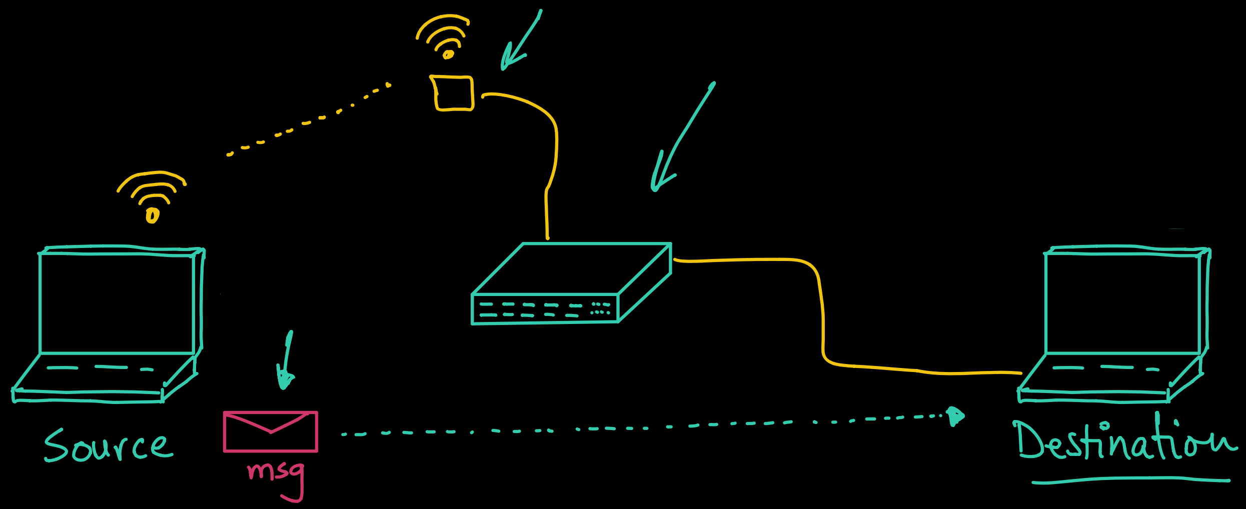


" 131 , SECTOR G5, PHASE 2"

Cryptography

≈ $\boxed{131}$

↳ Plaintext → Cypher Text → Plain text

Encryption

Decryption

131

$\downarrow$ +2

353

+1

$\boxed{242}$

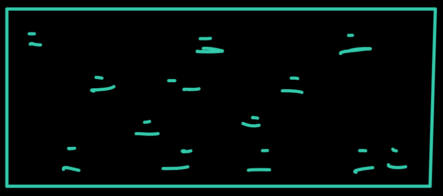
"substitution Cypher"

↳ Algo for performing encryption

x————————x

Easy to break :

