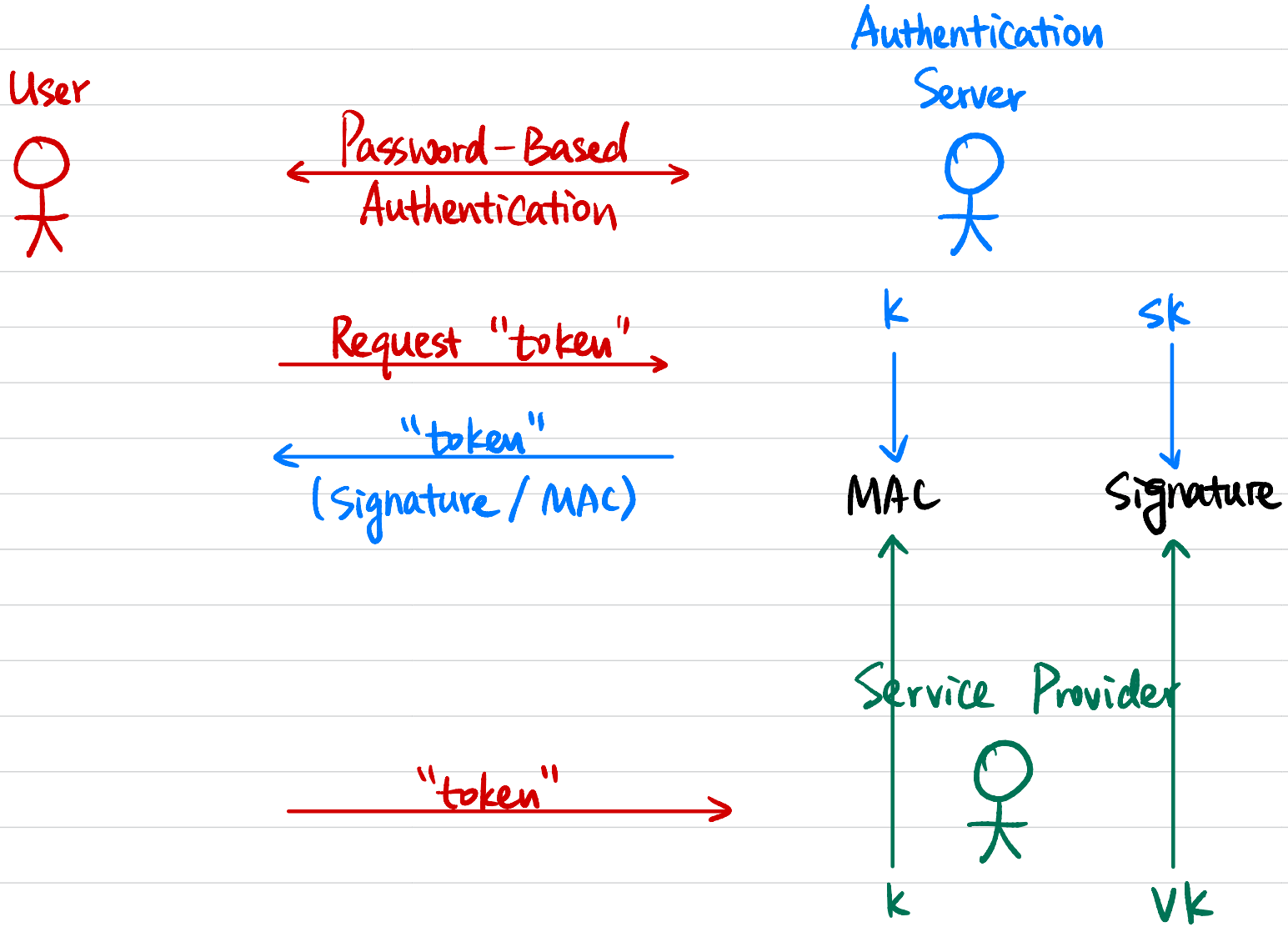


Figure 3. Schematic depiction of Signal's traffic, generated for a message m from sender A to receivers B and C in group gr with $\mathcal{G}_{gr} = \{A, B, C\}$. Transport layer protection is not in the analysis scope (gray).

