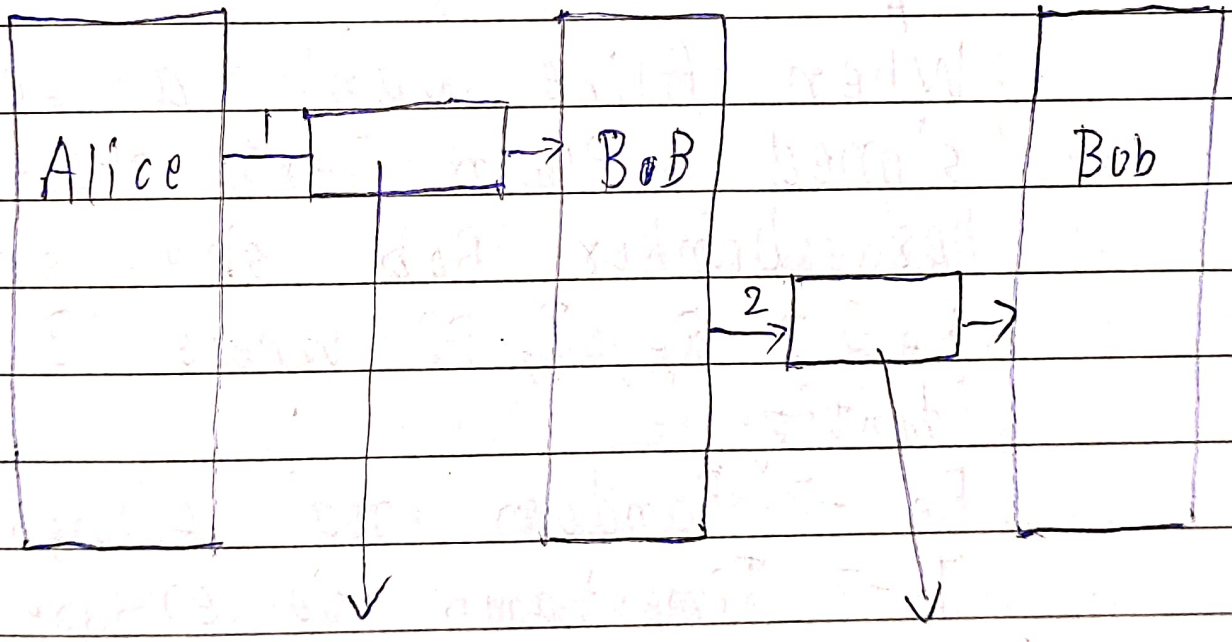
Unit - III

Digital Signatures

URBAN EDGE 5/11/25 97

1. Non Repudiation: Once msg. is signed sender cannot deny
2. Integrity: Message should not be tampered
3. Authentication

i. Symmetric Key Signatures
ii. Public Key Signatures
iii. Message Digests
iv. The Birthday Attack



$AK_A(B, R_A, T, P)$ $K_B(A, R_A, T, P, K_{BB}(A, T, P))$

K_B - Private Key of receiver B

P - Plain Text T - OTP(Timestamp)

B - Bob (receiver)

R_A - Random no. chosen by Alice

URBAN EDGE 10.11.25

One approach to digital signatures is to have a central authority that knows everything and whom everyone trusts say Big Brother (BB) each user then chooses a secret key and carries it by hand to BB's office thus only Alice & BB know Alice's secret key K_A

- When Alice wants a send a signed Plain Text message P to her banker Bob she generates $K_A(B, R_A, T, P)$ where B is Bob's identity
  R_A -> Random no. chosen by Alice
  T -> Timestamp to ensure freshness
  $K_A(B, R_A, T, P)$ -> Message encrypted with her key K_A then she sends to it BB

- BB sees that the message is from Alice decrypts it and sends a message to Bob, the message to Bob contains the PT of Alice's message and also the signed message $K_{BB}(A, T, P)$
- Bob now carries out Alice's request
- What happens if Alice later denies sending the message?

Step 1. Everyone sues everyone

Step 2. Finally when the case comes to court and Alice rigourously denies sending the ~~me~~ disputed message to Bob.

- The judge will ask Bob, how he can be sure that the disputed message came from Alice and not from someone ~~from~~ say Trudy
- Bob 1st points out that BB will not accept a message from Alice unles

it is encrypted with K_A. So, there is no possibility of Trudy sending a BB a false message from Alice without BB detecting it immediately.

