CS Final 2017

1- Hash Function Requirements

- Variable Output Size
- Fixed Input Size
- efficient
  → given M, computing $h = H(M)$ is easy on SW/HW
- Pre-image resistant (One-way Property)
  → given h, it's computationally infeasible to find M such that $h = H(M)$
- 2nd Pre-image Resistant (Weak-Collision res.)
  → given M, it's computationally infeasible to find $M' \neq M$ but $H(M') = H(M)$
- Collision Resistant (Strong-Collision resistant)
  → computationally infeasible to find any M, M' such $M \neq M'$ but $H(M) = H(M')$
- Pseudorandom Function
  → Meets standard tests of Pseudorandomness

* Note that for "characteristics of cryptographic hash function" → Book mentions one-way & strong collision properties

"good hash function" → applying it on a large set of inputs produces evenly distributed apparently random outputs

2- Attacks to identify for message security

1- Disclosure
→ release of message content

2- Traffic Analysis
→ observing frequency / duration of communication

#message confidentiality
→ use encryption
→ Dummy traffic
→ Link encryption

3- Masquerade
→ insertion of messages into the network from fradulent source

4- Content Modification
→ Message content is changed (including insertion, deletion,..)

5- Sequence Modification
→ Message order is changed (or insertion, deletion,...) ← w.r.t other msgs

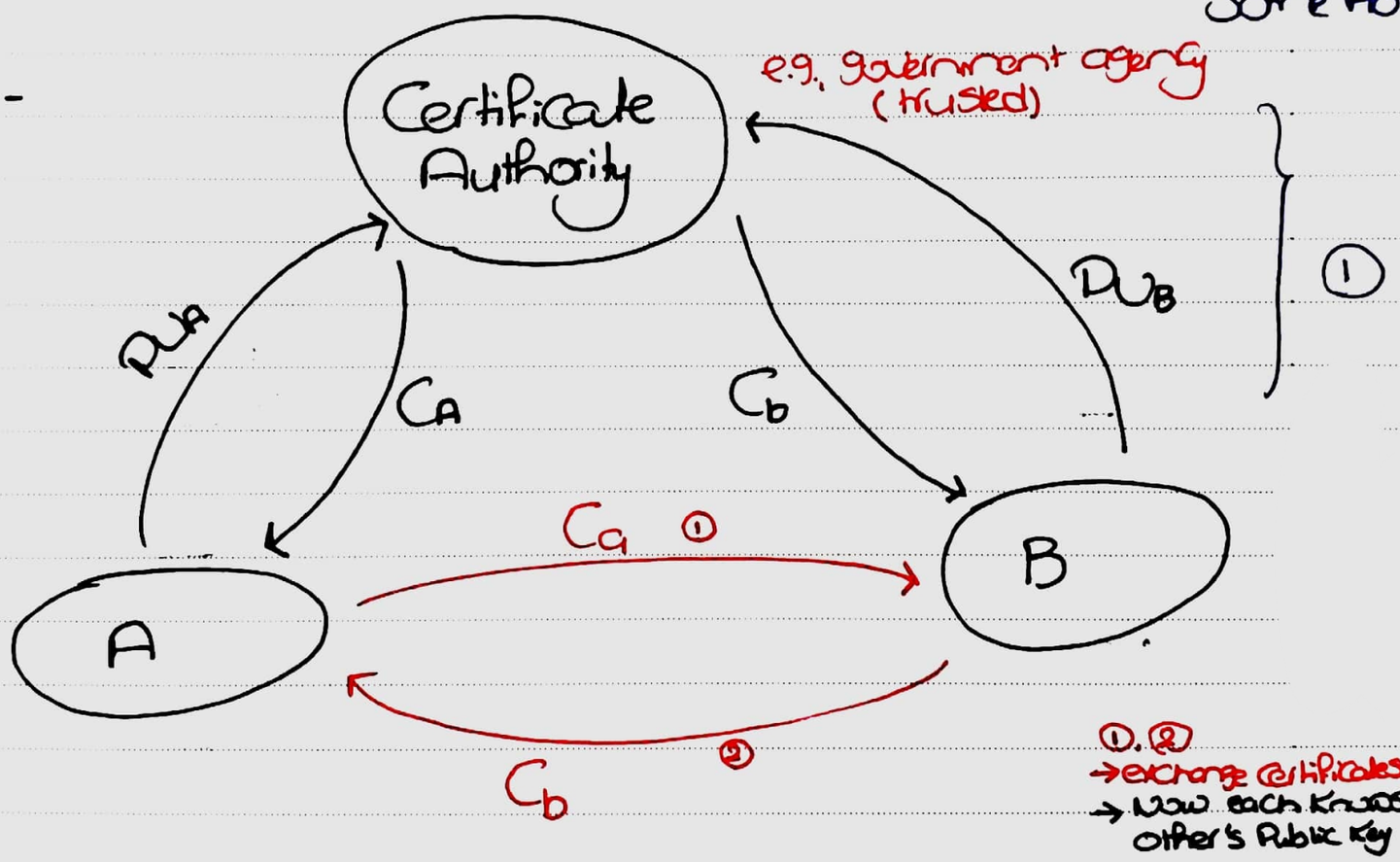
6- Timing Modification
→ Message is delayed or replayed

