INFORMATION SECURITY

* Attack Types → Active
→ passive → e.g. packet sniffing

* SQL injection attacks

Characteristics of Information:

1) Confidentiality
2) Integrity
3) Availability

→ DDOS → distributed
→ MITM : man in the middle.
hijack, delete, forge data in transit.
→ IP spoofing
→ Packet sniffing.

* Attack vectors ← number of ways to get into a system.

* Attack surface mapping :

list of all assets → draw architecture diagram → label components & interactions → identify vectors → Rate vectors.

* Threat modeling

- S poofing identity
- T empering data
- R epudiation
- I nformation disclosure
- D enial of service
- E scalation of privileges.

Security Design Principles.

(1) Least privilege

→ bare minimum permissions
→ temporary permissions should be taken away.

(2) Separation of privilege.

→ multiple conditions for one access.
e.g OTP.

(3) Fail Safe defaults

→ default settings should be conservative
→ whitelisting
→ Blacklisting ← should not be used.

(4) Complete mediation

→ Each & every access is subject to security checks.

* Backdraw is that performance goes down.

⑤ Open design.

→ Security mechanisms should NOT be secret
→ use standard algorithms, controls.

⑥ Economy of mechanism

→ aim for simple design.

⑦ Psychological Acceptability

→ security mechanisms should not sacrifice usability

⑧ Least common mechanism

→ minimize shared resources b/w applications.
→ use isolation (software based)

Security Policies → Standards.

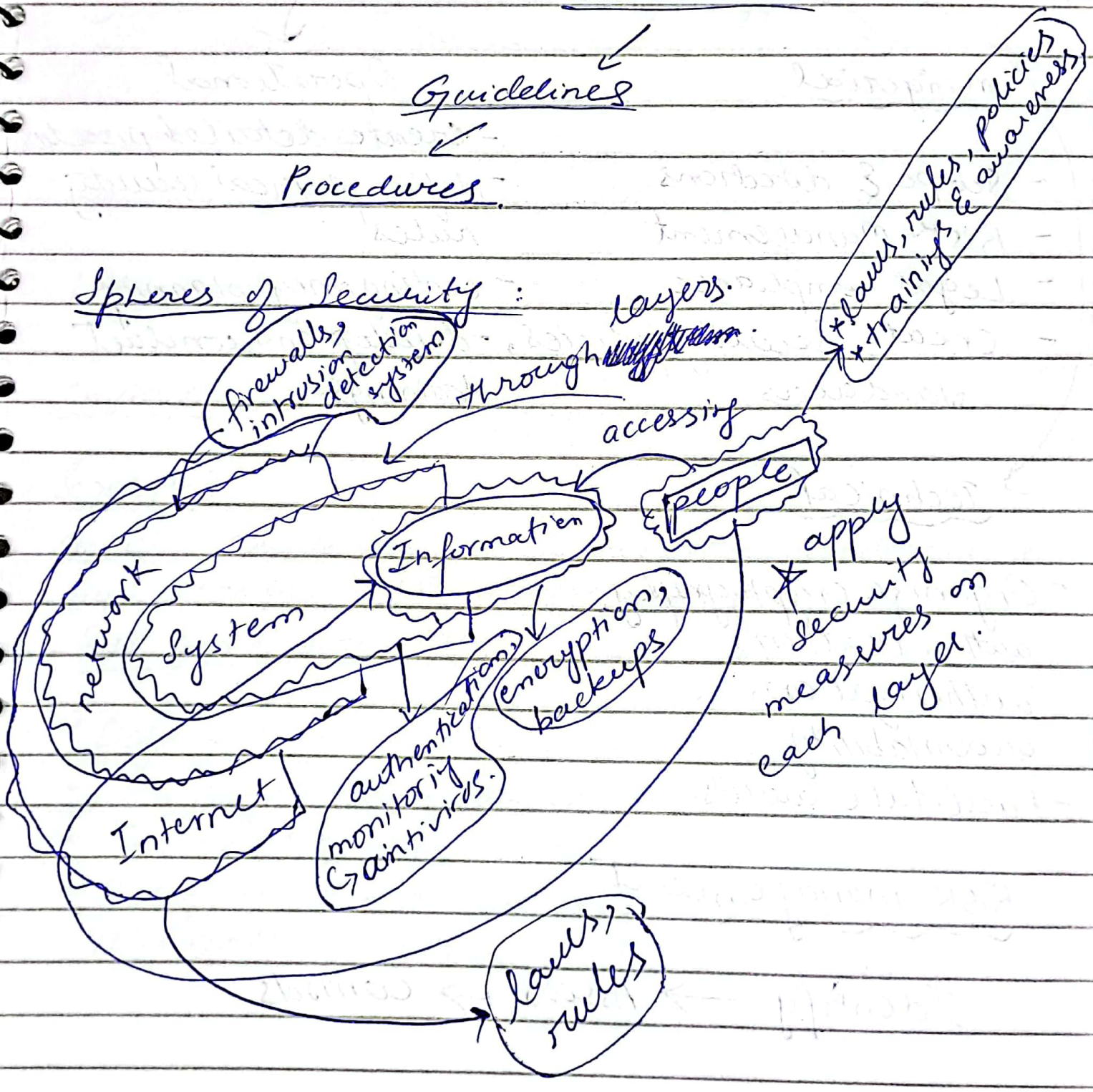
↓

Guidelines

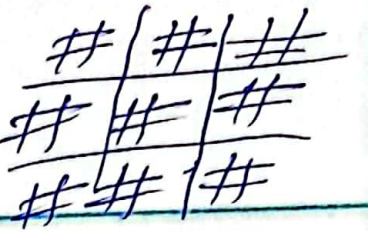
↓

Procedures.

Spheres of Security :



I found a hard game for us to play its called supertic tac toe looks like this



Controls

Managerial

- Scope & directions.
- Risk Management
- Legal compliance
- Create Security policies, standards

Operational

- Create detailed procedures
- define physical security rules
- contingency planning
- develop and conduct trainings.

Technical.

- Organize cryptography authentication, authorization, accountability
- Facilitate audits.

Risk management

Identify → Assess → Controls

{Assessment}

Risk: exploiting vulnerability in application server

SQLi on DB server
DDos on Web server
Ransomware attack on org

Likelihood: scale (1-10).
low, high
Impact: scale (1-10).

Score A·B

Control: Pen testing, code inspection, ~~Redundant~~ web server, data backs. DRP.

Risk control strategies

- Defense - apply safeguards
- Transference - outsource insurance
- Termination - avoid risky business activity
- Acceptance - do nothing
- Mitigation - planning for response.