most commonly occurring letter : E



Carefully ←

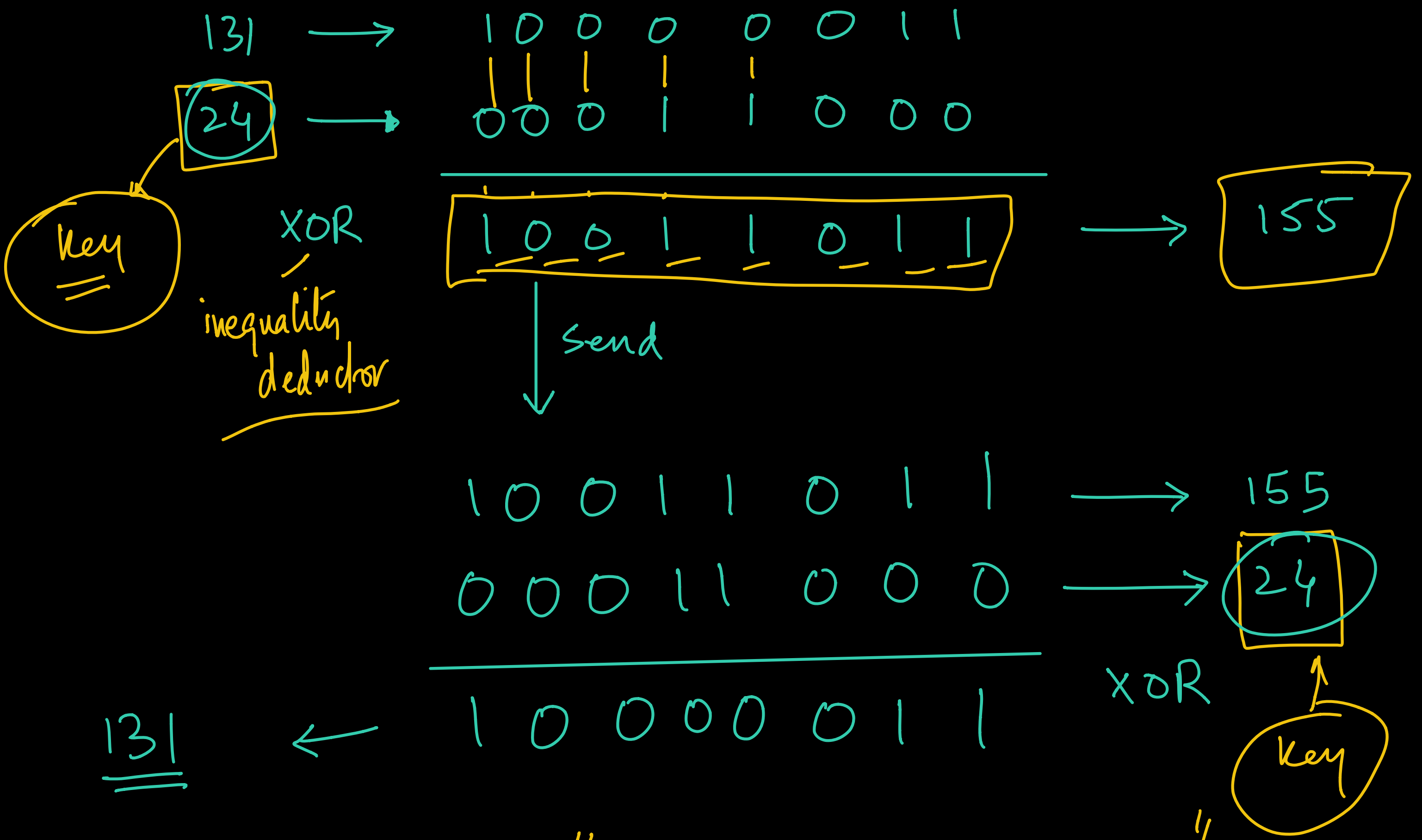
↓ ↓

ectgwnna

e

↓ $\boxed{+2}$

g



"Shared key", "Symmetric key"

"Private key"

8-bit key

keylength

crypto

— Cryptanalysis

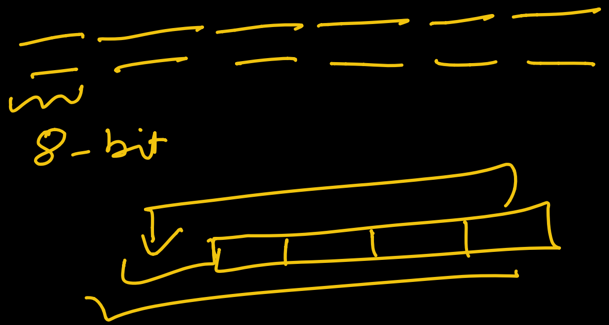
A

$A \text{ XOR } \underbrace{K \text{ XOR } K}_{0}$

$$\begin{array}{c}1\\0\\\hline 1\end{array}\qquad\begin{array}{c}0\\0\\\hline 0\end{array}\qquad\begin{array}{c}00\\00\\\hline 00\end{array}$$

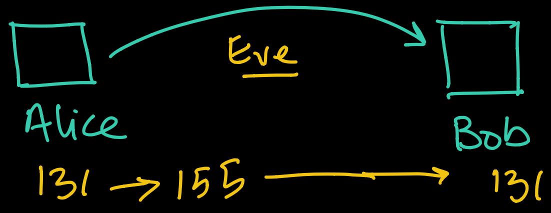
$$\begin{array}{c}10\\10\\\hline 00\end{array}$$

$$\begin{array}{c}01\\01\\\hline 00\end{array}$$



Public key Crypto:

message: $\boxed{2}$



Bob's

$p = 2$ } prime

$q = 7$

$$\boxed{n = p * q = 14}$$

$$\varphi \quad r = (\overset{1}{p-1})(\overset{6}{q-1}) = (1)(6) = \overset{\downarrow}{6}$$

