ssh / SMTP } can use (Application protocol)
mail } TLS

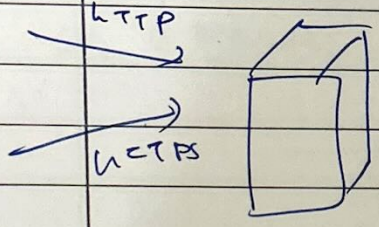
__/__/__

→ Each ticket will have certain age

HTTP over TLS

part - III : HTTPS & the lock

→ Integrating HTTPS into the browser



support legacy devices

connection initiation

① TCP } handshake
② TLS }

③ HTTP request

connection close

reverse order

connection: close header

close notify alert

FIN & Ack

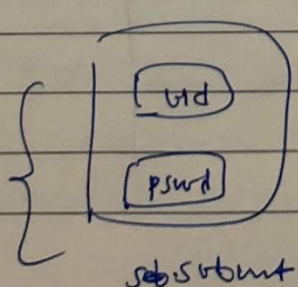
Integrity of TLS/HTTPS

- historically → sparing encryption is costly (only when needed)

Just in time HTTPS

using to send Just Config. Info

conf.



JIT HTTPS :-

client ← HTTP response server

TLS/SSL stripping Attacks

~~for~~ JIT HTTPS (HTTPS) post

HTML file present front

JIT HTTPS ⟶ POSSIBLE WITH SSL

__/__/____

Browser normally forces every session to HTTPS

TCP PIPE

HTTPS CONNECT

domain name | Port

HTTP connect request in plain text

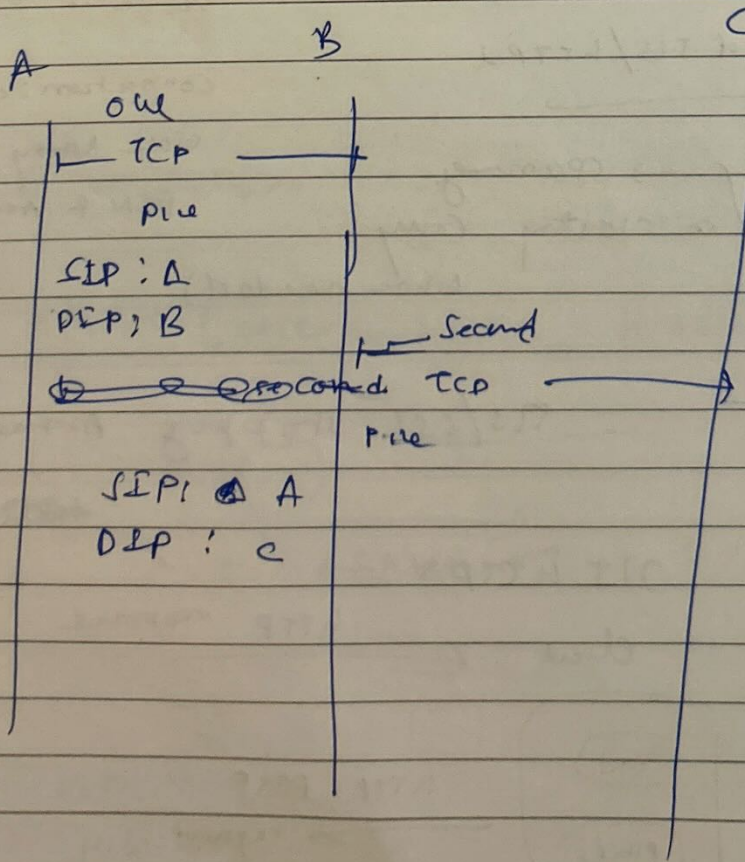
connect

HTTP/1.0 200

same tcp pipe

chat happens

- 2 TCP
- 1 TLS



- similar to NAT, bypass / partially proxy working. __/__/____

similar bypass proxy does

① Intercept HTTP/HTTPS traffic → TCP → TLS

2 ways to do

① Enterprises [Squid] sitting in organization
② End host [Burp suite] sitting on computer

Browser (A) ⟷ Burp (B) ⟷ google.com

A → B: CH
S = A
D = P

B → A: SH + fake certificate signed by proxy
S = P
D = A

fake certificate

SN = google.com
Issuer = proxy
pk_{google} = temp

signed by proxy] private key

for HTTPS request

→ Caching benefit but controversy in privacy related issues

no benefit for

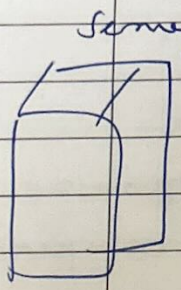
90% uses bypass for VBV buttons

__/__/____

Integrating SSL/TLS with HTTP : HTTPS

(*) (1) Two complications

(1) virtual hosting



Two site hosted @ same ip address

more than one web site on same IP address

CH →

← Ser cert ?

SSL ?