Single Sign-On (SSO) Authentication



- OAuth / OpenID : Sign-in with Google / Apple / ···
- Kerberos: enterprises

Zero-Knowledge Proofs

[Coca-Cola & Pepsi taste differently]

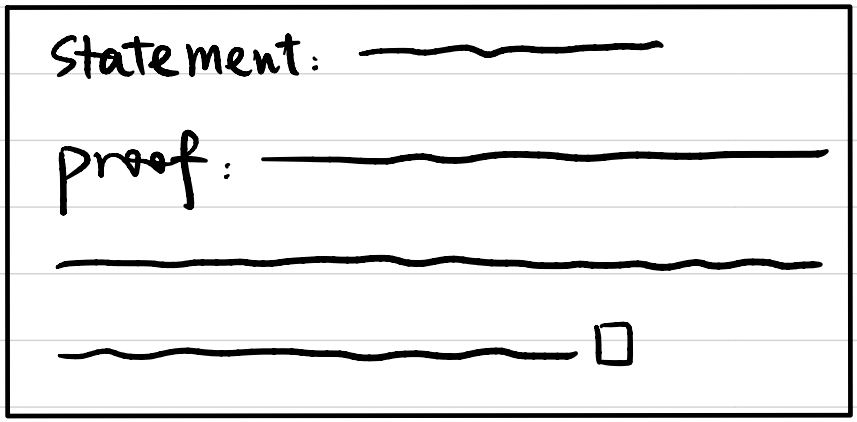
[There is a bug in your code]

[I have the secret key for this ciphertext]

What is a proof?

What does zero-knowledge mean?

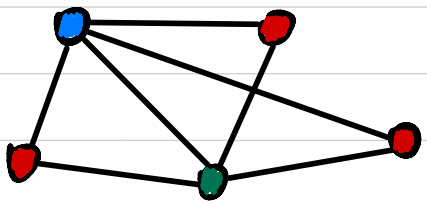
What is a "proof system"?



- **Completeness:** If statement is true, then $\exists$ proof that proves it's true.
- **Soundness:** If statement is false, then $\forall$ proof can't prove it's true.

NP as a Proof System

Ex: Gragh 3-Coloring



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

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

graph (↑ G) 3-coloring (↑ 3COL)