Need to find e and d

e need to be co-prime with r

"No shared factors"

6 : ~~1~~, ✓2, ✓3, 6

× 2 : 1, $\boxed{2}$

× 3 : 1, $\boxed{3}$

× 4 : 1, $\boxed{2}$, 4

✓ 5 : 1 5

$\approx$ GCD(e,r)==1

$$\boxed{e = 5}$$

↳ public key

Publish: $(\overset{e}{5}, \overset{n}{14})$

Alice: $\boxed{2}$

msg → $2^{\overset{e}{5}} \bmod \underset{\uparrow n}{14} = \boxed{4}$ → sends to Bob

$(6^{5} \bmod 14 = 6)$

$\boxed{4}$ → enc

Bob:

computes d such that: d → private key

$$\boxed{\overset{5}{e} \cdot d \bmod \overset{6}{r} = 1}$$

$e = 5$

$r = 6$

$n = 14$

$$\boxed{d = 11}$$

d →	1	2	3	4	5	....	11
	×	×	×	×	✓		✓
5:	5	10	15	20	25		55
	5	4	3	2	1		1

To get message

$$\boxed{enc^{d} \bmod n}$$

$4^{11} \bmod 14 = 2$

$(6^{11} \bmod 14 = 6)$

Reason:

$$enc^{d} \bmod n$$

$$\left(msg^{\cancel{e}}\right)^{\overset{1}{\cancel{d}}} \bmod n$$

$$msg \bmod n$$

(need the conditions to hold for this cancellation — fermat's little theorem etc.)

Rivest

Shamir

Adelman

This is the RSA algorithm: the backbone of all secure communication on the internet!!

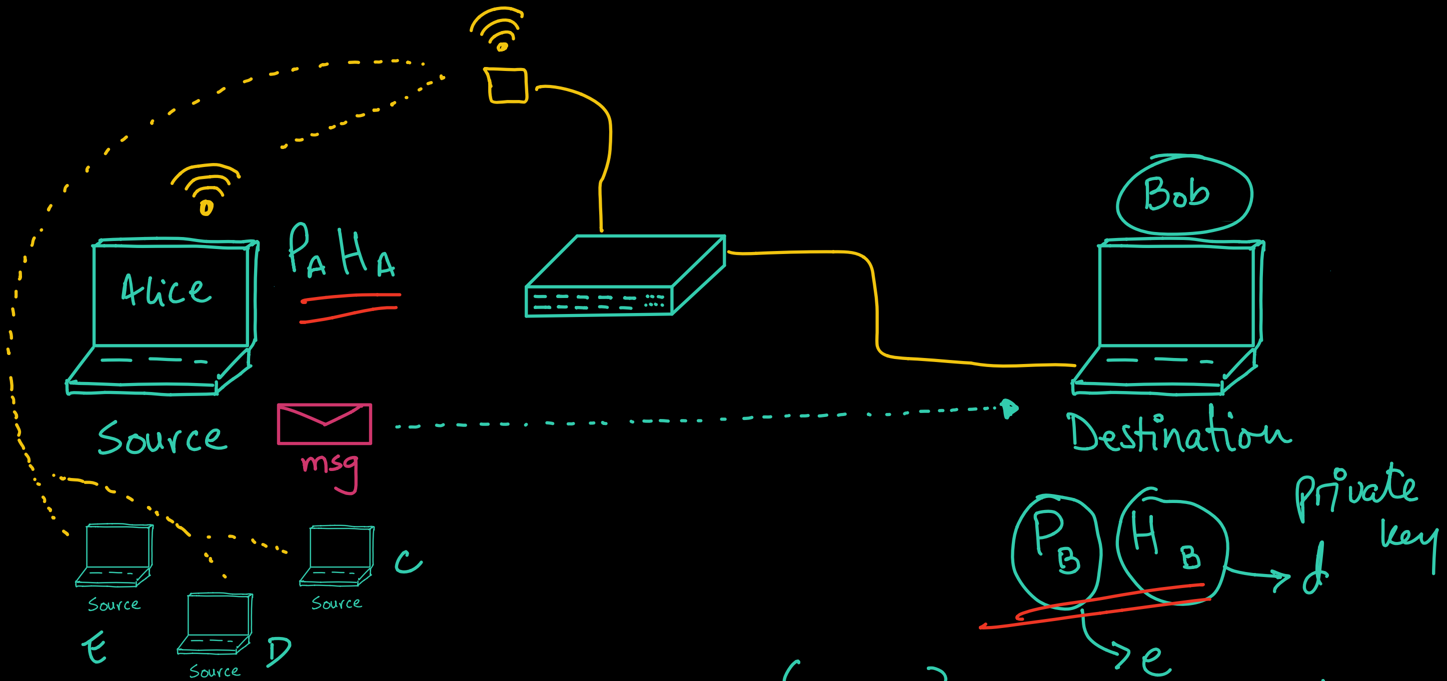
"prime factorization"

