SSL Handshake Protocol

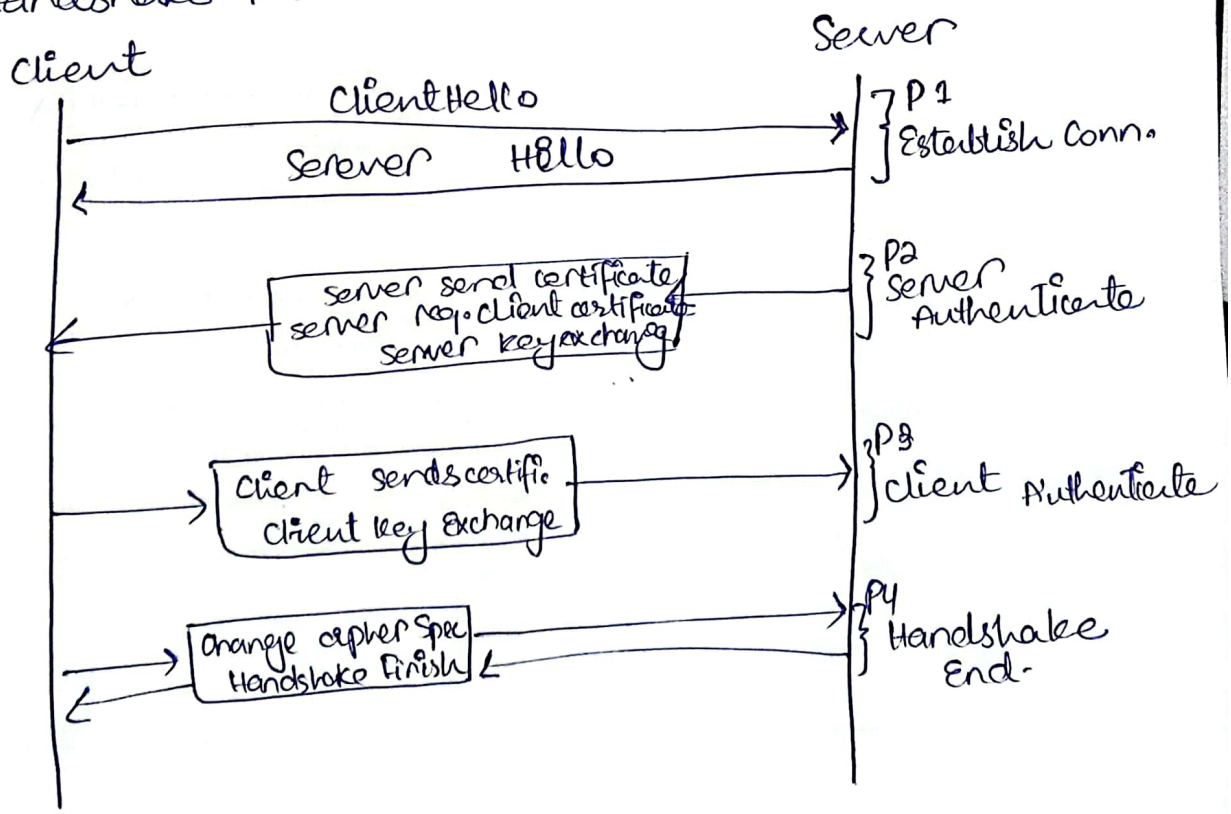
Handshake Protocol is used to establish session This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

Phase -I: In phase-I both client and server send hello-packets to each other. In this qp session, cipher suite and protocol version are exchanged for security purposes.

Phase -II: Server sends his certificate and Server-key -exchange. The server end phase-2 by sending the Server-hello-end packet.

Phase -III: In this phase, client replies to the server by sending his certificate and client-exchange-key.

Phase -IV: Change-cipher suite occuss and after this the Handshake Protocol ends.

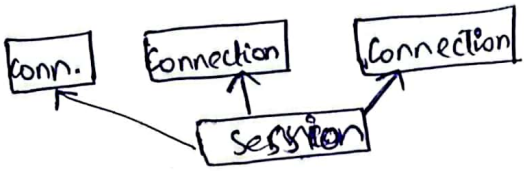


SSL Session

→ Session ID
→ Master secret key (48 byte)
→ Cipher sec
  ↳ data encryption algo (DES, IDEA).
  ↳ hash Function
  ↳ hash size
→ peer certificate (An X.509)
→ compression Method
→ Is resumable
  → Whether the session can be used to initiate new connection.