Contingency planning

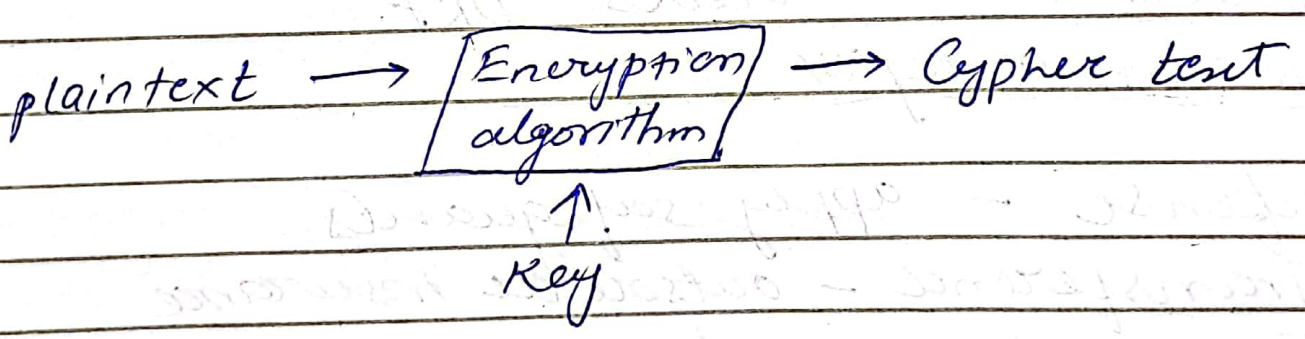
Incident Response plan

- Identify & classify incident indicators e.g unknown process running use of dormant account.
- Contain the incident.
- Restore services.
- Document all details.

Disaster Recovery plan

- Disruption is expected.

Intro to cryptography:



Symmetric Encryption: when encription & decription key is the same. Also called conventional cryptography.

Two primary techniques:

1) **Substitution**: it creates "confusion"
2) **Permutation**: it creates "diffusion"

Ancient Substitution Ciphers:

→ Replace plaintext characters according to a translation table
→ Substitution rules become secret key

Example: * **Caesar Cipher**

Size of key space = |alphabet count| - 1
for english alphabets = 25 : 26 - 1

* **Monoalphabetic substitution**: no need to map alphabets in their actual sequence. randomize the cipher alphabets.

Cryptanalysis techniques: code cracking.

→ **Frequency Analysis**: compute frequency of cypher text letters & compare with

Typical language frequencies.

* could be computed in combos e.g unigrams, bi-grams, trigrams. you may discover repeating words & compare them with same frequency words in typical language.

Polyalphabetic Substitution:

* Vigenère Cipher: uses caesar's cypher but uses multiple shifts.

* Simple columnar transposition: write in ~~columns~~ rows, read in columns using key sequence.

Exercise:

Cipher text = EBRYTSLTIEAY

Key = Batool
2 1 6 4 5 3

num of characters = 12
num of key chars = 6
num of rows = $\frac{12}{6} = 2$

2 1 6 4 5 3
R E A L I T
Y B Y T E S

2 1 ~~6~~ 4 5 3
~~R E~~
~~E B~~
~~I R~~
~~E Y~~
A T
Y S

Pros:

* symmetric crypto is very fast to compute
* hardware acceleration is available in many cases

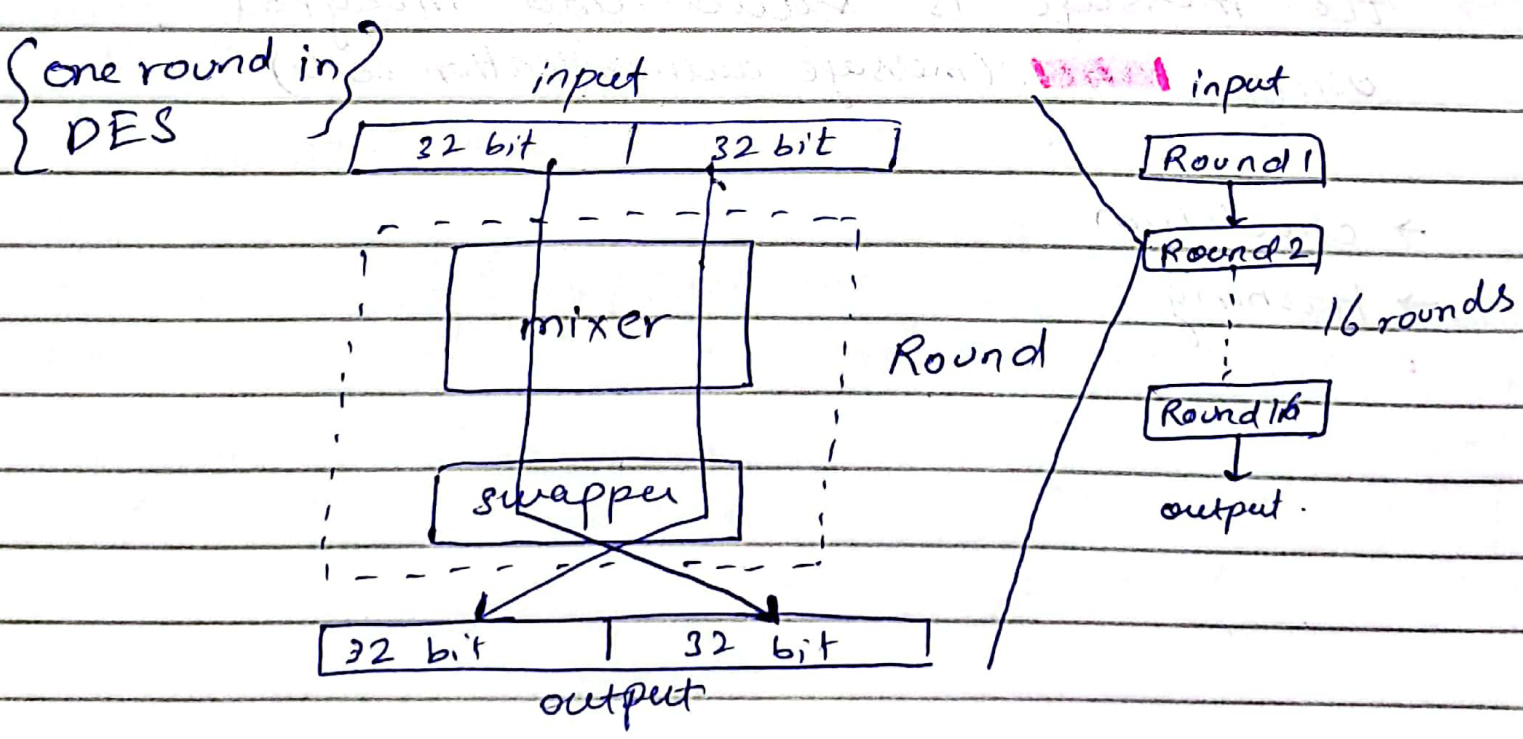
Cons:

* no way of sharing key

Modern cryptograph methods / algos

→ Data Encryption Standard:

* **DES** uses 64 bit plaintext block & 56 bit key to produce a 64 bit cipher text block
* permutation based.