- why is it hard to break?
  - p and q are huuge!
  - easy to compute n from p and q
  - ⊛ very hard to go back.
  - need p and q to compute d from e and n

$$\boxed{\begin{array}{l} \underline{e.d \bmod r} = 1 \\ r = (p-1)(q-1) \end{array}}$$

Assymetric key crypto

Demo!



① A wants to send B a message : $\{msg\}_{P_B}$

B decrypts : $\{\{msg\}_{\cancel{P_B}}\}_{\cancel{H_B}}$

⇝ msg

② ~~B~~ A wants to sign a message : $\{msg\}_{H_{\cancel{B}A}}$

Anyone wants to verify : $\{\{msg\}_{H_{\cancel{B}A}}\}_{P_{\cancel{B}A}}$

⇝ msg

③ A wants to send B a message But B needs verification that message is from A

a) Sign message : $\{m\}_{H_A}$

b) Encrypt for B : $\{\{m\}_{H_A}\}_{P_B}$

c) B opens the package

Decryption

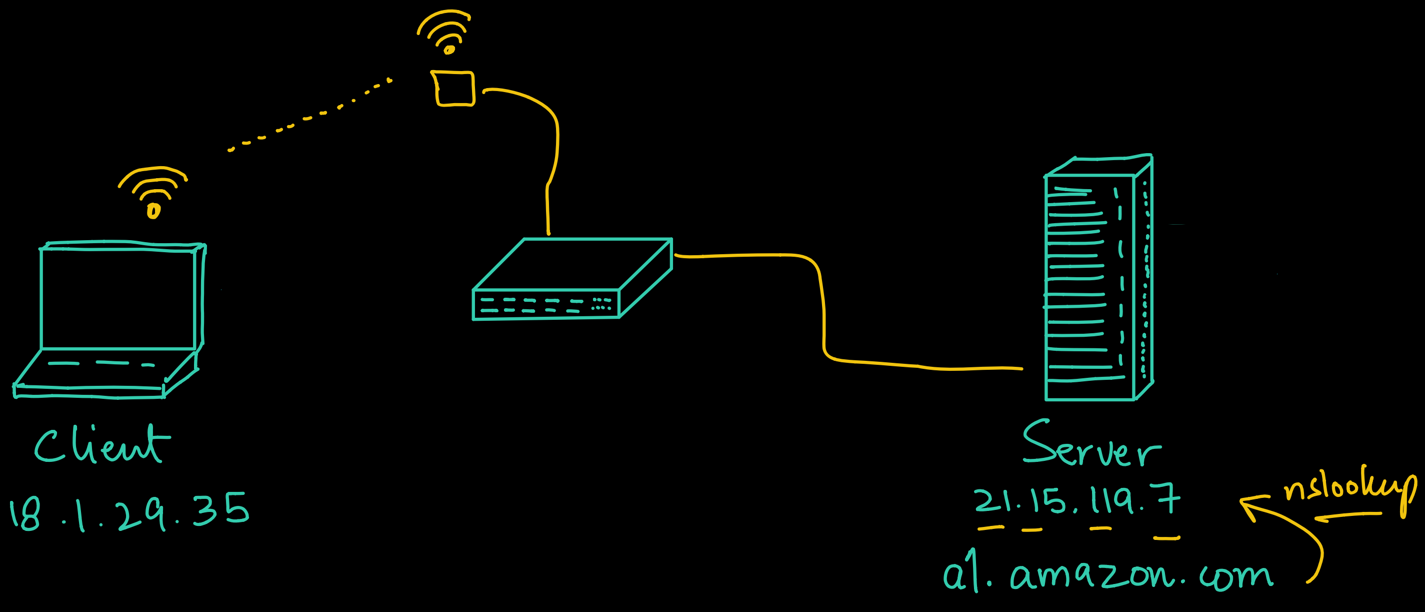
$$\{\underbrace{\overbrace{\{\{m\}_{H_A}\}_{\cancel{P_B}}}^{\text{Packet}}}\}_{\cancel{H_B}}$$