Statement: graph G

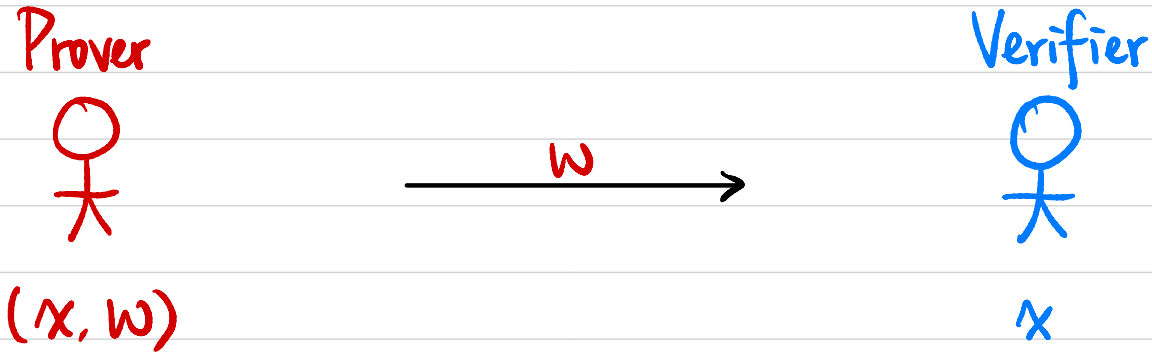
Proof: 3-coloring of G : 3COL

$(G, 3COL) \in R_L$

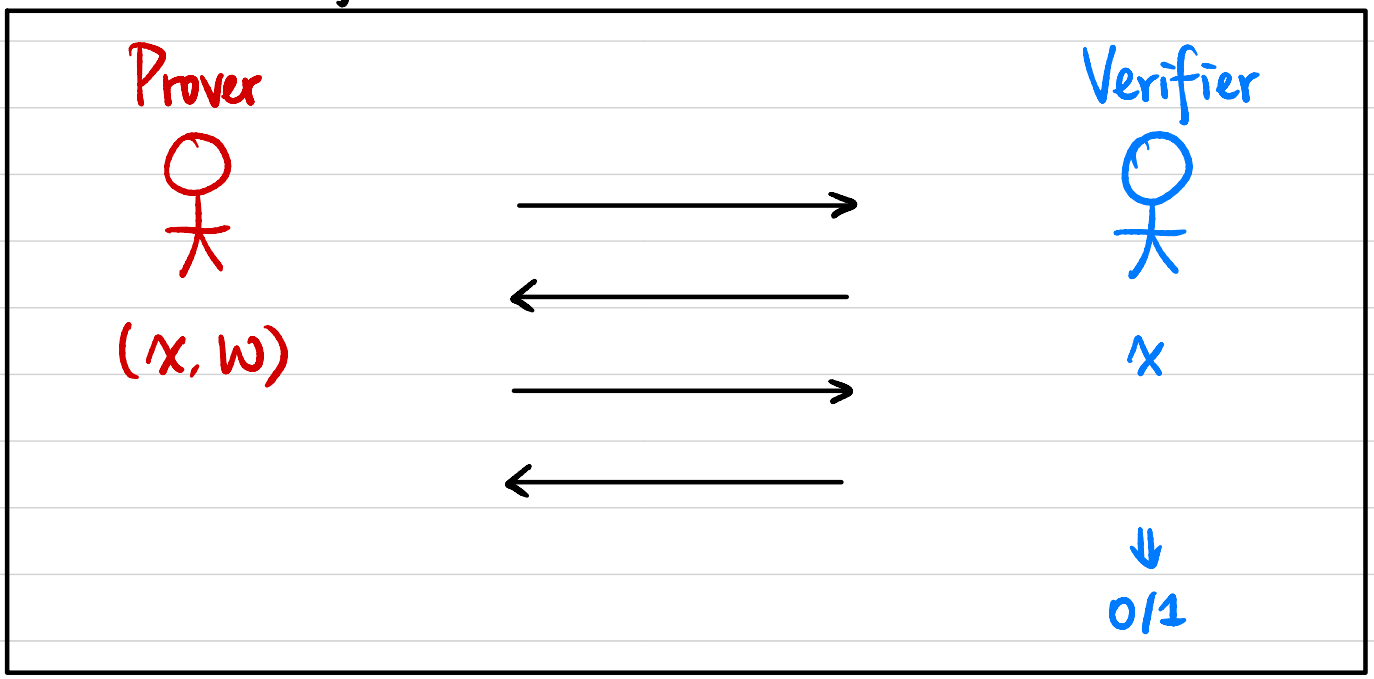
NP as a Proof System

A language L is in NP if $\exists$ poly-time V s.t.

- **Completeness:** $\forall x \in L$, $\exists w$ s.t. $V(x, w) = 1$
  (w: witness)
- **Soundness:** $\forall x \notin L$, $\forall w^*$, $V(x, w^*) = 0$



Zero-Knowledge Proof (ZKP)



Let (P, V) be a pair of probabilistic poly-time (PPT) interactive machines.