→ Triple DES (3DES)

* repeats basic DES algo 3 times
* using two or three unique keys
  - key size of 112 or 168 bits

Stream Ciphers

* random generators

* To fulfill integrity check there is a message authentication code attached to every message, when the same algo is run over the message on receiver end, he can verify if the message is valid and integral using MAC. (message authentication code)

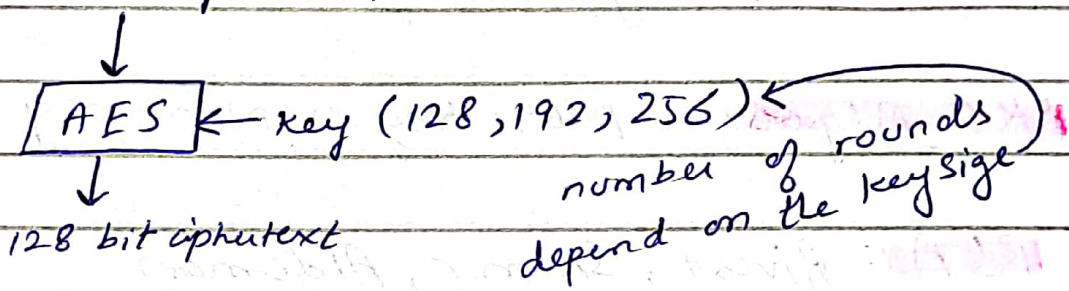
→ checksum

→ hashing

:

Advanced Encryption Standard (AES):

- → Iterative
- → processes 128 bits of plain text
- → operates entire data block in each round



Pocket AES example by hand:

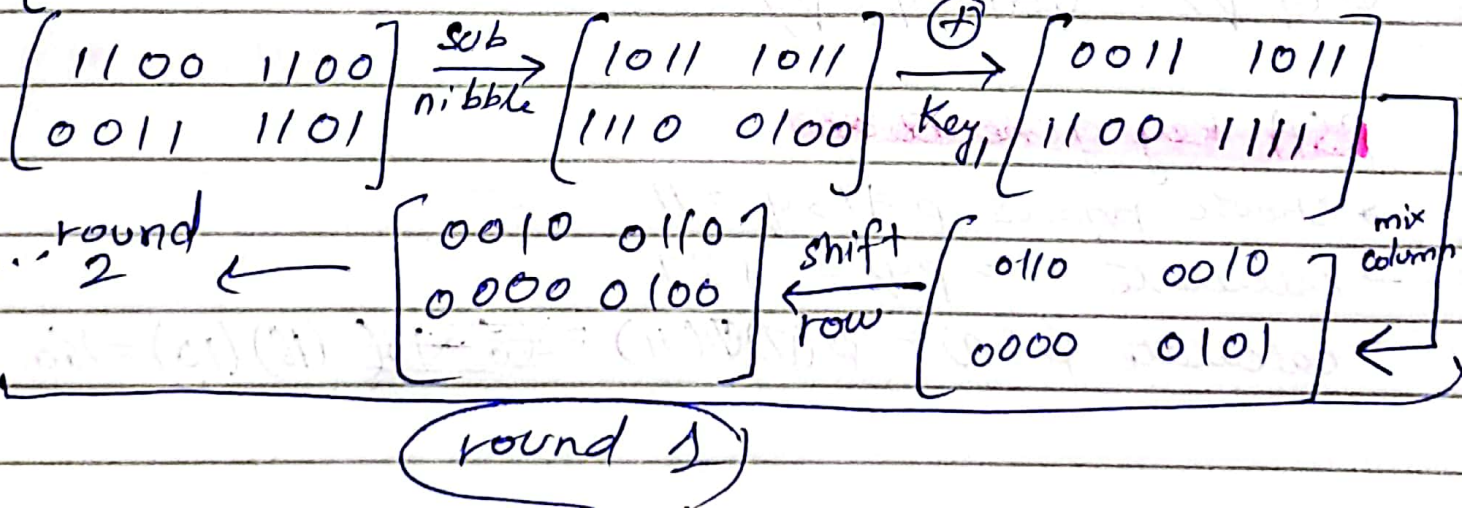
Let plaintext = C3CD , Key = 1A2B

State ← split plain text into nibbles

↳ subnibbles → new state → AddRoundKey(state, Key

(check diagram in assignment) ..... shift row ← mix columns ← state ←

[for hand solved, check notes of sir]



→ Data Encryption — PKC ⇒ public key cryptography

→ Message Authentication

↳ MAC : message authentication code

* MAC generated using one's private key is called Digital Signature.

PKC Algos : public key cryptography

RSA : Rivest, Shamir, Aldeman

* Co-prime numbers : two numbers are coprime if their GCD is 1.

* Euler's Totient function :

$\phi(n)$ = count of all coprimes from 1 to n

$\phi(\text{prime number } \{n\}) = n - 1$ (property)

$\phi(pq) = \phi(p) \times \phi(q)$

RSA Key generation :

→ choose primes $p = 17$, $q = 11$

→ calculate $n = pq = 187$

→ calculate $\phi(n) = \phi(17)\phi(11) = (16)(10) = 160$

→ Choose e such that $\gcd(e, \phi) = 1$

$$1 < e < \phi$$

let $e = 7$

→ Find another number d such that

$$(d \times e) \bmod \phi = 1$$

$$7d \bmod 160 = 1$$

$160K + 1 = 161, 321, 401, \ldots$

$\frac{161}{7} = 23 \Rightarrow d$

public key : (e, n) private key : (d, n)

Encryption:

message $= m$

$$c = m^e \bmod n$$

↑ ciphertext

Decryption:

$$m = c^d \bmod n$$

↑ plaintext/ message

+ factorization of huge numbers is not possible/feasible

numerical example for RSA:

* Let $p = 7$, $q = 13$

  $n = p \times q = 7 \times 13 = 91$

* Select $e = 5$ since is satisfies

  $1 < e < (p-1)(q-1)$

* $d = e^{-1} \bmod (p-1)(q-1) = 29$

  $5^{-1} \bmod (6)(12) = 29$

* public key $= (n, e) \Rightarrow (91, 5)$

  private key $= (n, d) \Rightarrow (91, 29)$

* Let plaintext $m = 10$

  ciphertext $(C) = m^e \bmod n$

  $= 10^5 \bmod 91 \Rightarrow 82$

  plaintext $(m) = c^d \bmod n$

  $= 82^{29} \bmod 91 \Rightarrow 10$

Advantages of RSA

* No key sharing needed
* proof of owner's authenticity
* data can't be modified in transit
* fast encryption

Diffie Hellman Key exchange Algorithm

→ helps exchanging key safely

Steps

(1) * Choose a prime number q
* Select α as a primitive root of q

To be primitive root,

$$\left.\begin{array}{l} \alpha \bmod q \\ \alpha^2 \bmod q \\ \alpha^3 \bmod q \\ \vdots \\ \alpha^{q-1} \bmod q \end{array}\right\} < q \quad (1, 2, 3, \cdots, q-1)$$

(2) user 1

private key = X_a where $X_a < q$

public key = Y_a where $Y_a = \alpha^{X_a} \bmod q$

Key pair = $\{X_a, Y_a\}$

user 2

private key = X_b where $X_b < q$

public key = Y_b where $Y_b = \alpha^{X_b} \bmod q$

key pair = $\{X_b, Y_b\}$

(3) user 1 has X_a, X_b, q, secret key generated:

$K = (Y_b)^{X_a} \bmod q$ similarly user 2 has

X_b, Y_a, q, secret key generated:

$$K = (Y_a)^{X_b} \bmod q$$

where

$$(Y_b)^{X_a} \bmod q = (Y_a)^{X_b} \bmod q$$

numerical example of Deffie Hellman

* Let $q = 17$, $d = 3$ as

$$\left.\begin{aligned} 3 \bmod 17 &= 3 \\ 3^2 \bmod 17 &= 9 \\ 3^3 \bmod 17 &= 10 \\ &\vdots \\ 3^{16} \bmod 17 &= 1 \end{aligned}\right\} < 17$$