$$\{\overbrace{\{m\}_{\cancel{H_A}}}^{\text{sign}}\}_{\cancel{P_A}}$$

d) Verify the signature

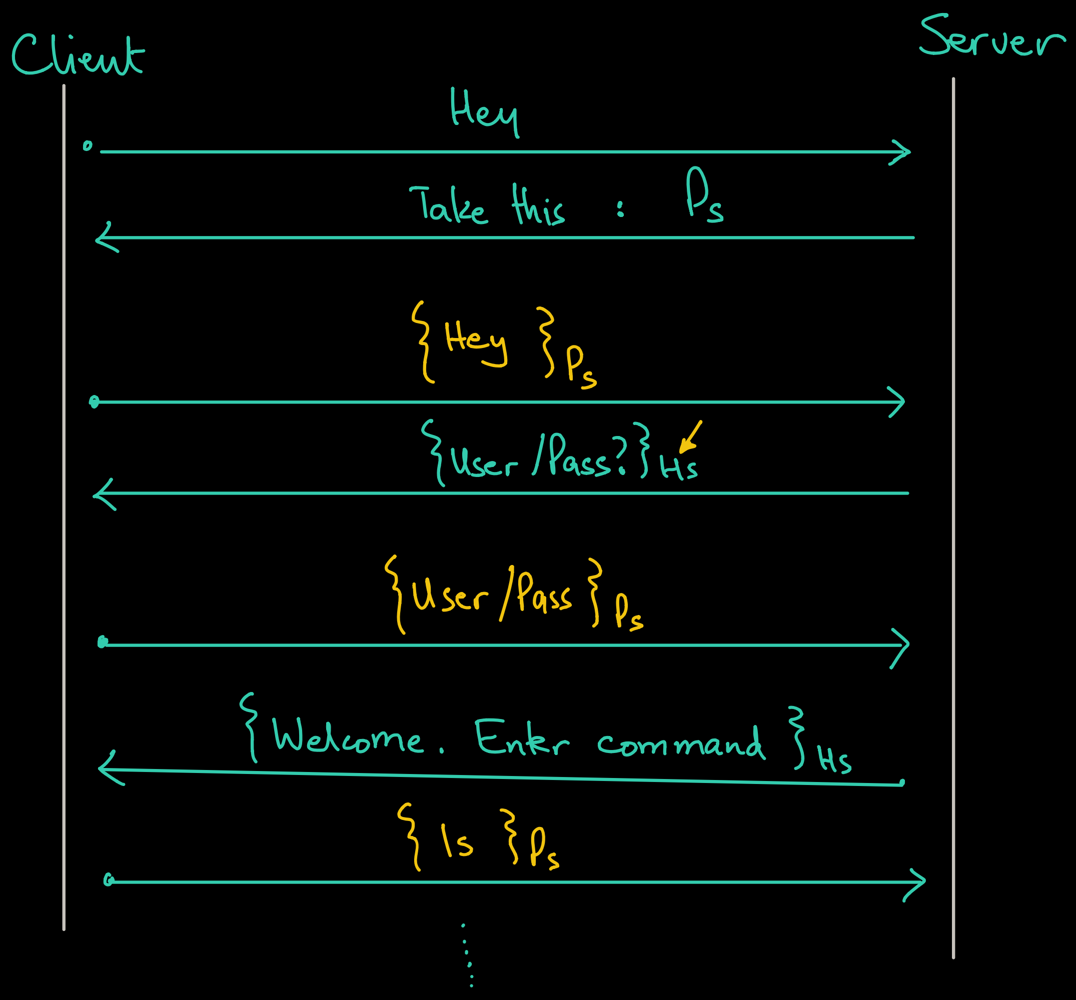
$$\{\{m\}_{\cancel{H_A}}\}_{\cancel{P_A}}$$