TLS 1-1

client_hello_extension:

servername : name)

But as they go as plaintext

Middle or Firewall knows what you are surfing

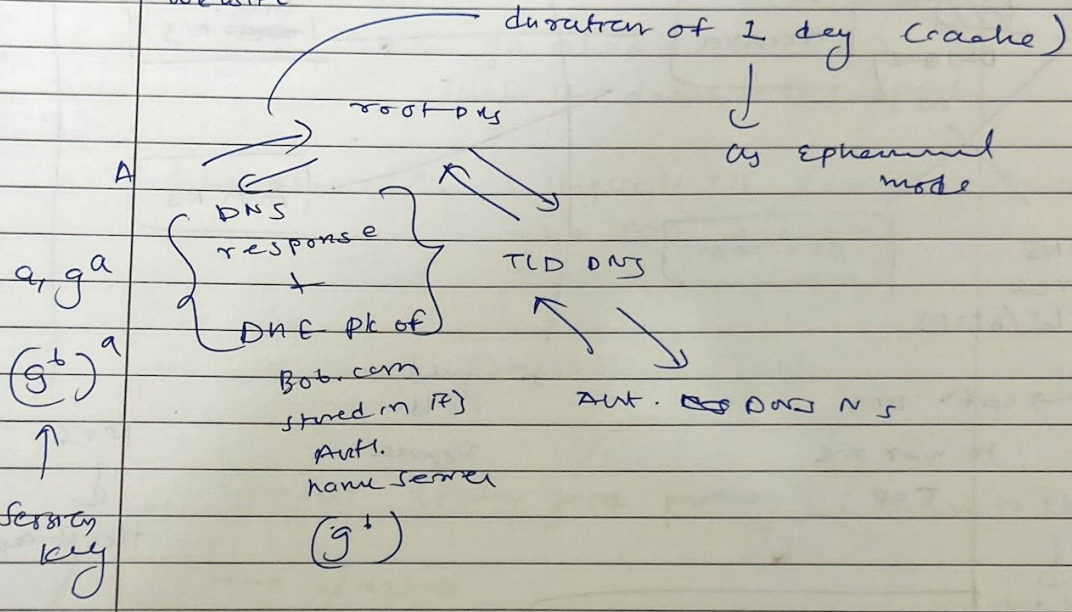
SSL ?

encrypted_server_name:

But how ?