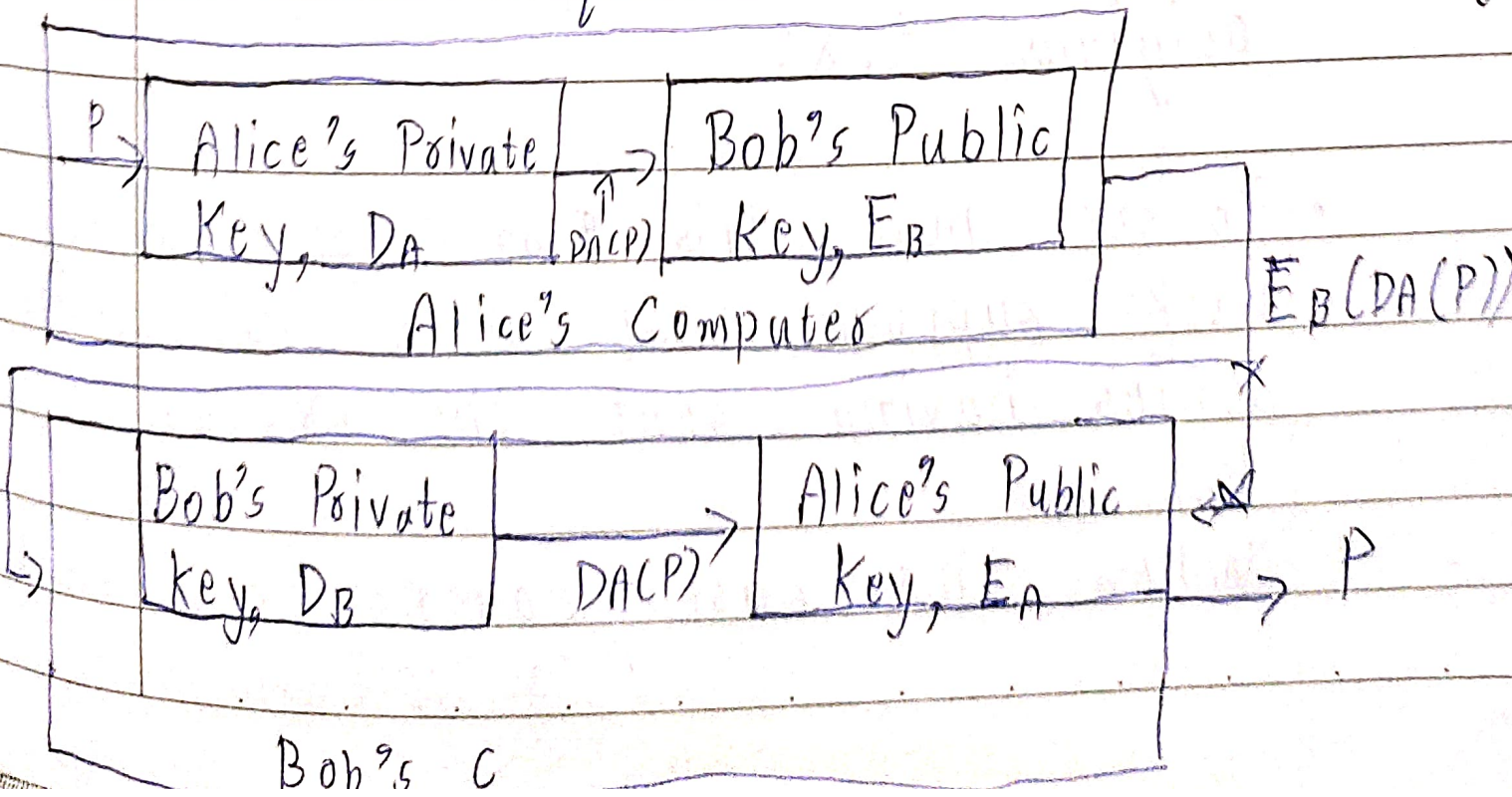
- Bob then dramatically produces exhibit A $K_{BB}(A, T, P)$
- Bob says that this is the message signed by BB, that proves Alice sent P to Bob
- The judge then asks BB to decrypt exhibit A when BB testifies that Bob is telling the truth
- The judge then decides in favour of Bob and the case is dismissed

Public Key Signatures

A structural problem with using

URBAN EDGE 17, 11, 25
101

symmetric key cryptography for digital signatures is that everyone has to agree to trust Big Brother furthermore BB gets to read all signed messages, the most logical candidates for running the BB server are the Government, the Banks, the Accountants and the Lawyers.
Unfortunately none of these inspire total confidence in all citizens. Hence, it would be nice if signing documents did not require a trusted authority



- Alice can send a signed Plain Text message P to Bob by transmitting $E_B(D_A(P))$
- Alice knows her own Private Key (D_A) ass well as Bob's Public key (E_B) so constructing this message by Alice is easier one.