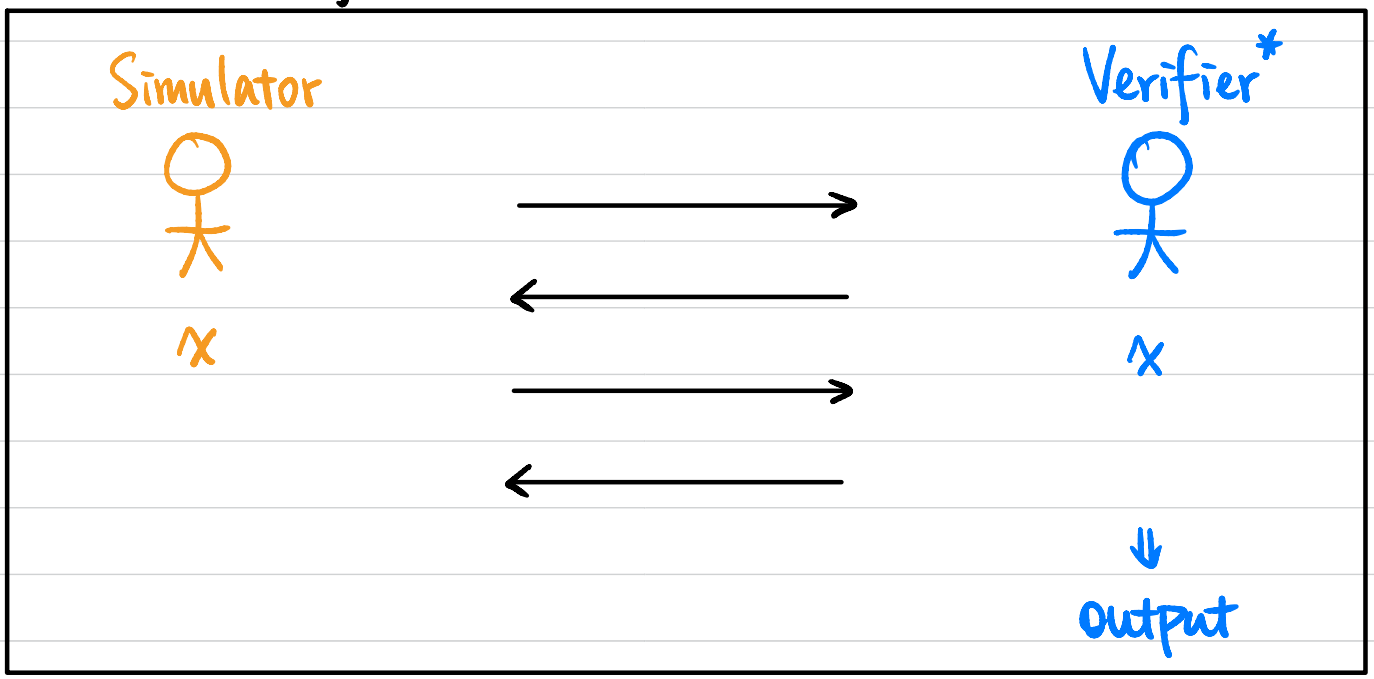
(P, V) is a **zero-knowledge proof system** for a language L with associated relation R_L if

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

  (PPT): argument

- **Zero-Knowledge?**

Zero-Knowledge Proof (ZKP)