DNS coming into the picture

We need Ip address before communicating to any website



But problems

so blocking very domain

high pe. web → 100's of lookups of Ip come to domain

so no middle boxes can know which website you are visiting

can't try brute very Ip's

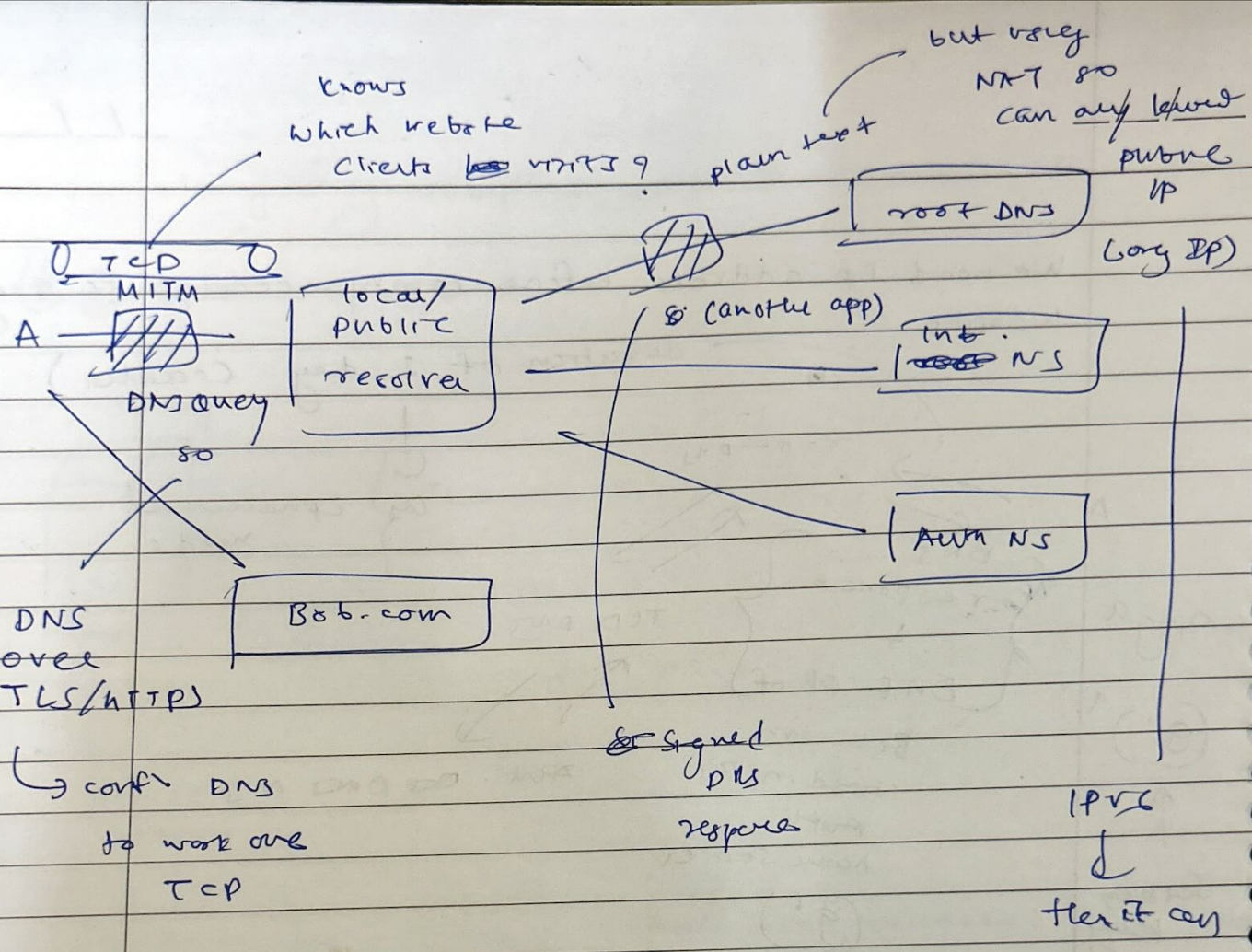
① single Ip for multiple website

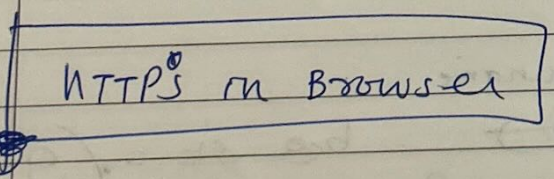
② single domain → multiple Ip

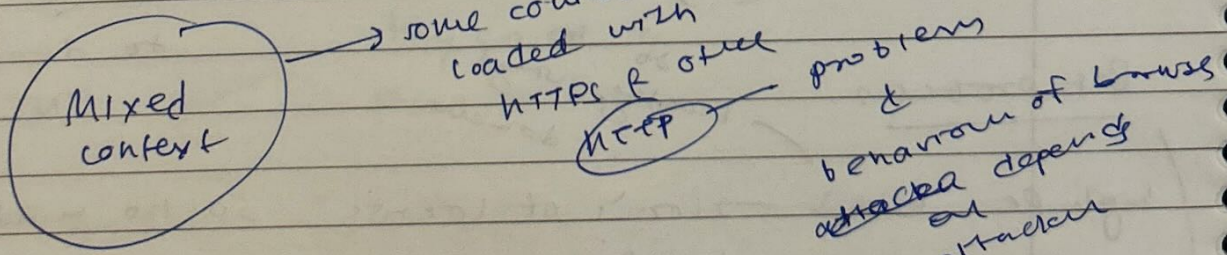
Still DNS is encrypted

① DNSSEC

② DNS over TLS or HTTPS







problems with HTTPS & Defenses

SMTP extension: STARTTLS

DSN
(Delivery Status Notification)

→ effective channel / pipe with.
→ no encryption to flow
→ This is what content ISP wanted to do

What MITM → remove STARTTLS or modify 220 go ahead to show that it doesn't support

Taken by AT&T to apply filter

Alternative : Implicit TSL (not opportunistic TLS)
- no E2E encryption
- hop to hop encryption

for E2E Email & then digitally sign

→ PGP (Pretty good privacy) (Web of Trust) (similar to PKI)

→ S/MIME (similar to PKI)

similar approach but provided by User Agent

we have chain of certificates

A — B
Exchanged public key & prior TSL encrypted using that

A — B — C
A & C doesn't know each other by as (B) knows (C)
(A) trust (C)

GNU (PG) → privacy god (open source)

Always on SSL ! 301 redirect} (eg: new URL & don't wanna lose oldy so redirect)

Still MITM possible

301: Moved permanently.

GET http://google.com →

← 301 redirect to https://google.com

preload of main domain ✓ / all subdomains taken care of

Solution: HTTP Strict Transport security (HSTS)

1st time user connects → STS header / directive

max age: → Always connect using https for next max age time

includeSubDomains → not only for main domain but also for subDomains

STS valid till private data is not wiped out (Browser's)

Sequence: First req HTTPS also called TFU (Trust of First Use)

301 → preload chromium project google owns

HSTS server / hsts preload.org

TLS

get index.html

With website

3 ways

(1) Preload list

(TFU) (2) client ↔ server

(3) Dynamic HSTS

(Based on server implementation) (4) 301/302 redirect

Key

checks for HSTS header

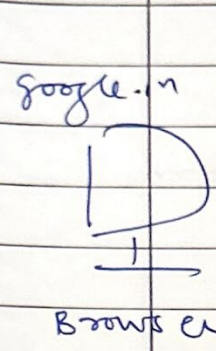
Includes in preload of next chromium updates

Not Browser won't apply HTTPS by default

Using HTTP makes sense from testing of that browser what it can support

NUS

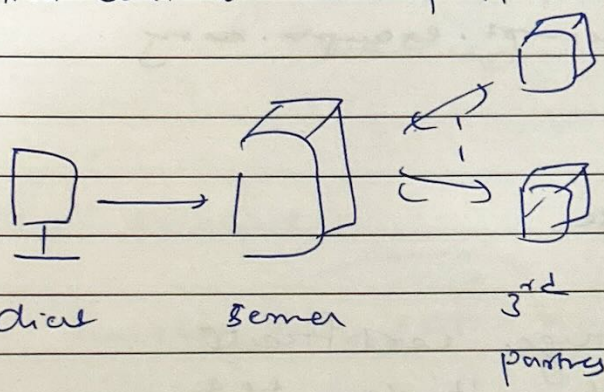
__/__/____



other are made by some consulting 3rd party website for Analytics cookies etc — only one we want

on average 24 TCP connection are established when you connect

(9) Mixed content : HTTP & HTTPS



Insecure loading of content } Insecure, CSRF } Attacks

Mixed content

Active

eg: JS
Stylesheet
iframe

Passive
Audio, video
eg: Images etc

older browser allowed options to open such content over https but newer doesn't.

we can still link with legacy website

CSP : content ~~protection~~ security policy

→ can't manually handwrite https for every http

CSP directive : upgrade-insecure-request

(2) Use relative URL's

(3) script-src 'self' → content only from site own origin

img-src * ;
media-src example.org example.net
script-src userscript.example.org

(3) Rogue certificate

→ MITM using Rouge certificate
→ Two Factor auth. can't solve this
→ Really big issue