SSL Connection

→ Server and client random.
→ Server write MAC secret
  • The secret key used by MAC send by the server
→ Client write MAC secret
→ Server write key
  • Encryption key for data encrypted by the server & decrypted by the client.
→ Client write key
→ Initialization vector
→ Seq number



Web security Considerations:

① **Web is very visible:** Website on internet can be easily seen by anyone, increasing the risk of unauthorized access or attacks.

② **Web server are easy to configure and manage** Setting up the web server is straight forward, but it can lead to mistake or oversights that hacker can exploit.

③ **Complex software hide security flaws** Web application rely on intricate software, which can have hidden vulnerabilities that attackers can target.

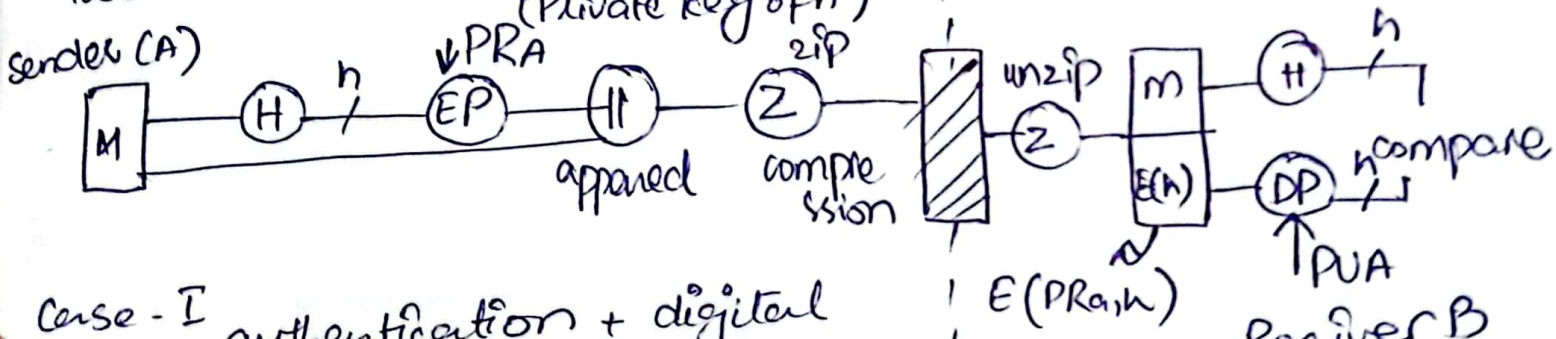
④ **User not aware of risk:** many internet use lack awareness about online dangers, making them more vulnerable to scams & attacks.

Email Security → PGP (Pretty good Privacy).

PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage application - Used for (signing encryp decryp

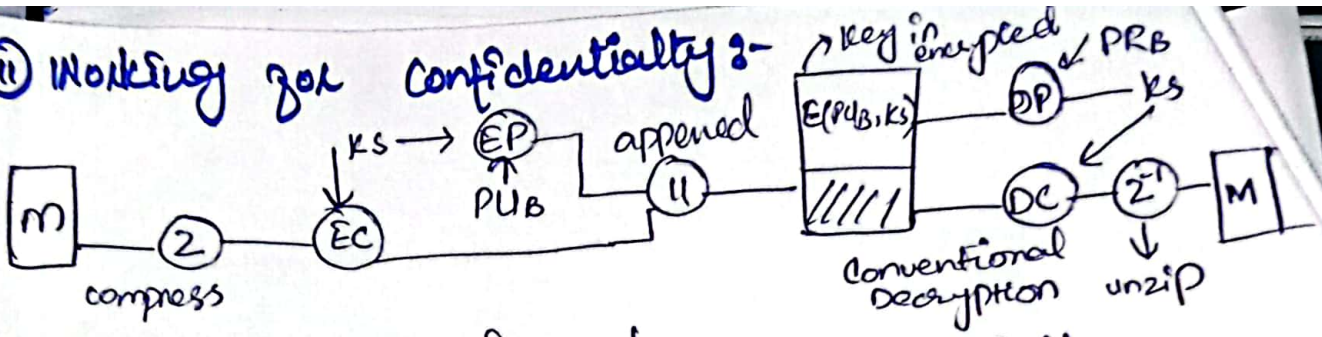
② Working for Authentication :-

① The sender creates a message
② SHA-1 is used to generate a 160-bit hash code of the message
③. Hash code is encrypted with RSA using sender private key, and the result it prepended the message
④ Reciver uses RSA with the sender public key to decrypt and recover the hash code.
⑤ Reciver generate a new hash code for the message and compress it with the decrypted hash code. If the two match, the message is accepted as Authentic.



Case - I authentication + digital signature ko use kar kary.