* **user 1:**
  $X_a = 15$ (private)
  $Y_a = 3^{15} \bmod 17 = 6$ (public)

user 2:
$X_b = 13$ (private)
$Y_b = 3^{13} \bmod 17 = 12$

* user 1 key generate $= K = 12^{15} \bmod 17 = 10$
  user 2 key generate $= K = 6^{13} \bmod 17 = 10$

→ Diffie Hellman is **vulnerable to "man in the middle attack"** MITM

Public - Key Certificate (PKC)

→ assigned by a 3rd party
→ A Certificate authority (CA) issues a digital certificate which is a piece of data that helps in authentication of original user.
→ This certificate includes:
① Who issued the certificate
② Whom was it issued to
③ public key of owner
④ validity period
⑤ digital signature by CA

* X.509 : International standard for the format of public key certificate

Public - Key Infrastructure (PKI)

* set of hardware, software and procedures needed to create, store or revoke, distribute digital certificates

{Public, Private} → CA → Digital certificate

involves ; CA, RA, CRL

Online Certificate Status protocol (OCSP)

standard to verify if certificate is valid.

Questions for mid 1

* causes of insufficient security (case scenario where you'll have to find causes)
* what steps are involved in hacking attack of other attacks e.g packet spoofing etc.
* Case scenarios in characteristics of information
* Explain goals of security (CIA)
* What is the benefit of STRIDE / how do you use STRIDE / what are the different ways your data can be compromised?
* case scenarios and identify which security design principles are being compromised.
* case scenario, create a standard for a policy.
* How can you calculate the probability of a risk (risk assesment)
* case scenario, what's the best measure for a scenario from risk control strategies. learn examples for each strategy.
* Encrypt/decrypt text with a classical cypher e.g Caesar's, Vigenére
* DES input/output & key size (MCQs)
* Do DES expansion on given data (32-bit) e.g 3A02 4 7BD
* Do DES substitution on data

(remember the input/output & key sizes of all algos)

* What is the disadvantage of CBC mode?
* Discuss/Compare the block cipher modes.
* Encrypt with stream cypher. $C = m \oplus$ random stream.
* could ask to make diagrams like for MAC hash.
* Calculations for RSA. (cram formulas)

After Mid - 1.

Malware:

- Program that impacts CIA

Malware can:

delete, corrupt, steal files, create zombies, cause anoyance in general.

3 common types of Malware

- Virus → replicates & spreads, host needed
- worm → self-replicating, no host, an independent program
- Trojan → apparantly useful program with malicious code.

* A **virus** needs to attach to host file and then self-replicates (stays in a computer)
* A **worm** is a program that doesn't need a host & self replicates & spreads to other computers itself in network.

* spyware
* ransomware
* scareware
* Keyloggers
* Backdoor

→ behaviours of malware.

Computer Antivirus Research Org. CARO

Naming convention: There's no standard to name virus.

Worm . Win32 / Taterf . K / dll

Type . Target . Family . extra info.

Virus Phases / Structure.

replication code	payload

} **Structure**

payload → goes into a host

Phases

Dormant → (some condition) Triggered → Spreding → execution of the payload

- Triggered ↑ loaded in memory
- Spreding ↑ execute replication code.
- execution of the payload → cause harm.

Targets
- compiled files (exe)
- multipartied → has many targets.
- Interpreted < scripts / macros in docs

Memory Residence

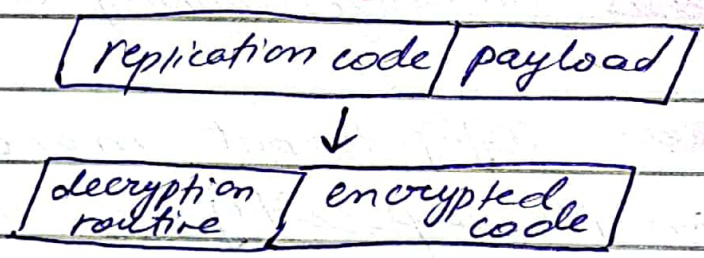
- Resident (lives in memory)
- Temporary Resident (stays in memory but unloads)
- Non-Resident (doesn't live in memory for long time)
  ↑ infects files on hard disk.

~~Malware~~ Virus

Obfuscation ← trying to hide itself.

- None : signature can be developed by antivirus orgs.
- Encrypted : the code is encrypted to avoid signature development and a decryption routine is executed on fly.
- Oligo morphic : choose for a # of decryption routines
- polymorphic : ~~choose for~~ create new decryption routine on runtime.
- Metamorphic : mutate virus body as well.
  uses a mutation engine

* Signature : how a virus look like (the code)
  ↳ bit/byte pattern.



Simple

better

* Stealth : tries to hide itself.
* Armoring : uses files packers, copies itself to multiple parts of file
* Tunneling : uses OS interrupts
* Retro Virus : hinders the operation of antiviruses or protection measures

* overwriting malware : overwrites files
* prepending malware: prepends itself to a file exe
* appending malware : appends itself to file exe
* Cavity malware: when a malware tampers with the exe file, the size may change; to tackle this malware creates cavity i.e empty body of code which takes up space.
* It exploits the cavities in an already existing exe file.
* Packer : The original file is passed through a packer routine & a new .exe file is formed along with the virus init.

Malware Analysis

* Static Analysis : analyze dead code
* Dynamic Analysis : analyze running code.

Buffer Overflow: trying to put more data in buffer than its capacity

Outcomes: ① program crash
② program behaving unexpectedly
③ attacker gaining control.

To exploit:

- Identify vulnerabilities
  - Code inspection
  - debugging
  - fuzzing.
- understand how buffer is stored in mem
- craft special inputs.

defenses:

Compile time

- choose high level language
- safe coding (plan for unexpected inputs)
- safe libraries
- compiler flags for stack guarding
- system programming (in C/C++)