(CAA)

Dynamic HTTP (PKP)

① public-key pinning (HPKP)

(assuming we have earlier communicated)

- Browser stores digest of certificate & compare with what it received correctly.

problem

① self DOS Attack
i.e. genuinely the certificate is expired

② DNS CA Authorization Records

→ Authoritative NS
We store CAA entry/Record which tells who can issue my certificate
(i.e. on my domain (main) & subdomain)

③ Certificate Transparency (CT)

__/__/____

→ Create log of pre-certificate

Append only DS: Merkle tree (hash) (easy to verify)
(like Block Chain kind mechanism)

SCT → sign certificate & signed Timestamp
Timestamp

SCT Extension
↳ Log ID, SCT → how many logger (log server) logged for
(in EV atleast 3 ~~format~~)

→ Browser can implement policy that if not logged in atleast 2 loggers show warnings (google requirement) (CA logger)
↳ 2 log server (google Trust Log server)

Who does it?

Monitor → regular scan for all ~~scan~~ logger for any changes

Even browser can follow the same thing (patterns)
(eg: Brave suit) learned knowledge & run proof what's happening

__/__/____

WiFi security : Threats & solutions

WiFi Arch. : WLAN controller : For managing access points, conf can be pushed to all the AP.

BSS → Basic service set / cell

Why WiFi security? : more than 50% of Intern Traffic is carried by wifi

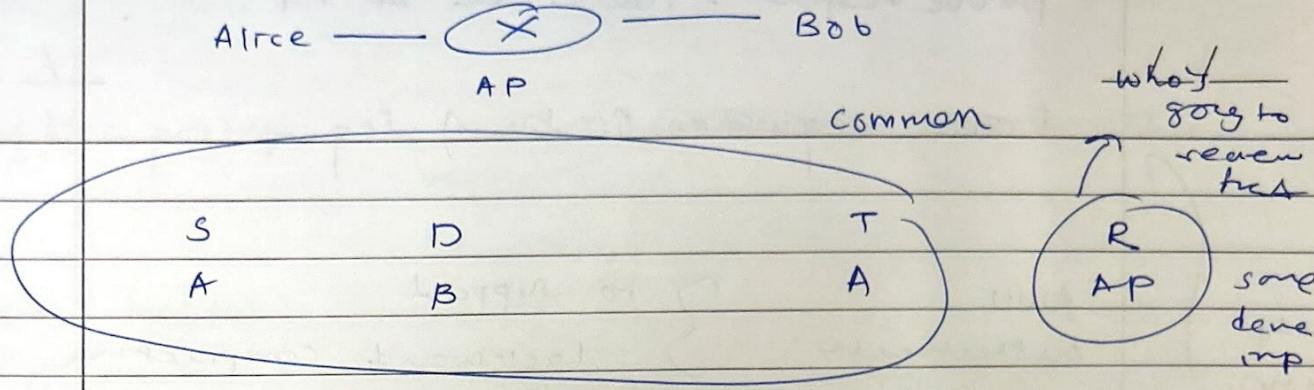
WiFi / wireless ethernet

802.3 ethernet
802.11 WiFi → IT wireless version

So, IT already resembles with frame of ethernet

Access Point : Two NIC → one for WiFi & other for switch

Duration = Air time = 100 B (payload) / 100 Mbps



so we can travel over Distributed system hop by hop.

Seq control: To Trace whether we get Ack or not / used expect Ack

802.11s mesh

Ad-hoc Mesh / mode

QoS control : prioritize traffic

High Throughput (HT) control : (10 MB) As a single packet (eg) send as Jumbo / to packets

https encryption: End to End (Application layer encr)
Wifi encryption : hop to hop (MAC layer encryption)

① → Wifi - Secure has nothing to do with what's above that
② → Security
③ → HTTP
④ → public place examples

probe respon → Identifier of AP

_/_/_

Frame ~~sequence~~ control sequen

until now no IP header just layer 2 header

Null
Authentication } to support backward compatibility

Association Message } similar to ethernet

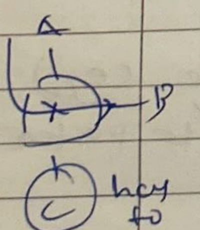
→ can be achieved by monitor mode of wifi

(Broad. Beacon frame — every AP typically shows presence (100 ms)

PR → Broadcast
PResponse → unicast

(costly) Active scanning (✓)

(delay) Passive scanning



how to reject.

costly operation ↓ keeps on sending probe request even tho we don't want to connect

↓ not sending probe request

Sol^n → don't do aggressive probe request

promiscuous mode ↓ also show packets that are not meant

Can do logon to hotels in wired scenario

__/__/____

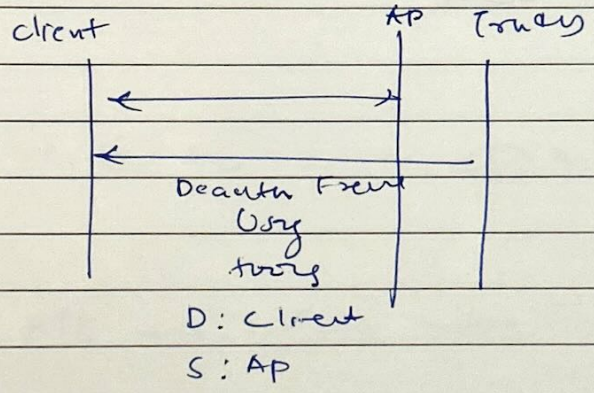
captive portal → to capture user details, ~~mobile~~ mobile number