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

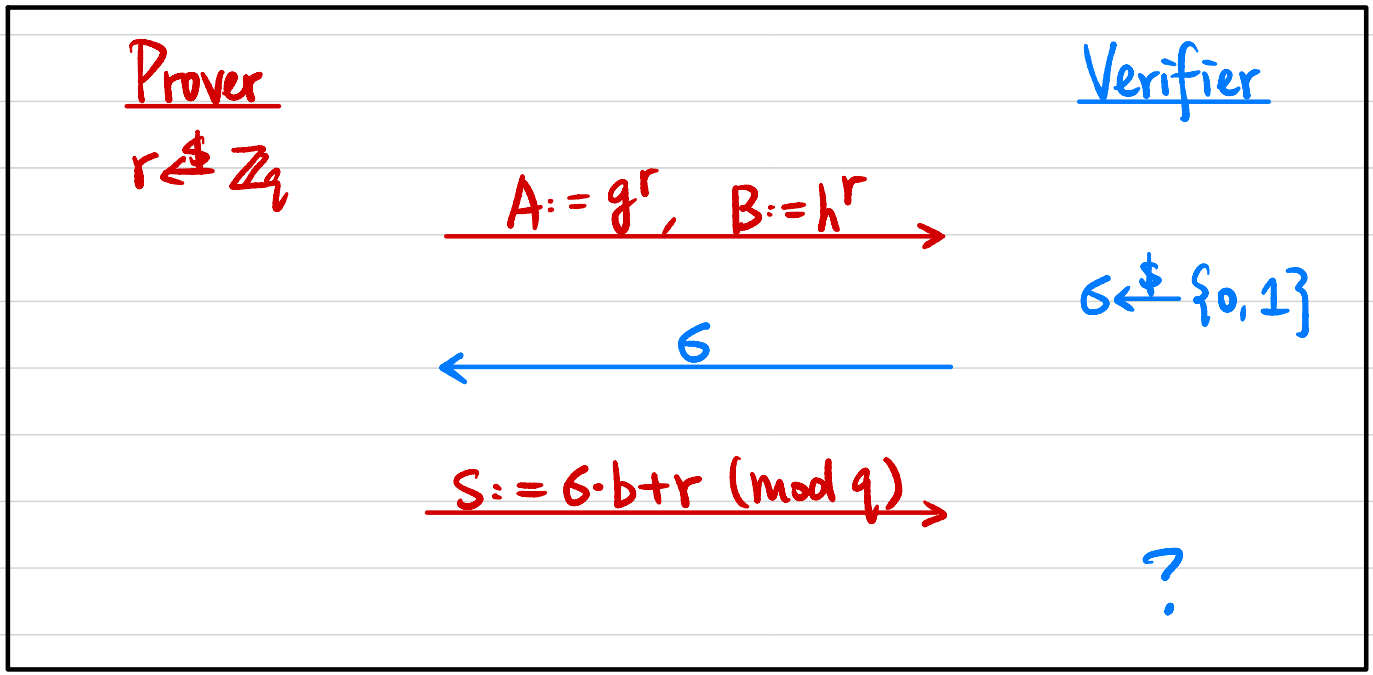
$$Output_{V^*}[P(x,w) \longleftrightarrow V^*(x)] \simeq S(x)$$

Example: Diffie-Hellman 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 ?

If $\sigma = 0 \Rightarrow s = r \Rightarrow$ Verify $A = g^s,\ B = h^s$

If $\sigma = 1 \Rightarrow s = b + r \Rightarrow$ $u \cdot A = g^s,\ v \cdot B = h^s$