$$\boxed{m}$$

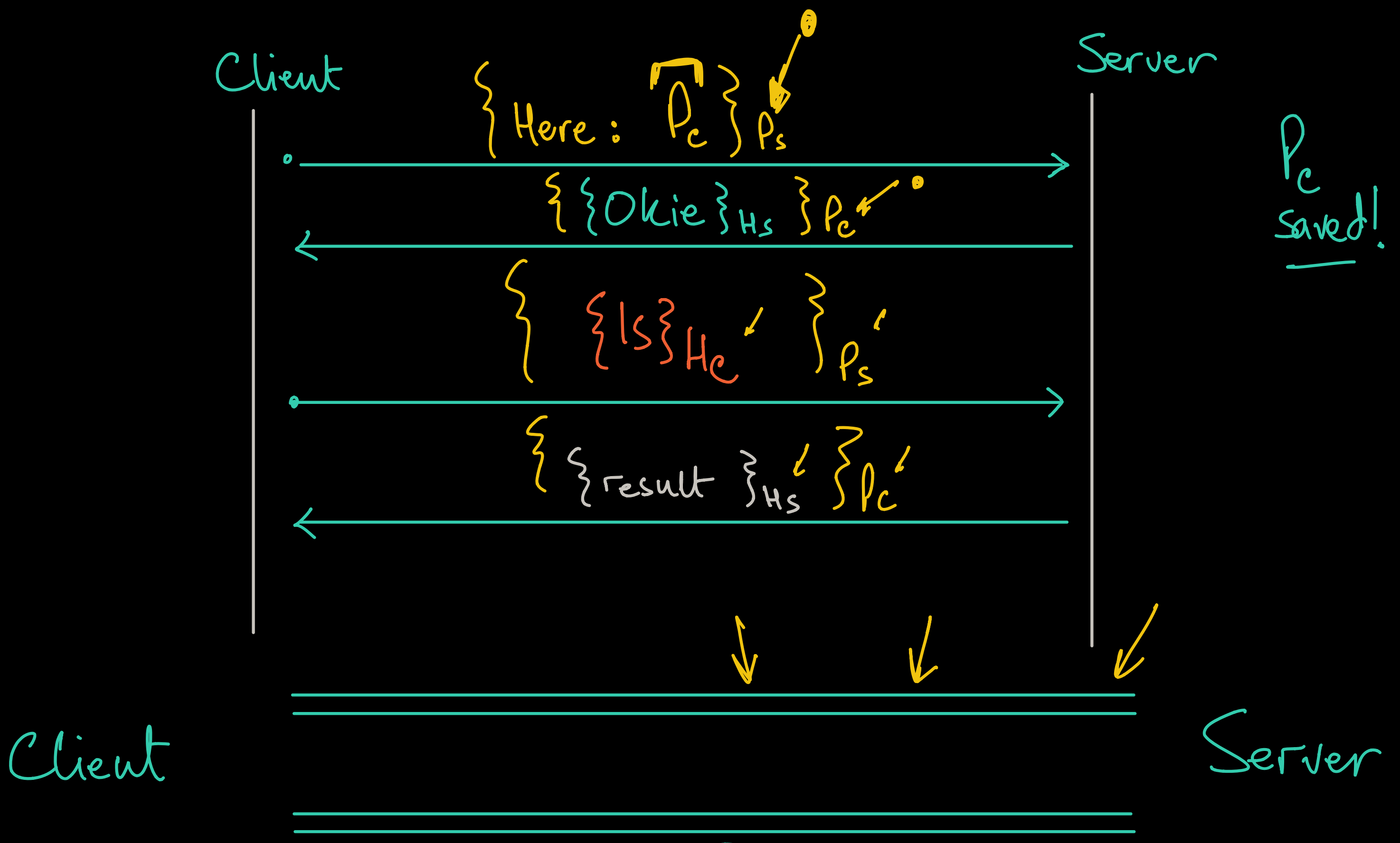

Client
18.1.29.35
Server
21.15.119.7
a1.amazon.com
nslookup

P_c , H_c

P_s H_s


Client
Server
Hey
Take this : P_s
$\{Hey\}_{P_s}$
$\{User/Pass?\}_{Hs}$
$\{User/Pass\}_{P_s}$
$\{Welcome. Enter command\}_{Hs}$
$\{ls\}_{P_s}$

Verifying the client


Client
Server
$\{ \text{Here} : P_c \} P_s$
$\{ \{ \text{Okie} \}_{H_s} \} P_c$
$\{ \{ ls \}_{H_c'} \} P_s'$
$\{ \{ \text{result} \}_{H_s'} \} P_c'$
P_c saved!
Client
Server

Secure Tunnel

SSH

Secure shell

→ Python notebooks

Symmetric ✓

Assymmetric ✓