digital hygine

DOS

① Frequency Jammy

② DDassociation/Deauthentication message



③ Evil Twin → Only client Authentication

① WEP → $C = m \oplus k$

② WPA $c \oplus k$

③ WPA2 plain text $\leftarrow m \oplus k \oplus k$

④ WPA3

Issues

① no key management

How peripherals should exchange

② Users share same keys

Same as hotspot/guest wifi

Anyone who has password can decrypt things → for encryption & authentication

WLC → wireless LAN controller

AS: LDAP / Active directory

CAPWAP (UDP)
control & proxy of wireless access part

__/__/____

WPA:2 Authentication & key management

← same →

WEPX

psk = PBKDF2 (SSID, passphrase, MAC STA)
↑ different

traffic

1. control
2. user
3. Management

Attacks on WPA2

passphrase
- → client authentication
- → keys → encryption

KRACK → key reinstallation Attacks on WPA2
↳ forces the client to reuse the nonce

__/__/__

mitrally
if
(basically)

$n_1 \rightarrow k_1 \rightarrow m_1 \rightarrow c_1$

$c_1 = k_1 \oplus m_1$

Attacker from

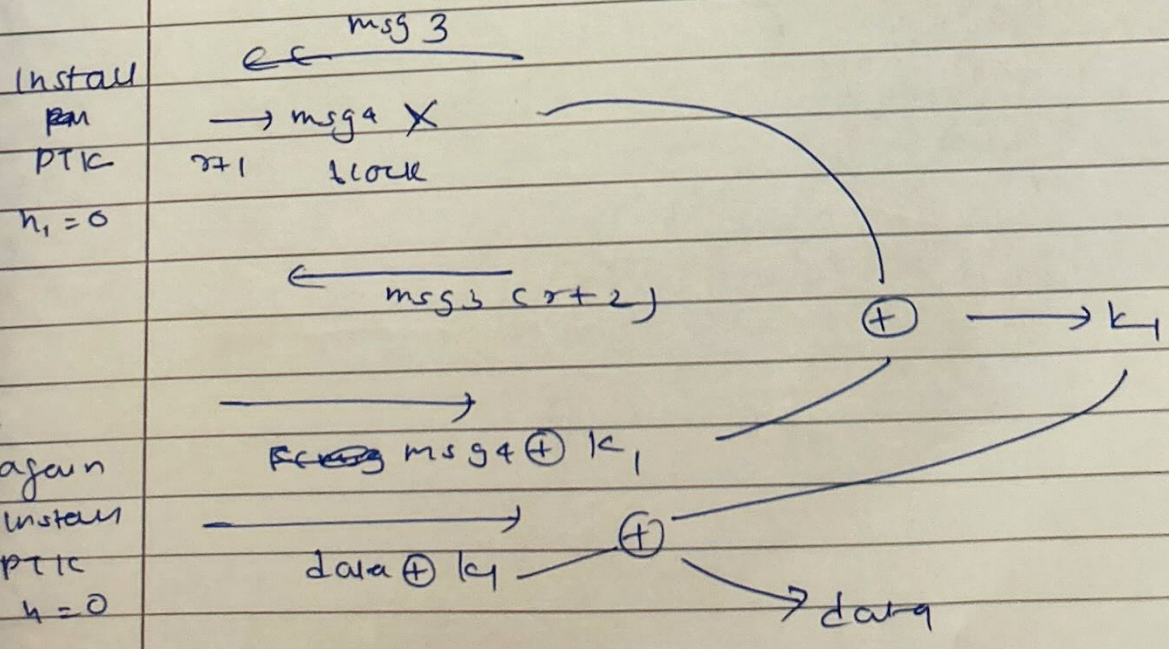
$n_1 \rightarrow k_1 \rightarrow m_2 \rightarrow c_2$

$c_2 = k_1 \oplus m_2$

$c_1 \oplus c_2 = m_1 \oplus m_2$

Instally PTK → initiate nonce to zero

MITM attack on 4-way h/s



__/__/____

① Multi-channel MITM (force deauth)
② SSLstrip
③ KRACK

preconfig → internally used

another name — EC-DHKE

WPA ③

① Personal ⟶ SAE to protect against offline dic. Attack
② Enterprise
③ Enhanced open ⟶ Encryption without Auth.

Enter.

supports
Management Frame Protection (MFP)

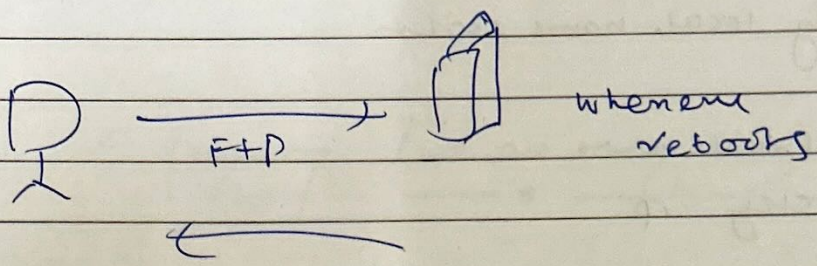
DNS Basic
security
P censorship

__/__/__

DNSSEC

analogy: phone to no. lookup

- Initially, centralized rep.
  SRI (Stanford research Institute)



store in
/etc/hosts in Unix

problem:

Not scalable

- Uniqueness of name (hard to enforce)

soln

DNS in hierarchy

DNS: depth upto 128
typically 3 to 4

zone → collection of name space } under adm. of single entity (partially)

Same name __/__/___
diff location
replicated

(A-M)

13 root servers → ICANN (6 are [illegible] owned

TLD → company

Why local name servers

① Distributed
② Biology ip

Why puiz

→ If local fails
→ To collect & analyse click info

If some one host website → Auth. NS
↓
ip (Name → ip)

→ Caching Tradeoff
↳ Caching happens for all hierarchy of NS

(local)

→ DNS will repurpose for it has cached

Recursive:

- load on Root NS (burden on entire hierarchy)

Iterative

- no Forwarding of query (we are using this) —— * ——

DNSSEC we sign

RRset → multiple entry for single domain

New resource record for DNSSEC

RRSIG → Signature over RRset

DNSKEY → public key

DS → Delegation signer → building chain of Trust

DNSKEY

Domain, TTL, Type, Type of key, Protocol, algo, key id.

__/__/____

RRSIG

Domain, TTL, Type, Type of record signed, Algo, no. of labels, Digest, TTL,

Sig exp date + time, Signature Begin date + time, key tag, signers name, sign

Delegation signer → hash of key

Domain, TTL, Type, key tag, Algo, Digest type, hash

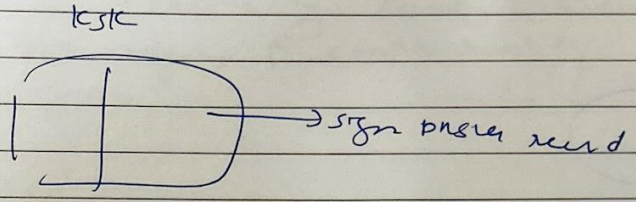
NSEC record → authenticated non-availability of data

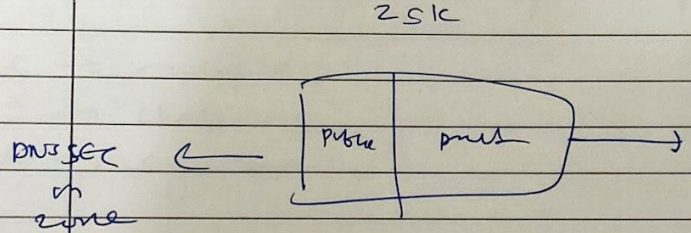
→ Alph order
→ if not found then give next record.

problem : NSEC walk

privacy issues

NSEC3 → hash it





key rollover method
→ pre published
→ double signature

Does DNSSEC solves all our problems ?

→

→ DNS reflection

→ small packet, big response problem (amplification)

TLD mostly
SLD 1-2%

DNSSEC → half baked mature
→ not widely deployed.

1/3 of DNSSEC can be validated

__/__/__

No ~~encryp~~ Crypto encry in DNSSEC
as anyone can read the data as it has a public key
authentication / check trust of ~~R~~ server.

IPSEC

VPN
IPSEC
Benefits
TEC mode
Archi
key exchange Modern
Components.

VPN — secure tunnel over public network

~~IP~~ IPsec : layer 3 (secure @ encry level) Then "

→ not a single protocol

→ Group of protocol / framework
flexibility to user
What to use.

AH → only Authenticat (Integrity protection)

ESP → payload protection

Why IPsec?

no talk

IP is not secure → not in begining

Int. for secure entity (Internal they) so
no security Meter later Public reason (bad actors)

Possible security issues

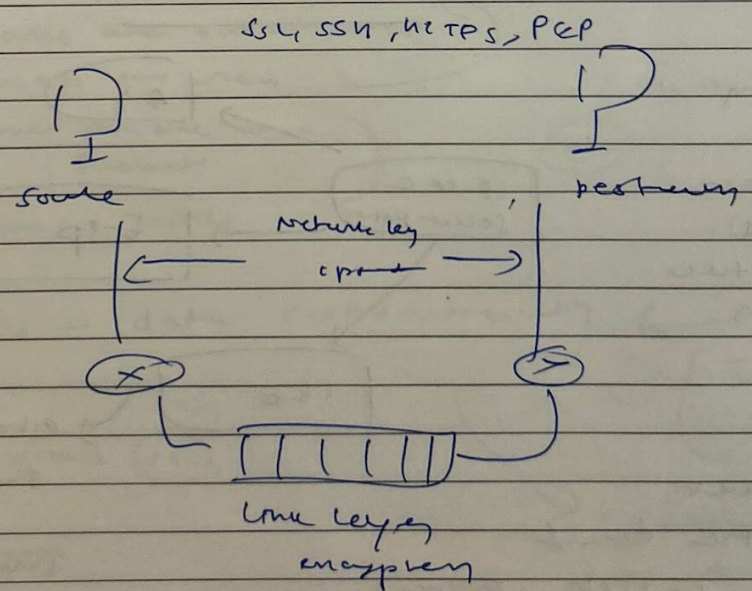
- source spoofing : Packet with our IP address
- replay attack (recording same request)
- No data integrity & confidentiality.