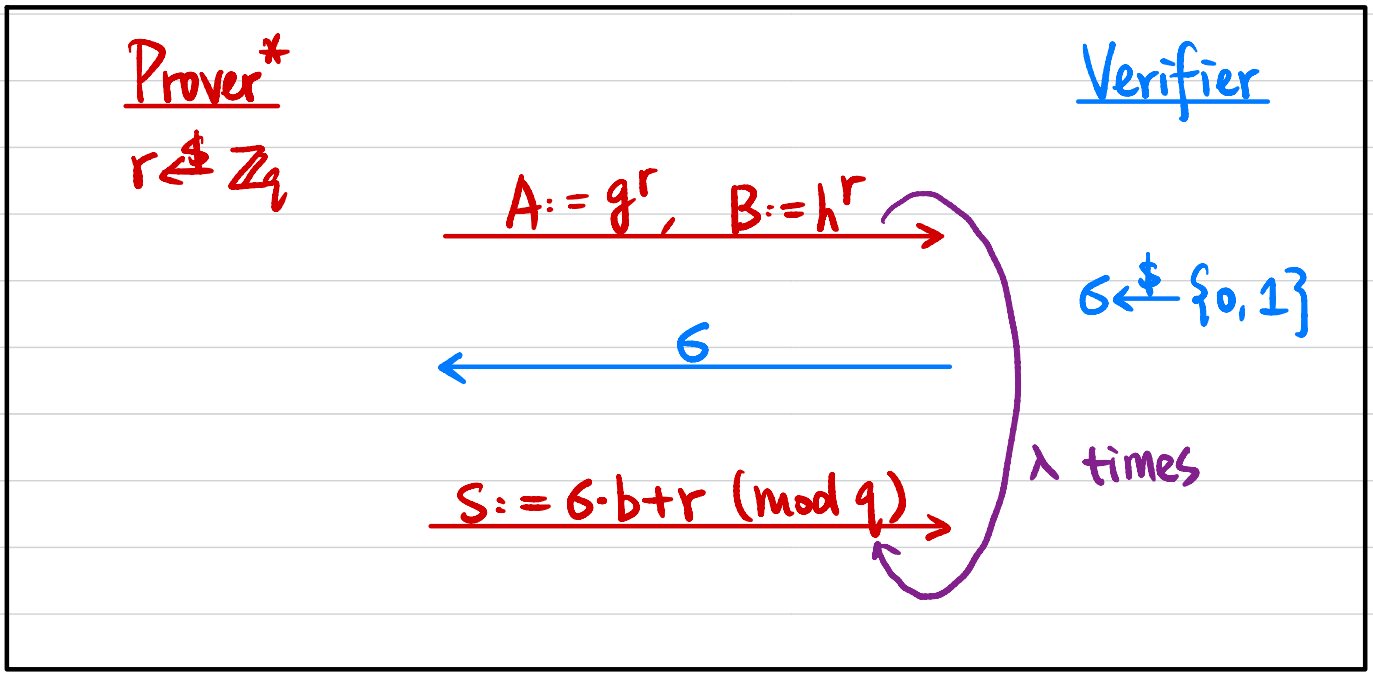
Soundness? $(g, h, u, v) \notin L$

$g^a \quad g^b \quad g^c$

in λ iterations $\leq (\frac{1}{2})^\lambda$

in a single iteration $\leq \frac{1}{2}$

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



If $\sigma = 0 \Rightarrow S = r \Rightarrow$ Verify $A = g^S, \ B = h^S$

If $\sigma = 1 \Rightarrow S = b + r \Rightarrow$

$s \xleftarrow{\$} \mathbb{Z}_q$

$u \cdot A = g^S, \ v \cdot B = h^S$

$A := g^S / u, \ B := h^S / v$

① $A = g^r, \ B = h^r$ valid

If $\sigma = 1 \Rightarrow$ caught

s s.t. $u \cdot g^r = g^s$

$v \cdot h^r = h^s$

$\Downarrow$

$u = g^{s-r}$

$v = h^{s-r}$

$\Downarrow$

$b = s - r$

DOES NOT EXIST!