Run Time

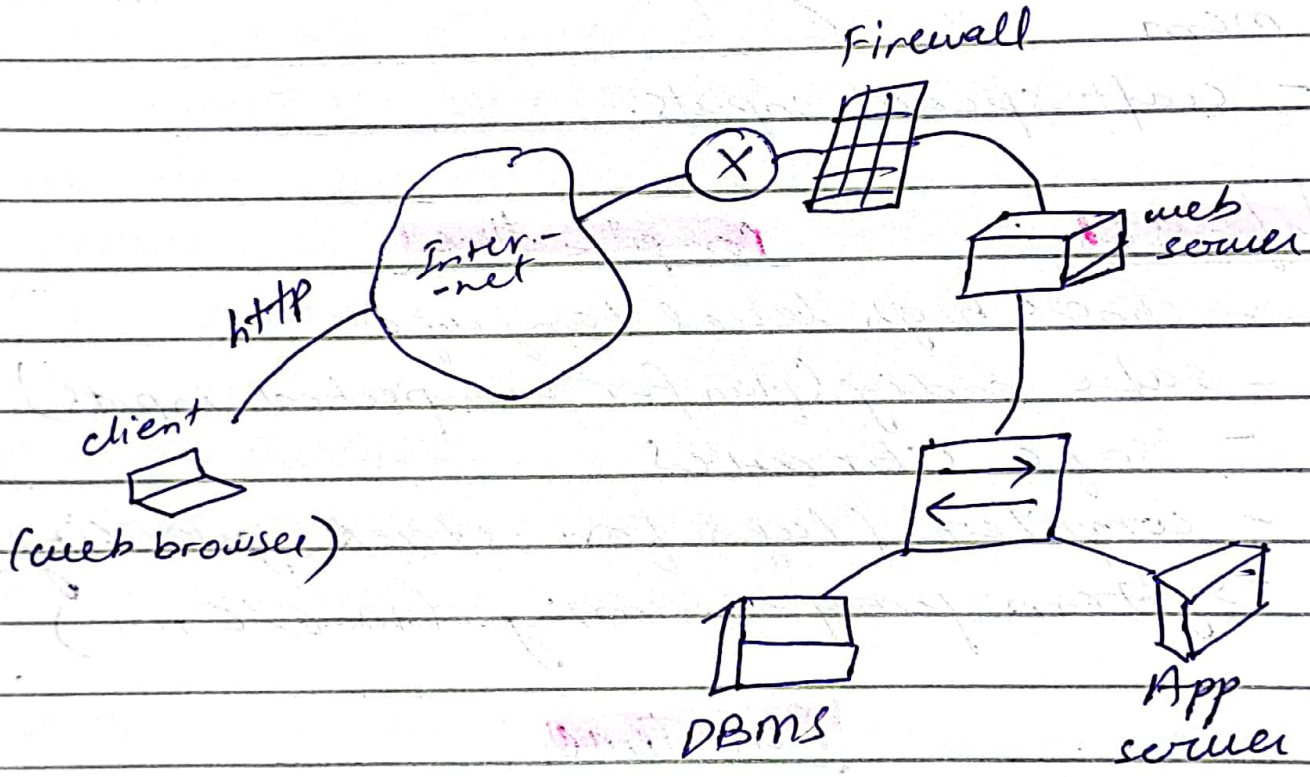
- Non executable address space
- Randomize address space.
- Guard pages → not accessible.

Integer Overflow:

Format string attacks.

Database Security.

SQLI : SQL injection.



* Inputs may contain malicious SQL.
* If application server is vulnerable it will forward malicious query to database.

Strategies:

① Premature String termination

② End-of-line comment
③ Piggy backed query
④ Tautology → inserting a condition which is always true to get more data than needed.

{Prevention}

- Sanitizing inputs.
  - Encode all special characters.
- Data validation. (in backend.)
  - Ensure input patterns.
- Monitor queries for injection signatures.

Database access control

privilege | table | user

GRANT SELECT ON Orders TO ibrahim
REVOKE SELECT ON Orders FROM ibrahim.

* restrict the access.

Role-based access control:

* based on a user's role.

① assign role to user:

GRANT developer TO ibrahim
REVOKE developer FROM ibrahim.

② define privilege for role.

GRANT INSERT, UPDATE, DELETE, SELECT
ON Orders TO developer;

* A clause "WITH GRANT OPTION" can be appended in role granting queries ~~and it is called "casta"~~ which would permit a certain user to GRANT access to other users.

Inference.

* attacker can potentially find out the sensitive data using info insensitive data.
e.g Here's a table of employee info

Name	Position	Salary

Either of these individual columns can be accessed separately by if Name and Salary is queried together

that in confidential. So, an attacker can make 2 separate queries, one for name & one for salary, then combine & form the actual Name, Salary violating confidentiality.

Counter measures.

① **Restructure Tables.**
→ Keep sensitive & insensitive data separate

② **Query time detection.**

Statistical Databases.

→ **Aggregate functions.**
sum, count, min, max, avg, median etc

Applications: Reports, Analytics.

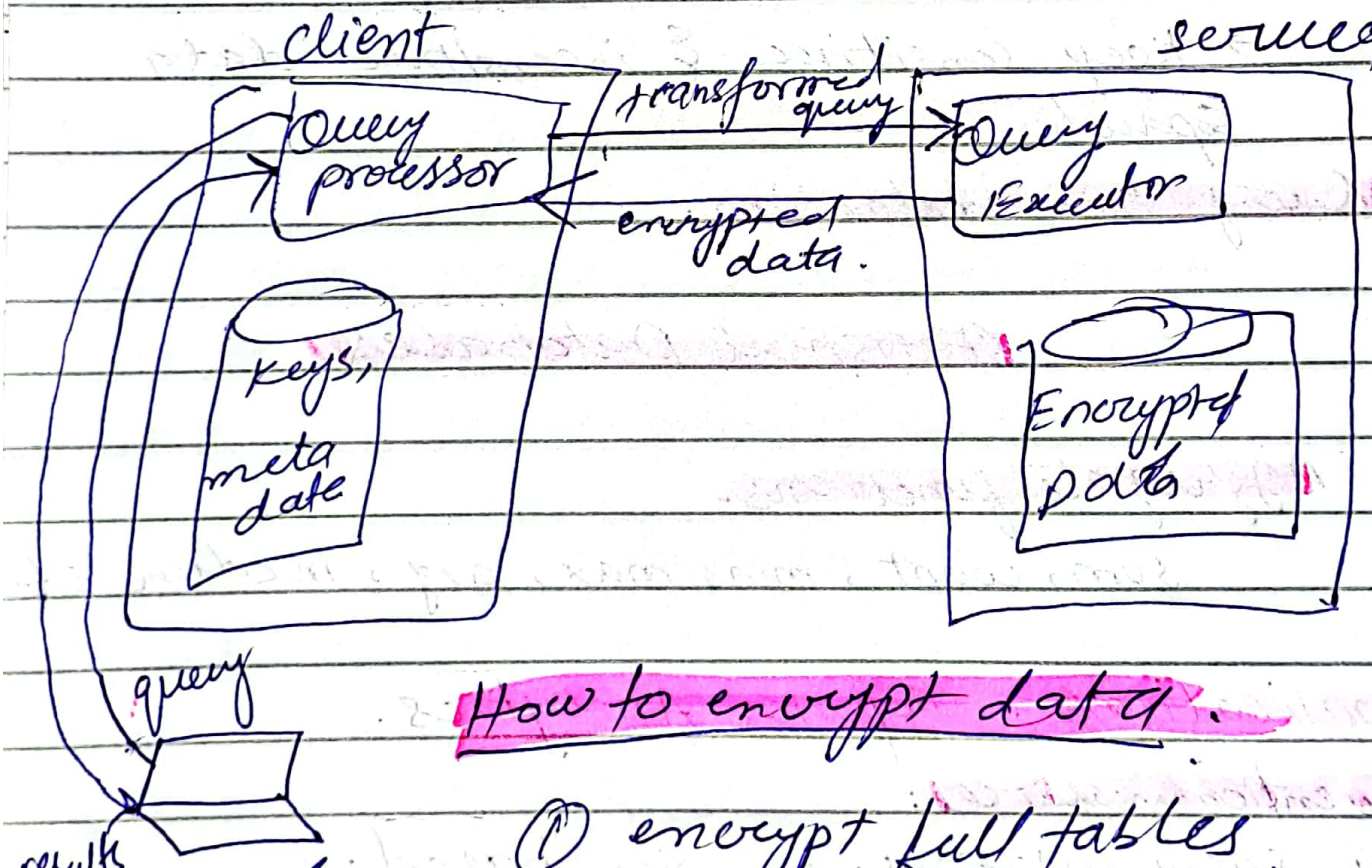
Prevention measure:
① Query restrictions. e.g. size based.
② Perturbation: return noisy stats.
e.g data swapping b/w rows
manual adjustments, Random sampling

Database Encryption.