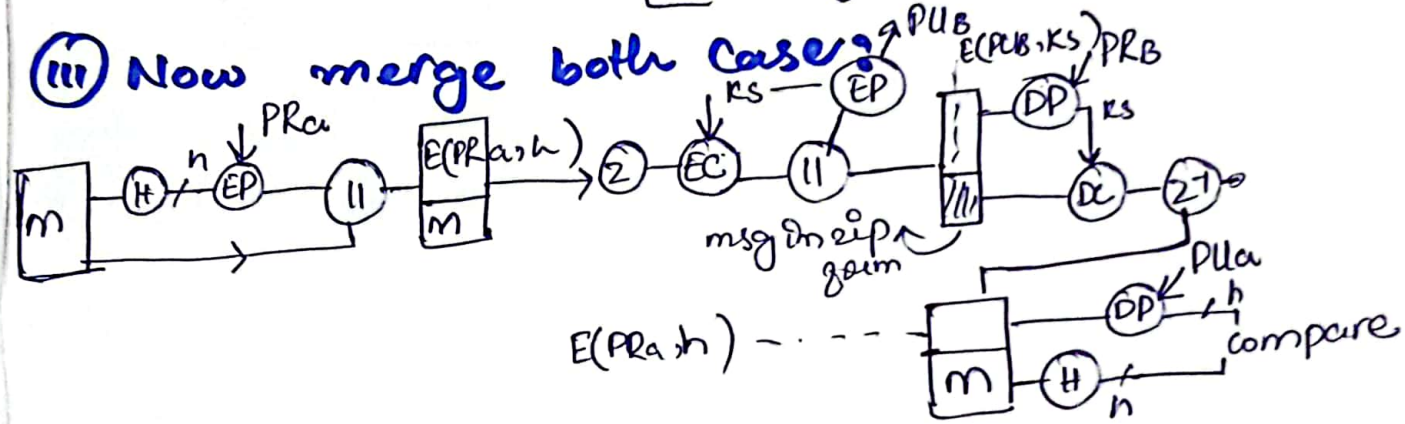
(ii) Working for Confidentiality :-



EC = Convential encryption
use 1 key only

only confidentially
NO signature ; no authentical

(iii) Now merge both case :-



Advantages :-

- Sensitive info always protected.
- It can't be stolen or viewed by other.
- Confidentiality : Protect email content from unaut. access
- Authentication : Verify the identity of sender.
- Integrity : Ensure all email don't alter during transmission
- Trust model : Allows users to ~~fete~~ independly verify the authenticity of public key.
- ~~Interop~~. Interoperability :- Work access different email client & platform.
- Privacy : Ensure privacy by encrypting email content -

7- Trap Door :/ (Back Door)

Defect in the computer code that allows malicious actors to exploit the flaw and gain access to valuable information.

8- Easter Egg:

Hidden code or functionality intentionally embedded within software for humful purpose to gain the sensitive information or to take control of your system.

9- Ransomeware:

Type of malware attack that encrypts a victim's data and prevent access until a ransom payment is made. Ransomware attackers often use social engineering techniques, such as phishing, to gain access to victim environment.

10- Bootkit:

- Modern malware used by a threat actor to attached malicious software to a computer.
- Malware that modify the boot sectors sector of a hard drive, including (MBR) Master boot record and VBR (volume Boot record)

Date: __/__/20

Malware Types

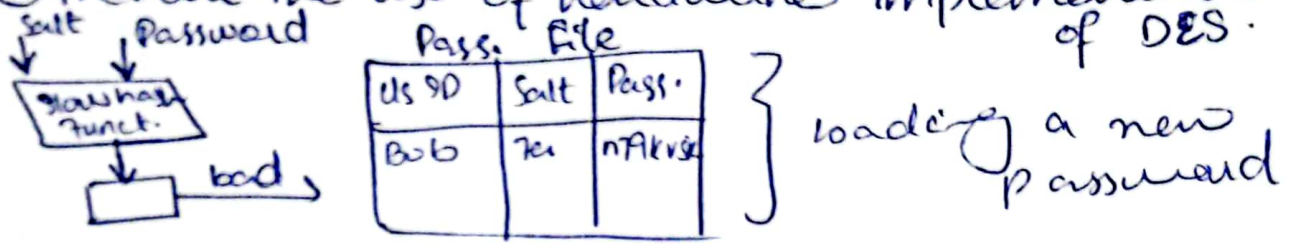
Malicious Software refer to an Software that are designed to damage computer system, steal info, gain unauthorized access