#Message authentication
⇒ use MAC

7- Source Repudiation
→ denial of transmission by source
→ use digital Signatures

8- Destination Repudiation
→ denial of receipt by receiver
→ use digital Signature + some Protocol

3-



①.② → exchange certificates → now each knows other's Public Key

① has to be done in Person or through secure authenticated communication.

$$C_a = E(PR_{Auth}, T_1 \| ID_A \| PU_A)$$
$$C_b = E(PR_{Auth}, T_2 \| ID_B \| PU_B)$$

- Only certificate authority can create or update certificates
- Any participant can verify → source is certificate authority (not counterfeit)
  - → currency of certificate
  - → read owner's name & Public key

4- Design Objectives of HMAC

→ Allow direct use of available hash functions (without modification & freely available)

→ allow replacibility of used hash function when another that is faster/more secure is found/required.

→ Preserve original performance of hash function (no significant degradation)

→ based on the used hash function the strength of the authentication mechanism can be analyzed (i.e., based on 'reasonable assumptions' on the hash...)

→ Use & handle keys in a simple way

5- OFB

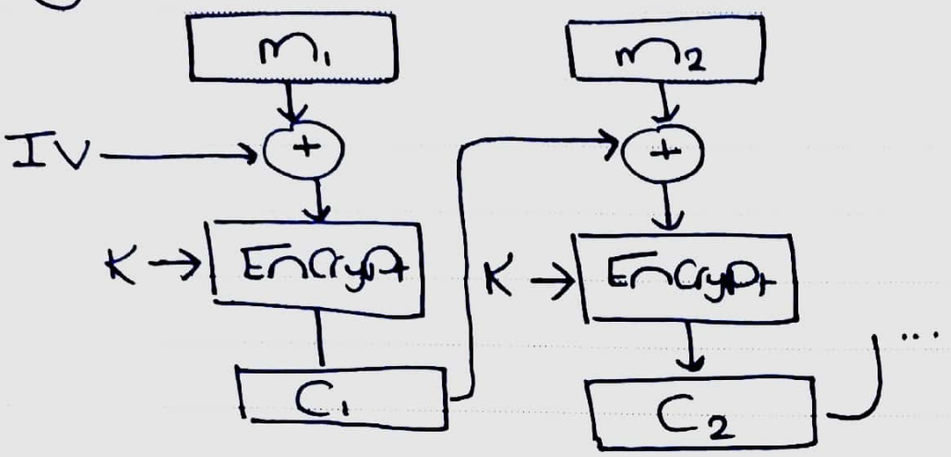
Advantages

- Repetition in message blocks are not exposed
- No need to Pad the msg
- If m_i gets corrupted, only C_i is affected (encryption)
- If C_i gets corrupted, only m_i is affected (decryption)
- Can generate O_i's before obtaining msg.
- Only need to implement Encrypt