* use only in extreme security circumstances.

Challenges:

- Querying records.
- key distribution.



How to encrypt data.

1. encrypt full tables
2. encrypt individual fields.
3. encrypt records & create indexes.

User Authentication.

- to verify the identity of user
- checking privileges of user - authorization.

means / modes.

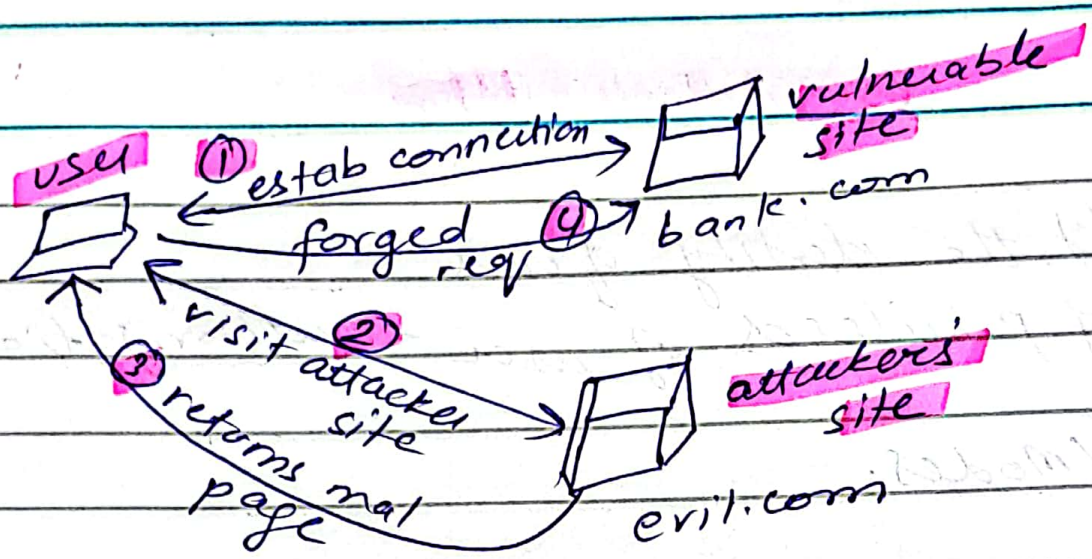
① Knowledge
② Possession
③ Biometrics.

Multi Service Authentication.

Kerberos
TGS : ticket Granting server
authentication server.

Web Security : Cross site request forgery

establish a session between the user & vulnerable site. attacker can redirect user to another site:



* the forged request from another site using the user will be authenticated due to coming from user. (but actually its from attacker)

Actions

- transfer funds
- place orders
- cast a vote.
- delete data

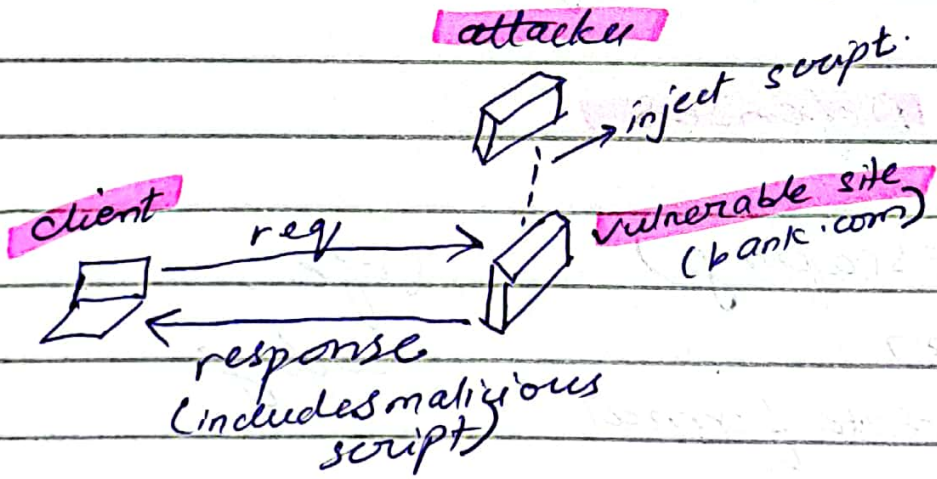
} forgery acts

(CSRF) attacks

* To surpass authentication the correct cookie is used to send requests to the vulnerable website.

* how can server distinguish b/w an authentic & forged request ?
server will generate a token and insert it into the webpage that is being returned in a response. Now, when the user sends that page back, the token should be there, if its not, its an invalid request.
* the token is 1-time used.

Web Security Cross site scripting (XSS)



possible outcomes :

1. stealing cookies
2. Log keystrokes
3. redirect to other malicious/phishing site

① Reflected XSS ← works when the server echoes back the script.

→ create malicious URL
→ send to client & make you click it

② Stored (Persistent) XSS

→ Attacker finds a vulnerability in server
→ send a script as part of input
→ server stores it in db.
→ server includes script in web pages.

③ DOM-based XSS ← document object model → for web doc manipulation

→ Extract some parameter from URL
→ Insert into page (using Dom API)

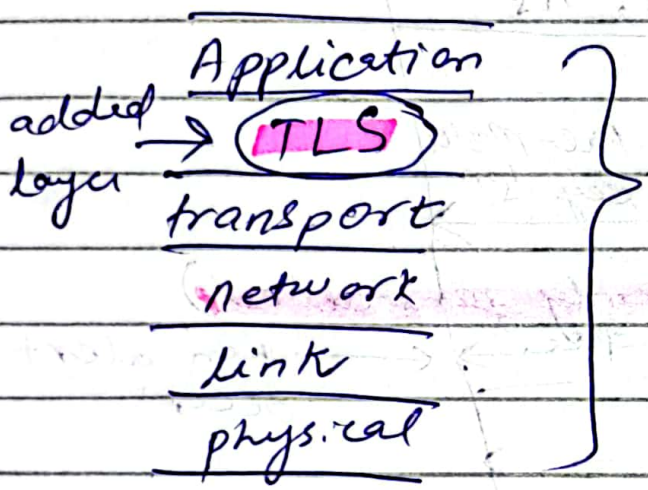
Defenses

① Encoding (escaping) — safe coding
② Validation
→ check data format

Network Security : TLS/SSL

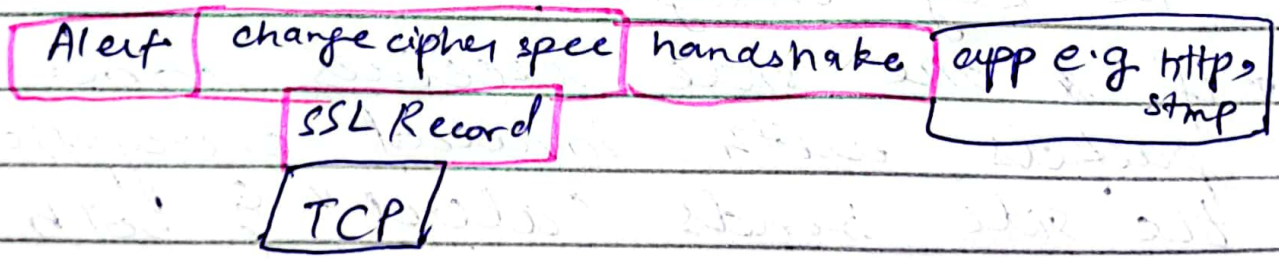
SSL : secure Socket Layer.