Benefits

VPN works transp. (like your app works)

C → encryption
Σ → checksum
A → signature

Diffent layers of encryption

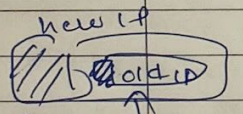


IPSec : Mode / Feature

Tunnel

Transport Mode

IPv4
IPv6
mode



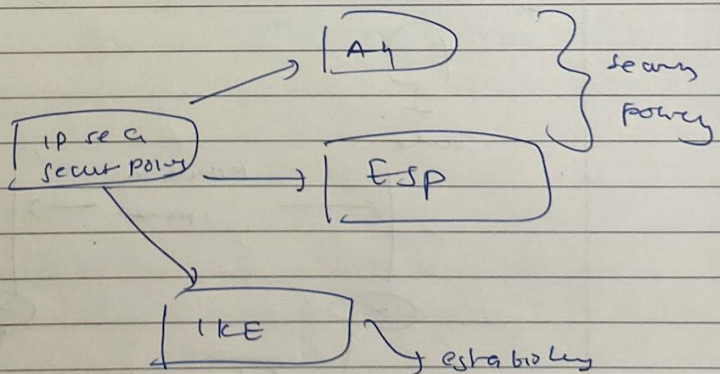
new IP Packet

same IP packet
add
additional info
req for IPSec
(IPSec header)

new IP header

as Payload

IPSec Architecture



→ protect entire header

→ used when VPN between two companies

→ establishing the tunnel key management

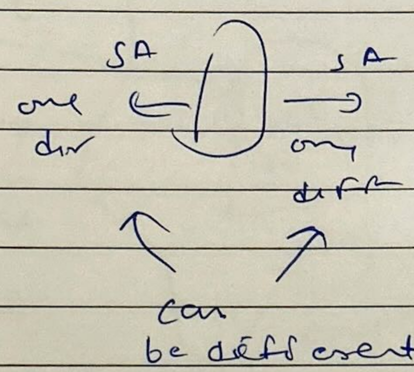
Internet traffic doesn't come to HTTP, Policy that does the same.

only weak security traffic.

for two or more VPN clients.

__/__/__

(SA) → defines security mechanism defined on a session



How to setup?

(1) manually (you may the VPN client)

(2) VPN client

only TCP ahead ds

useful if we have ASAP? security measures

hashing of entire packet

(ISAKMP)

(AH) : headers are authenticated
: protects only when (when someone receives packet)
: mutable fields

IP protocol (51)

ifunct

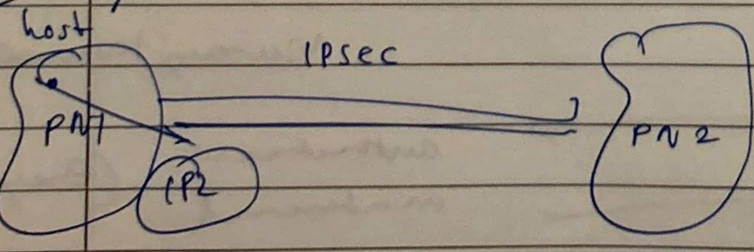
(ESP) : AH + data confidentiality

(Two company analogy)

IP protocol (50)

high end VPN server

payload (IP1)