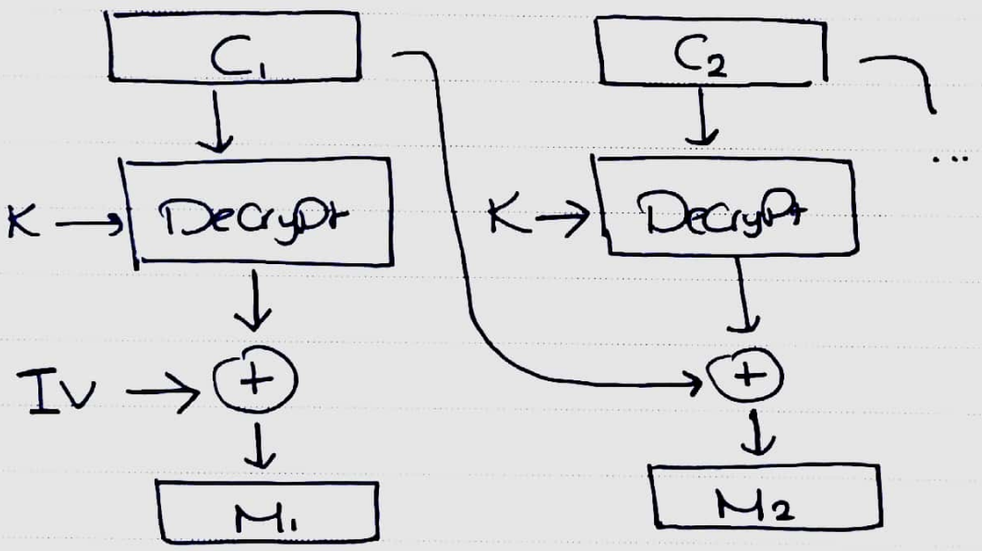
Disadvantages

- Neither encryption nor decryption can be parallelized / no random access
- Needs IV, (← NONCE) and error in that affects all C_i/m_i, needs to be synchr
- Change in m_i/C_i yields a predictable change in C_i/m_i (Vulnerable to stream modif.)

6-



- Can't Parallelize encryption (dep. on Previous Stage Output)



- Can Parallelize decryption (only dep. on input from Previous Stage)

7-

Class of Intruders

#OutSider

#InSider

#InSider or OutSider

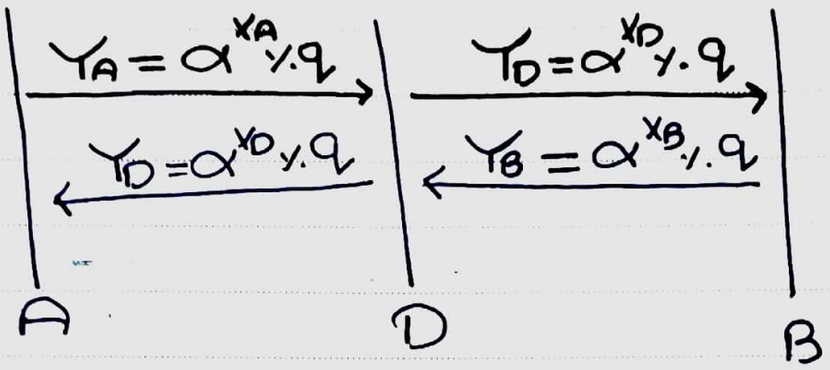
Masquerador
- Not authorized to use the system
- exploits legitimate user's account by Penetrating access control

MisFeasor
- legitimate user
- access data/res. for which such access is unauthorized

Clandestine user
- Seizes Supervisory Control then uses it to evade auditing access controls, Suppress audit Collec.

8- Diffie-Hellman Man in the middle attack with one key

- generate $X_A < q$
- generate $X_D < q$
- generate $X_B < q$



$K_{AD} = Y_D^{X_A} \% q$ $K_{AD} = Y_A^{X_D} \% q$ $K_{BD} = Y_D^{X_B} \% q$

$K_{BD} = Y_B^{X_D} \% q$

9- RSA Chosen Ciphertext Attack

- closest match in book
- Also Eng. Hu

Show that if Bob intercepts a ciphertext C intended for alice and wants to obtain the corresponding message $M = C^d \bmod n$ then he can choose random $r < n$ and compute