TLS : transport Layer security



TLS goals:

- message confidentiality
- message integrity
- Peer authenticity
- Protection against "message replay"

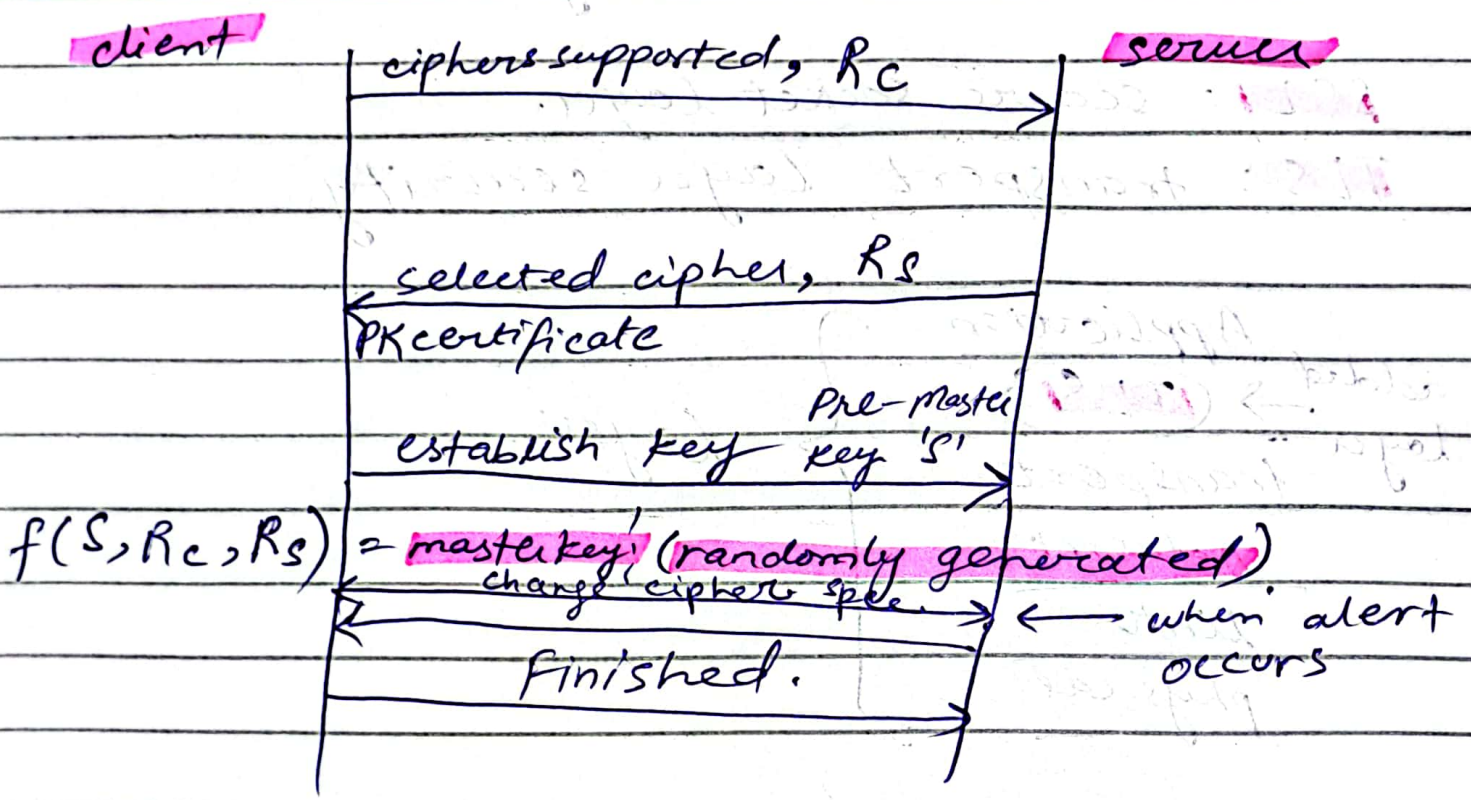


TLS add ons

TCP

set up session ↘

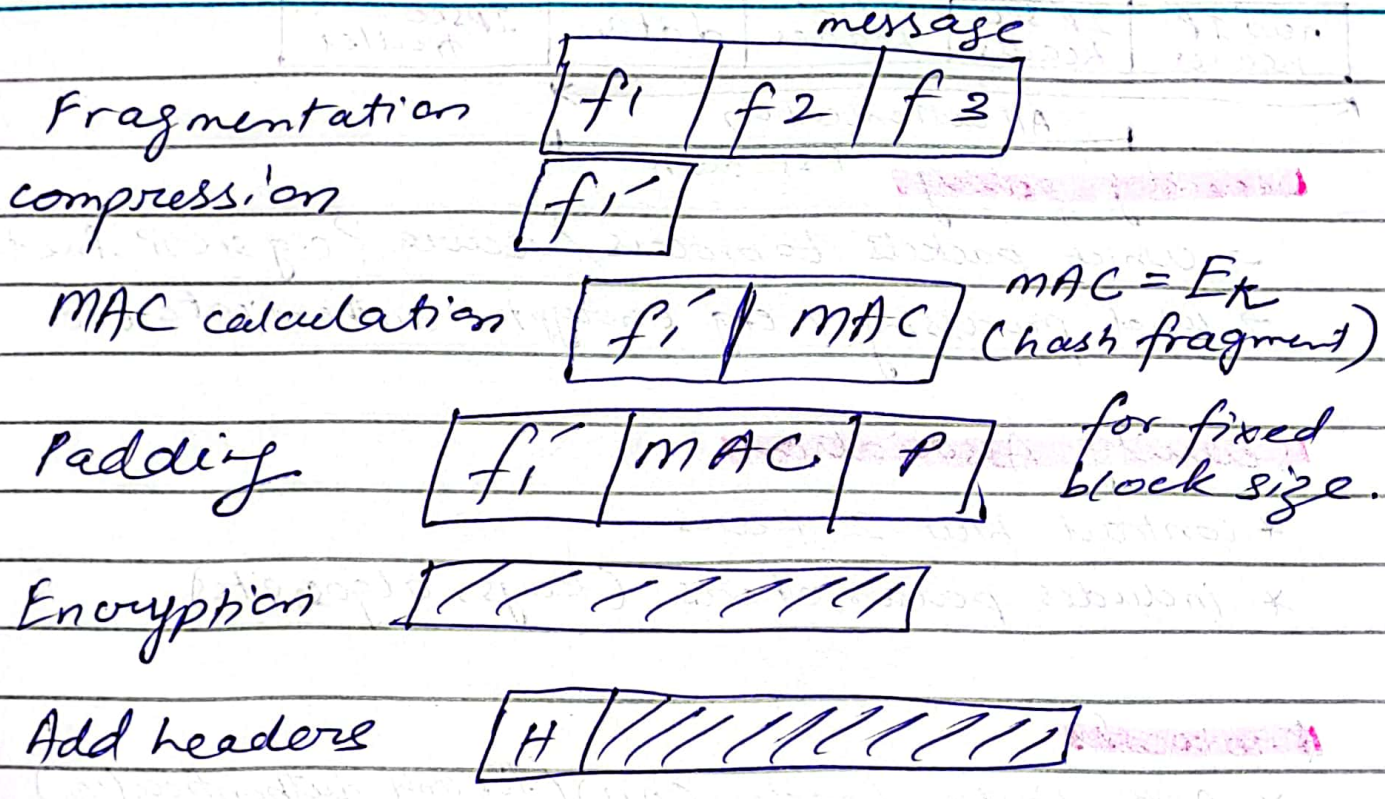
SSL Handshake.