Routed through internet

__/__/____

IKE — establishing IPsec session

Aggressive mode, main mode, Quick mode

→ limited no. of messages

→ All messages

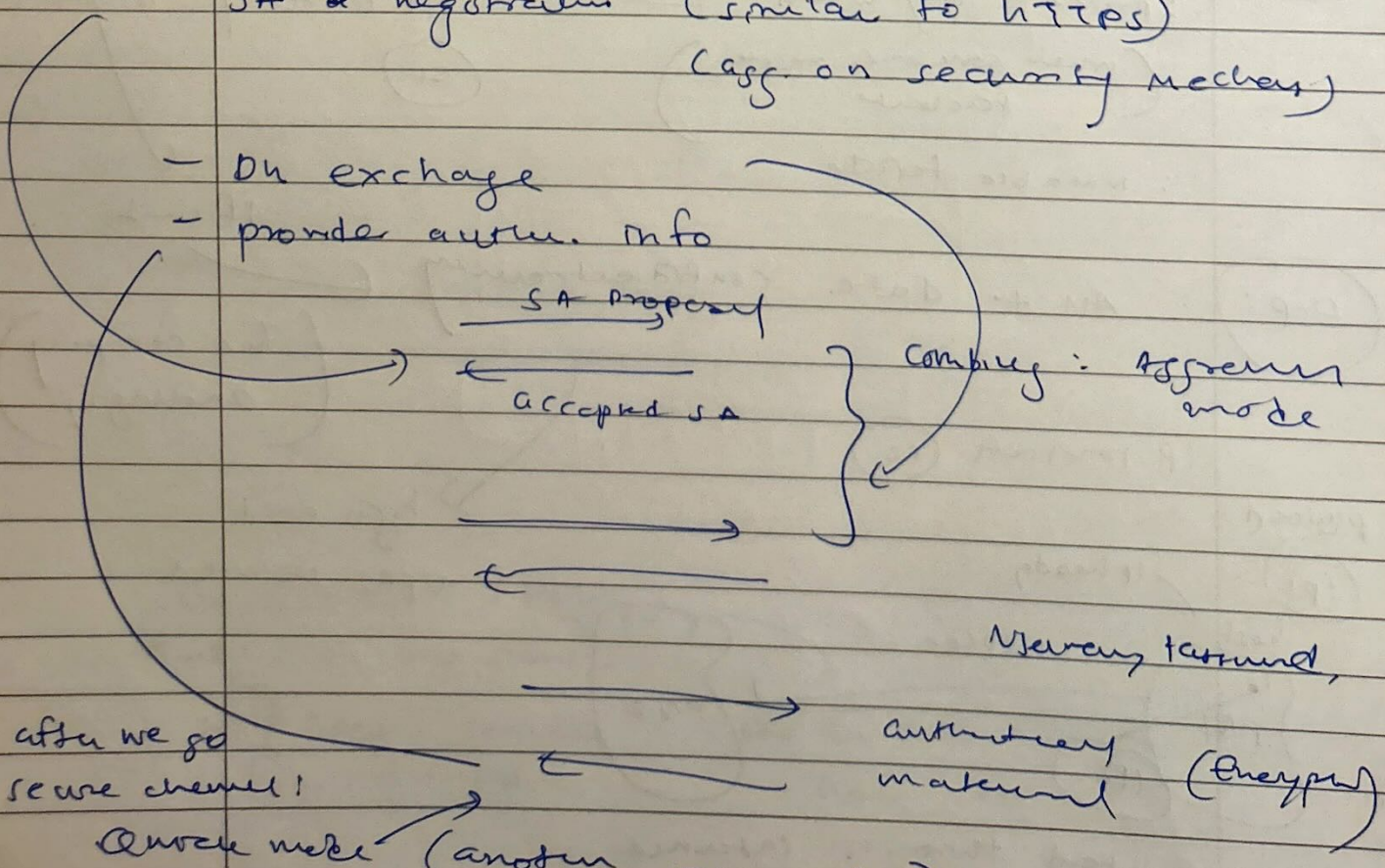
3 auth mode

① preshared key
② public key encryption
③ public key signature (certificates)

IKE — phase 1

— SA & negotiation (similar to HTTPS)
(agg. on security mechanism)

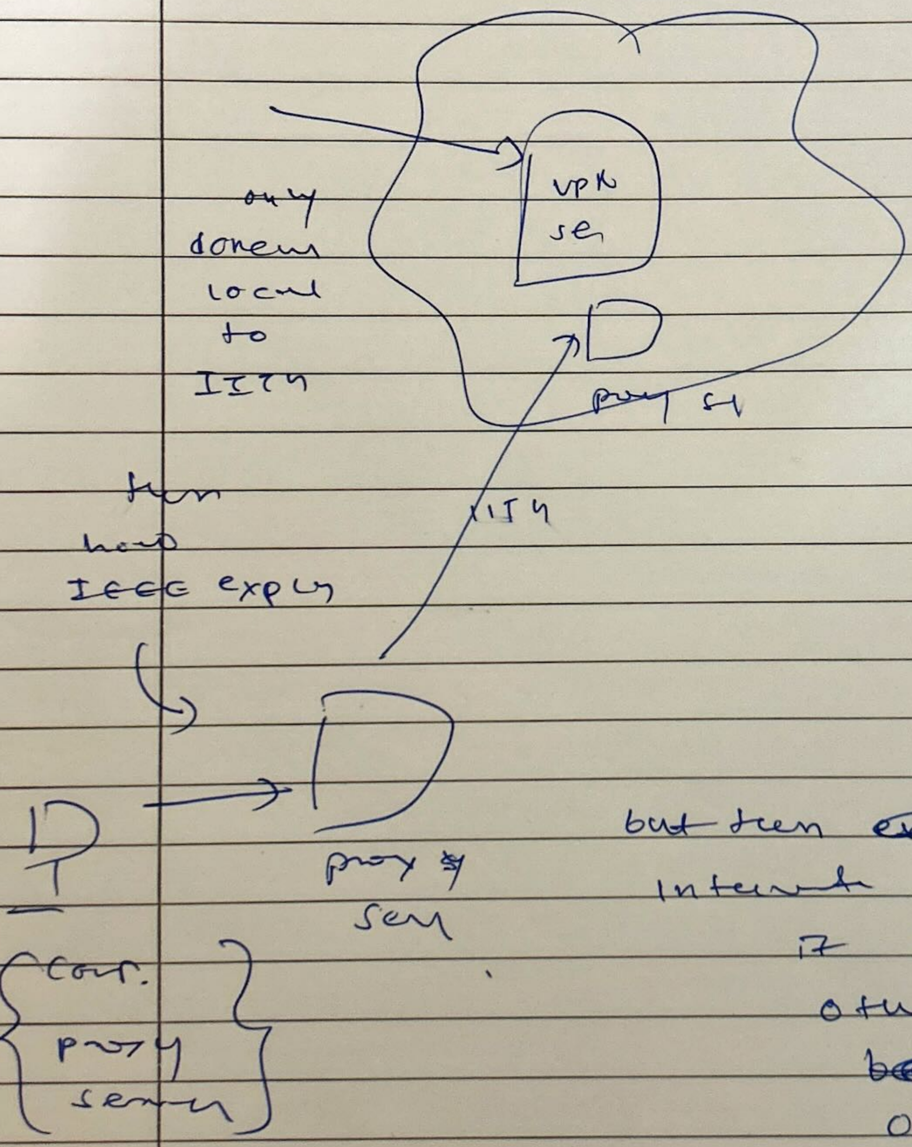
— DH exchange
— provide auth. info



→ less no. of messages to agree on auth + ~~protocol~~ S.M

Quick mode

→ (Auth. Method, SA proposal) →



but then ~~every~~ all ~~ip then~~ internet ~~traffic~~ goes to it but then any other website will ~~be~~ ~~to~~ have latency or ~~log~~ lag in recieving their response so we will filter or not use it, or we can also filter @ proxy of IITH but still it will hit ~~proxy~~ + ~~to~~ use resources of IITH