$Z = r^e \bmod n$

$X = ZC \bmod n$

$t = r^{-1} \bmod n$

then ask alice to sign X so that he has $Y = X^d \bmod n$

then he has enough info to find M

$\because X = ZC \bmod n$

$\therefore Y = X^d \bmod n = ((ZC \bmod n)^d \bmod n)$

- $(a \bmod n)^k \bmod n = a^k \bmod n \quad = (ZC)^d \bmod n$
- $ab \bmod n = (a \bmod n)(b \bmod n) \bmod n \quad = (Z^d \bmod n)(C^d \bmod n) \bmod n$

$r < n$, $m < n$ ↓

$= (r^{ed} \bmod n)(m \bmod n) \bmod n$

$= rm \bmod n$

Clearly, $m = r^{-1} Y \bmod n$

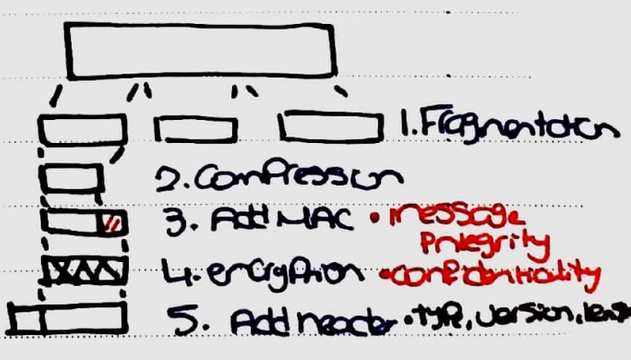
$= t Y \bmod n$

Is the Computation needed for Bob to recover the message.

10 - SSL Record Protocol Operation

→ Control flow of data between 2 end points

→ Provide services for higher level Prot.



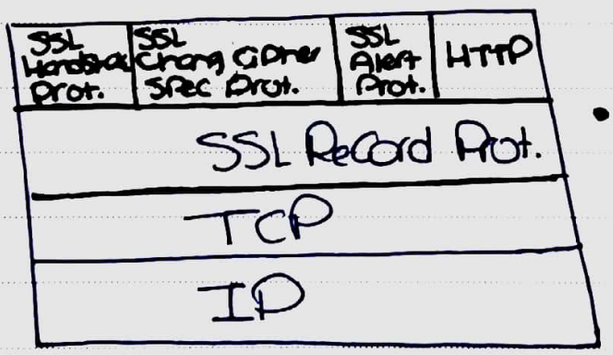
11 - SSL Protocol Header Fields

ContentType | MajorVer. | MinorVer. | CompressedLength

ContentType → e.g. data or 3 higher Protocols

CompressedLength → length of Plaintext Fragment after Comp.

12- The only figure under SSL Architecture is SSL Protocol Stack

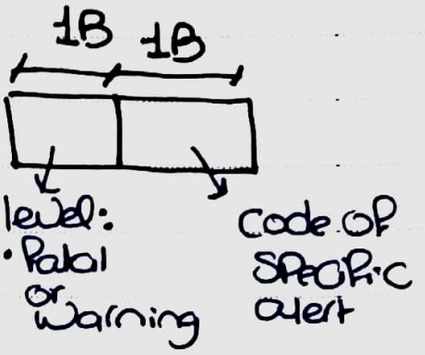


- SSL exchanges
- Security services for higher level Protocols

13- Brief description on 2 Components* of SSL Protocol

→ Change CipherSpec Protocol
- Consists of a single byte message of Content '1'
- Causes the Pending State to be Copied to Current State which updates Cipher Suite for this Connection

→ SSL Alert Protocol
- Consists of 2 bytes



→ The communication is immediately terminated if the level is fatal
e.g., bad_record_mac
→ An example with warning level is close_notify

*To convey SSL-related messages to peer-entity

14. Honey Pot

→ Decoy system used to divert potential attackers from critical systems

- Divert attackers from accessing critical systems
- Collect information based on their activity for future defense
- encourage them to stay for long enough until admins react

→ Filled with fabricated information that
- appears valuable
- wouldn't be accessed by legit user

→ System is instrumented with sensitive monitors & event loggers that detect access & collect info.