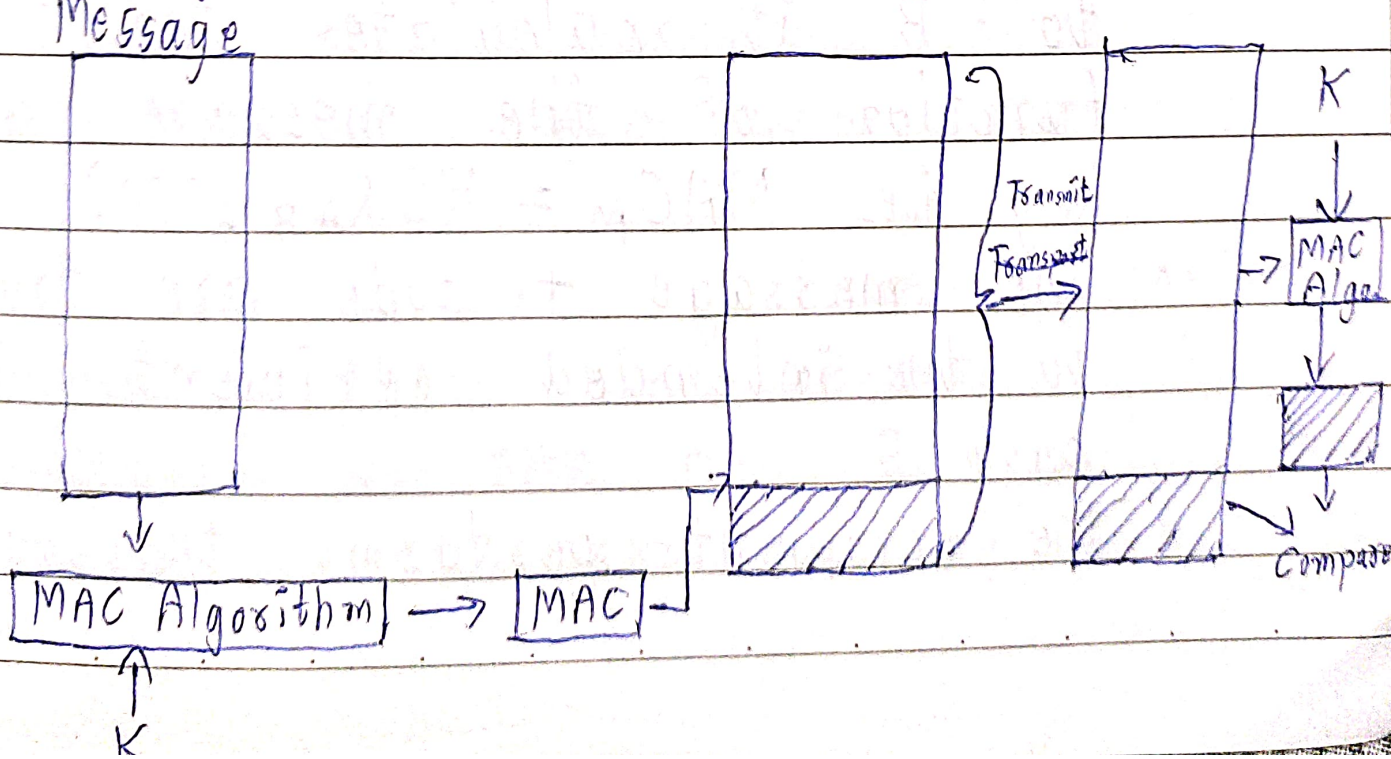
- When Bob receives the message he transforms it using his Private key D_B yielding $D_A(P)$
- He then applies E_A to get the original text

- To see how this signature property works suppose that Alice subsequently denies having sent the message P to Bob ~~at~~.
- When the case comes up in court

Bob can produce both P and $D_A(P)$

- The judge can easily verify that Bob indeed has a valid message encrypted by D_A, by simply applying E_A to it.
- Since Bob doesn't know Alice's private key, the only way Bob could have acquired a message Encrypted by Alice, if Alice did indeed send it.