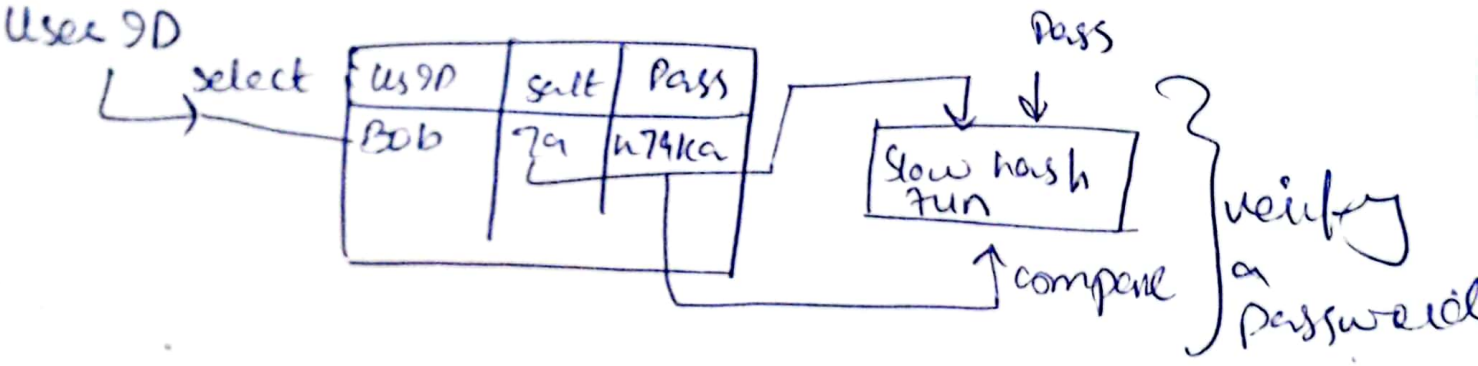
1. **Viruses:** Code that copy itself into other program.
2. **Bacteria:** Replicate until it fills all disk space or CPU cycle
3. **Payload:** Harmful things the malicious program does, after it has time to spread.
4. **Worm:** Program that replicates itself across the network (usually riding an email messages or attached document)
5. **To Trojan Horse:**
   A program download and installed on a computer that appear ~~harmfact~~ harmless, but is, infact malicious. Unexpected changes to computer settings and unusual activity. (eg: store user login info, and send to hacker).
6. **Logic Bomb:**
   Malicious code that activates on an event. Like type of malware that are activate on a specific date or time

Passwords → SALT

→ Purpose :

(i) Prevents duplicate passwords
(ii) Effectively increase the lenght of the password
(iii) Prevent the use of hardware implementation of DES.





(1) User creats a password : "Password123".
(2) Salt generated : "a92fbc7e"
(3) Salt is combined with the password "a92fbc7ePassword123"
(4) Salted password is hashed : d30f1d7c4de980821 & "
(5) Salted password and salt is securely stored together

KERERAS (V4)

- **Authentication Services Exchange: To obtain Ticket-Granting Ticket**

① **C → AS** IDc || IDtgs || TS1

Client sends a message to Authentication Services (AS) with its identity (IDc), the ID of Ticket-Granting Service (IDtgs) and a timestamp (TS1).

② **AS → C** $E_{Kc}[K_{c,tgs} || IDtgs || TS2 || Lifetime2 || Ticket_{tgs}]$

The AS respond to the client with an encrypted message (E) containing the session key b/w the client and TGS ($K_{c,tgs}$), the identity (ID) of the TGS (IDtgs), a new timestamp (TS2). The lifetime of the TGS Ticket (Lifetime 2) and the TGS Ticket ($Ticket_{tgs}$).

- **T-GS Exchange: To obtain Services-Granting Ticket.**

③ **C → TGS** IDv || $Ticket_{tgs}$ || Authenticator$_c$.

The client sends a message to Ticket-Granting Service (TGS) with its identity (IDc), the ID of the desired service (IDv), & an authenticator containing a timestamp.

④ **TGS → C** $E_{Kc}[K_{c,v} || IDV || TS4 || Ticket_v]$.

The TGS responds to the client with an encrypted message (E) containing the session key b/w the client and the request service ($K_{c,v}$), the ID of the service (IDv), a new timestamp (TS4) & the service ticket ($Ticket_v$).

Client / Server Authentication Exchange: To obtain Service.

⑤ **C → V** $Ticket_v$ || $Authenticator_C$

The client sends a message to the requested server (V) with the service ticket ($Ticket_v$) and an authenticator ($Authenticator_C$) containing a timestamp.

⑥ **V → C** $E_{Kc,v}[TS5 + 1]$ The server responds to the client with an encrypted message (E) containing a timestamp (TS5+1), indicating successful authentication and granting access to request service.