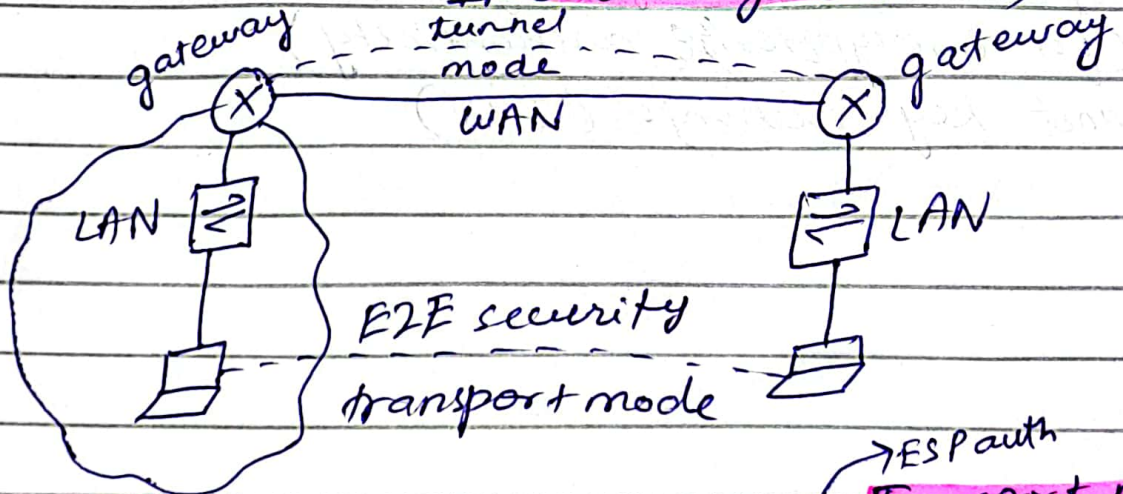
* The masterkey solves "message replay" issue because it keeps changing the packet encryption key and the packets used in one transaction won't be valid again because the key changed.

* if one side receives some packets with invalid encryptions, the side sends "alert" & both sides decide on new ciphers.

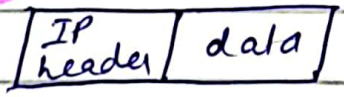
SSL Record



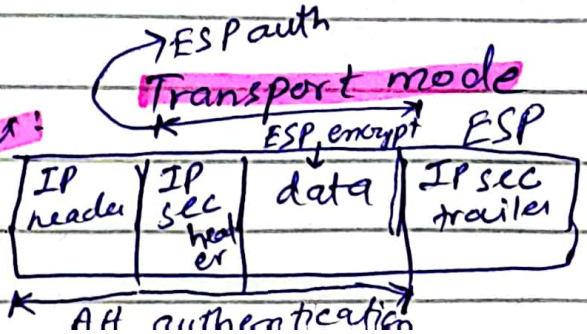
IP security (IP sec)



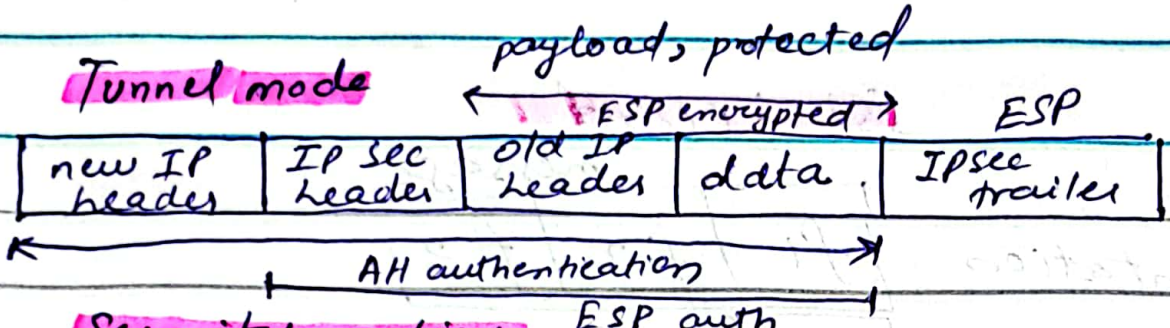
normal IP packet:



IP sec packet:



Tunnel mode



Security policy

→ which packets to process / secure ? e.g src IP fixed

→ what processing ? eg encrypt, authenticate etc

Security Association

* contract b/w 2 peers
* includes parameters (keys, algos etc)

Protocols

* Authentication header (AH) (for only authentication)
* Encapsulating security payload (ESP) (for encryption & authenticity)
* Internet key exchange (IKE)

mid 2 revision

* Malware
* Buffer overflow, string format attacks
* Database Security
* Authentication
* Web Security

Malware

* types: worms, trojan, spyware, ransomware
* obfuscation (methods)
  ↳ poly, meta, encrypted
* Infection methods
  overwriting, prepending, cavity

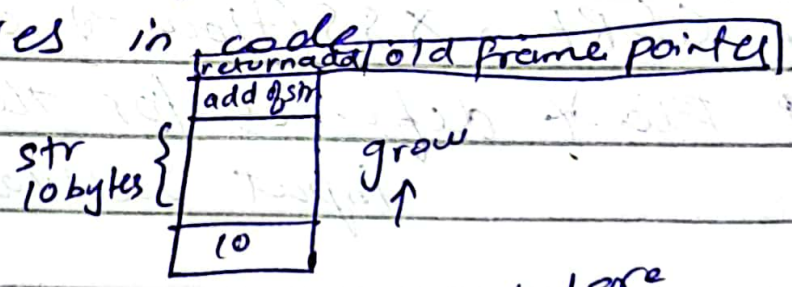
Q they look different but do the same thing when ran which obfuscation is this?

Buffer Overflow

* Identify vulnerabilities in code

e.g

```
int main(){
    int x = 10;
    char str[10];
    gets(str);
    return 0;
}
```



← draw the stack before or after this point/ inside gets

* Defense : compiletime, runtime

Q: if you don't have source code? how will you defend against buffer overflow?

Database Security

* Write malicious input (containing SQL) SQL injection queries
* Inference (concepts) (how to prevent?)

Authentication

* Types of authentication
* password based authentication – implimentation (how does server implement it?) (what issues arise?) (hashed, salted.) (challenge response, when is it used?)
* Kerberos : flow of usage (whole method)

Web Security

* CSRF, XSS (flow of events)
* ~~Flow~~ * defenses for these attacks. input validation, input encoding.