② $A = g^{r_1}, \ B = h^{r_2}$ invalid

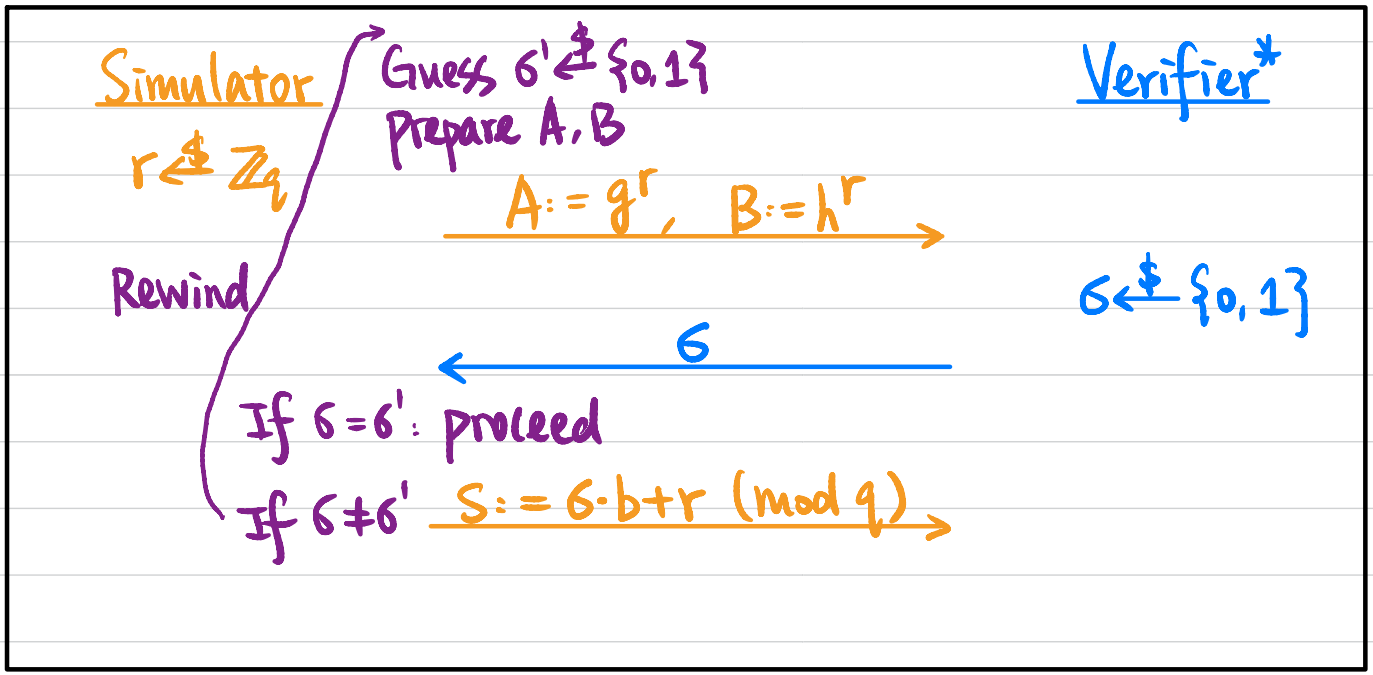
$r_1 \neq r_2$

If $\sigma = 0 \Rightarrow$ caught

Zero-Knowledge?

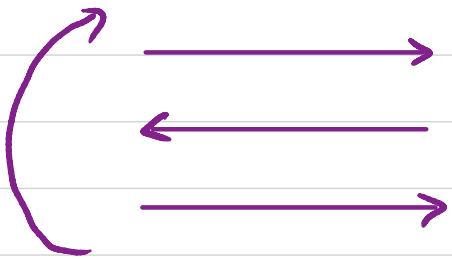
$\forall$ PPT V^*, $\exists$ PPT S s.t. $\forall (x, w) \in R_L$,

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



Rewind $\leq \lambda$ times:

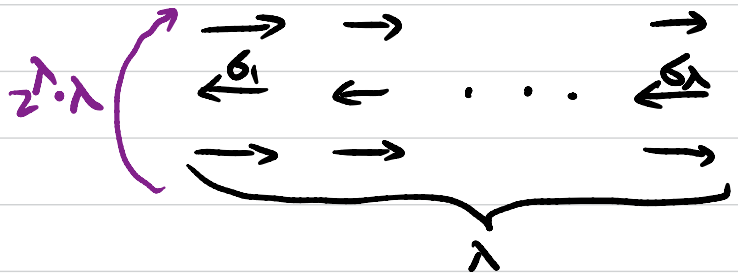
$\Pr[\text{success}] \geq 1 - (\frac{1}{2})^\lambda$



Parallel Repetition?

$\sigma_1', \cdots \sigma_\lambda'$

$\Pr[\sigma_1 = \sigma_1', \cdots, \sigma_\lambda = \sigma_\lambda'] = (\frac{1}{2})^\lambda$



NOT ZK!