IPSec Services:

- Access Control (Enforce security polices & control network access)
- ~~Connectionless Integrity~~
- Authentication → Data origin Authentication / verifies the identities of communicating parties - / source → check.
- Encryption → Confidentaility: Protect data confidentaility by encrypting it -
- Integrity → Connectionless integrity. Ensure data integrity by detecting modification -
- Anti-Replay Protection →

~~Not Rec~~ Prevent attackers from replaying intercepted packets

- Traffic flow confidentiality

Dump + Read → add J.

How to identify an SA usage:

- a one-way relationship b/w sender & receiver

3 parameters:

(i) Destination IP Address

(ii) Security Protocol : AH or ESP.

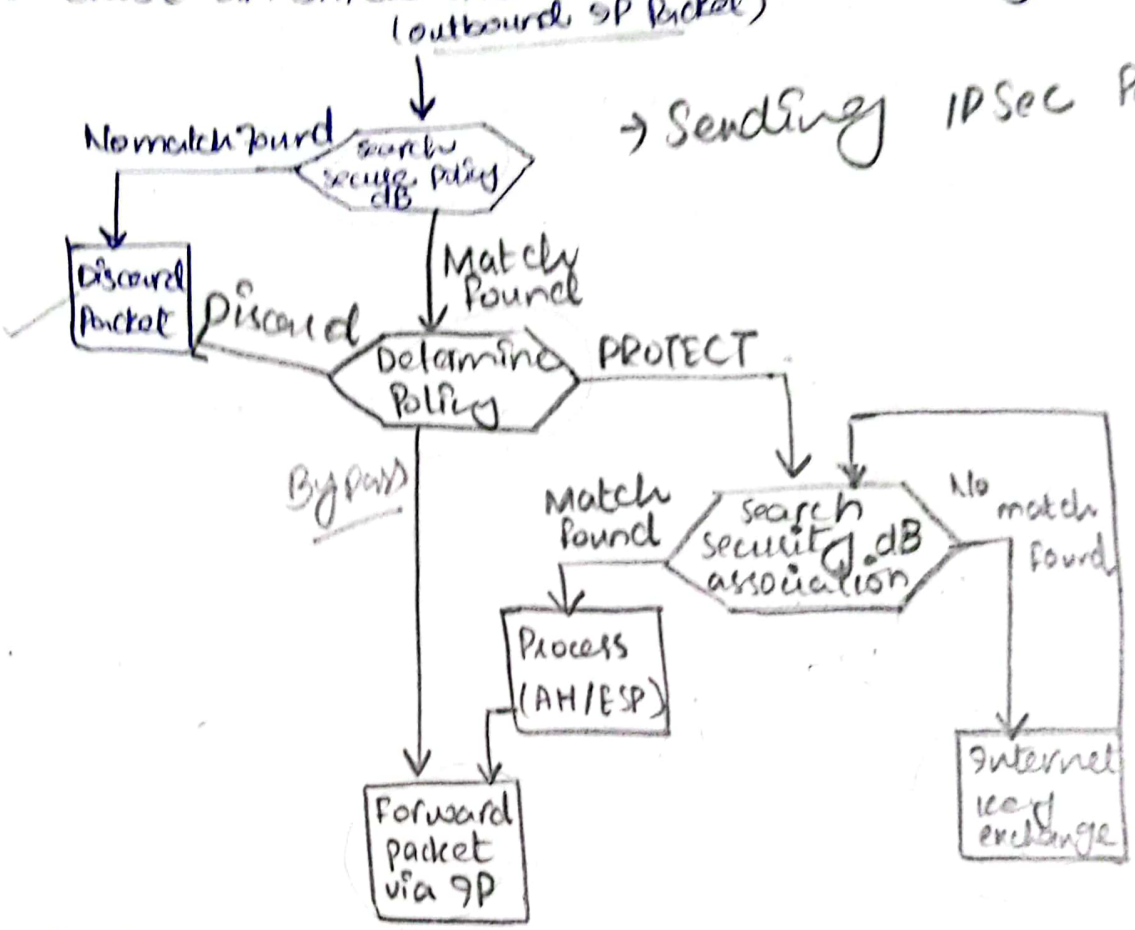
(iii) Security Parameter Index (SPI)

- A local 32-bit identifier (to be carried later to endpoints with AH & ESP.

Play

OutBound Traffic Flows

- Compare the selector fields of SPD with the one in the IP traffic.
- Determine the SA, if any
- If there exist an SA, do the AH or ESP processing.



→ Sending IPSec Packet

Inbound Processing

→ Check the incoming IPSec packet & process with AH or ESP.
→ Discard in case of an anomaly.