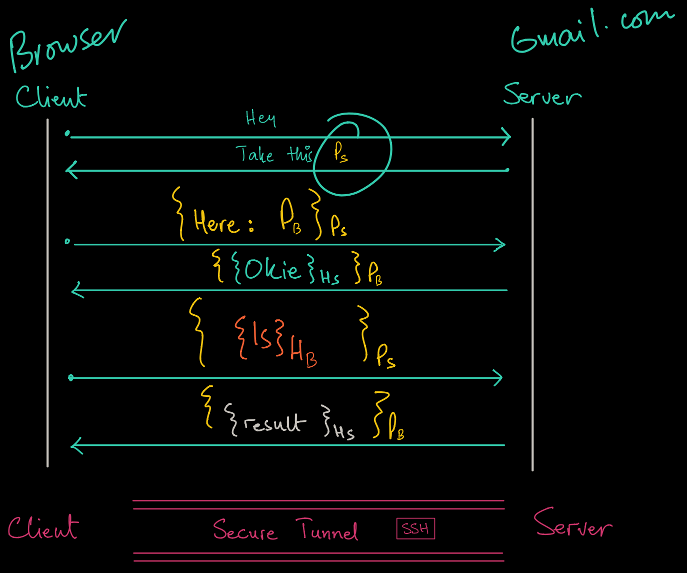
Secure Tunnels
(the missing piece)



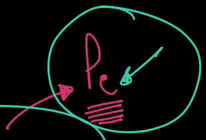
Server : gmail.com

IP : 172.217.9.229

— nslookup

Q: How do you verify that P_s is the public key of gmail.com?

— Just ask ?!   Nope!



— Do you trust someone?   Yes — verisign

— "Hey gmail! Get verisign to issue you a certificate of authenticity"

"Certification Authority (CA)"

— Gmail goes to verisign (pays them)

$\{$ " gmail.com has P_s on IP 172.217.9.229 " $\}_{H_c}$

Digital Certificate $Cert_{gmail}$

— Gmail sends $Cert_{gmail}$ to Browser

$\{ \{ \}_{\cancel{H_c}} \}_{\cancel{P_c}}$ → "Gmail has P_s on IP ________"

But:

Verisign

- cannot verify the whole world! (is expensive too)
- We want to trust just verisign

- New company:

mycert — m P_m H_m

nu .edu.pk — n P_N H_N

$\{$ " nu.edu.pk has P_N on IP _____ " $\}_{H_m}$

$Cert_{nuces}$

Browser: $\{ Cert_{nuces} \}_{P_m}$

But how do I trust P_m

- same way I trusted P_S → Gmail's Public key

- ~~Gmail~~ MyCert goes to verisign (pays them)

$\{$ " MyCert has P_m " $\}_{Hc}$

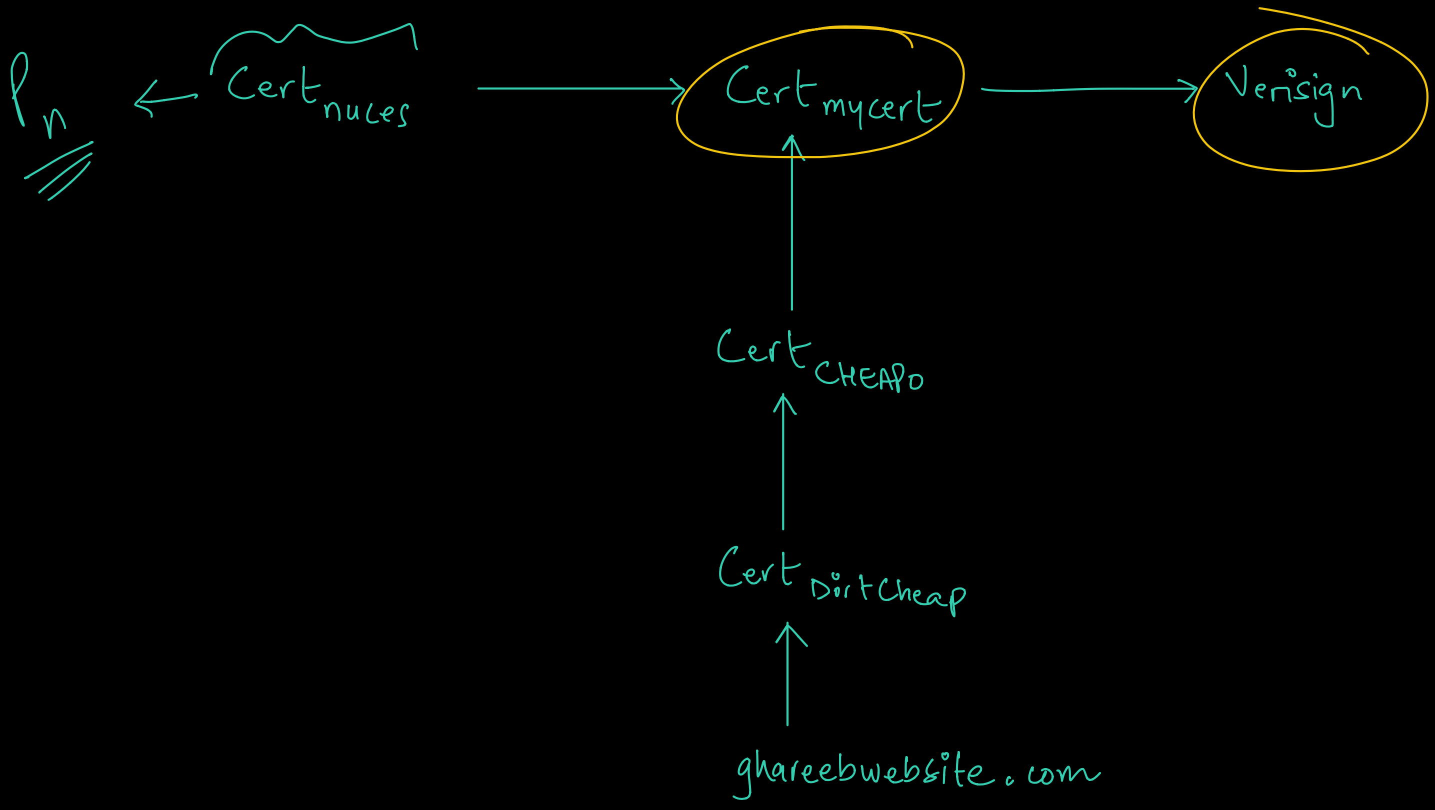
Digital Certificate $Cert_{mycert}$

Browser : $\{ Cert_{mycert} \}_{P_c}$ → this, we trust

new fact : P_m is trusted

$\{ Cert_{nuces} \}_{P_m}$ → now trusted

new fact : "nu.edu.pk has P_n on IP ____"



"Certificate chain".