Message Authentic Code (MAC)



- One authentication technique involves the use of a secret key to generate a small block of data known as a message authentication code (MAC)
- This MAC is appended to the message
- This technique assumes that the two communicating parties say A & B share a secret key K_{AB}
- When A has the message to send to B it calculates the MAC as function of the message and a key i.e. $MAC_M = F(K_{AB}, M)$
- The message + code are transmitted to the intended recipient in this case B
- The recipient performs the same

calculation on the received message. using the same secret key to generate a new authentication code.
- The received code is compared with the calculated code.
- If you assume that only the receiver & sender know the identity of the secret key, if the received code matches with the secret code then :-

i) The receiver is assured that the message has not been altered.
ii) The receiver is assured that the message is from the intended sender only
iii) If the message includes a sequence number then the receiver can be assured of the proper sequence. because an attacker cannot successfully

after the sequence number

Hashing

- The one-way hash function or secure hash function is ~~not~~ important not only in message authentication but also in digital signatures.
  i) H ~~ha~~ can be applied to a block of data of any size
  ii) It produces a fixed length output
  iii) H(x) is relatively easy to compute for any given x making both hardware and software implications practical.
  iv) For any given value h it is computationally infeasible to find * such that H(x) = h.
- This is sometimes referred to as

one-way property.

v) For any given block x it is computationally infeasible to find y != x with H(x) = H(y)
- This is sometimes referred to as weak collision resistance

vi) It is computationally infeasible to find any pair (x, y) such that H(x) = H(y)
- This is sometimes referred to as strong collision resistance.

Simple Hash Function

One of the simplest hash functions is Bit-by-Bit exclusive of every block, this can be expressed as follows:

$C_i = b_{i1} \oplus b_{i2} \oplus \ldots b_{im}$

- where C_i is the i^{th} bit of Hash code m is no. of n bit blocks in

the input
b_{ij} = i^{th} bit in j^{th} block
⊕ XOR operation

Hash based MAC

- A message digest is a fixed length unique fingerprint of a message created by a one way cryptographic hash function
- It ensures data integrity it is used to verify that a message has not been altered during transmission ~~or~~ because even a small change to the original message will produce a complete different message digest

How it Works?

i) Hashing: The sender takes a messa

of any size and runs it through a cryptographic hash function such as SHA-256 or MD-5 to produce a fixed size string called the message digest

ii) Transmission: The sender sends both the original message and its corresponding message digest to the receiver.

iii) Verification: The receiver uses the same hash function on the received message to generate a new message digest

iv) Comparison: The receiver compares the newly generated message with the one sent by the sender.

a) If the message values match, the msg. is verified as authentic & has not been tampered with

b) It don't match the msg. was altered during transmit

Key Characteristics

i) Fixed Length: The msg-digest always has the same length regardless of the size of the original message

ii) One-way function: It is computationally infeasible to reverse the process and reconstruct the original msg. from its digest

iii) Collision resistance: It is extreme difficult if not impossible t two different msgs to produc the same msg. digest

iv) Unique identifier: The digest acts as a unique identifier or digital

fingerprint for the message data.

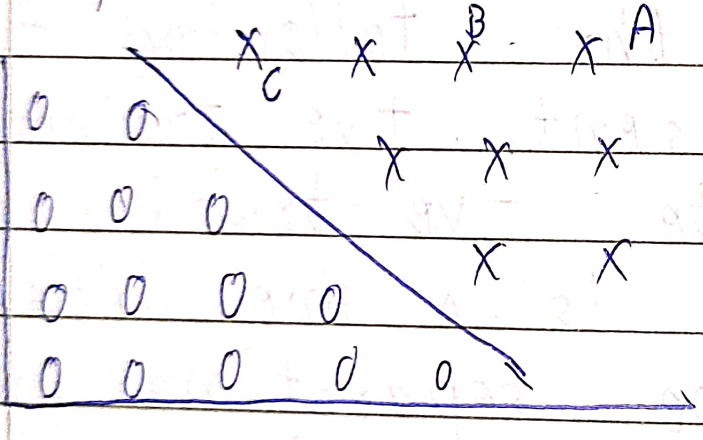
UNIT - IV

DETECTING CYBER THREATS WITH AI

Spam Detection using SVM

- It is a supervised ML algo. for classification.
- Consider the following figure
- X's represent +ve training examples
- O's denote -ve training examples
- Also there is a line decision boundary the separating hyperplane
- We have three points labelled A, B & C
- The point A is very far from the decision boundary.
- The point C is very close to the decision boundary.

- B is between A & C
- All the points A, B & C belong to the +ve class
- A little change to the hyperplane would make C belonging to -ve class.
- However, it will not have any impact on the class of Point A



Support Vectors are the data points that lie closest to the decision surface or hyperplane

- They are the data points most difficult to classify.