Cryptography & Network Security

Unit-I

Date __/__/____

* Basic Cryptographic Techniques

- The basic function of cryptography are encryption, decryption and cryptographic hashing.
- Cryptography can be used to provide message confidentiality, integrity & sender verification.
- In order to encrypt & decrypt message, the sender & recipient need to share a key that is used by cryptographic algorithm. The key used by sender to encrypt message (transform plain text into cipher text [encrypted text]) & by recipient to decrypt the message (transform cipher text into plain text). This process can be done on a fixed message such as email or a communication stream such as TCP/IP connection.
- Cryptographic hashing is the process of generating a fixed-length string from a message of arbitary length.
- Modern cryptographic systems are based on complex mathematical relation and processes.

* Computational Complexity.

- It is the study of the minimal resources needed to solve computational problems.
- In particular, it aims to distinguish between those problems that possess efficient algorithm (the "easy" problems) & those that are inherently intractable (the "hard" problems).

* Finite Fields

- A field is a set of elements on which two arithmetic operations (addition and multiplication) have been defined and which has properties of ordinary arithmetic such as closure, associativity, commutativity, distributivity and having both additive & multiplicative inverses.
- Modular arithmetic is a kind of integer arithmetic that reduces all numbers to one of a fixed set [0.....n-1] for some number n. Any integer outside this range is reduced to one in this range by taking the remainder after division by n.
- The greatest common divisor of two integers is the largest

Punk

Date __/__/____

positive integer that exactly divides both integers.

- Finite fields are important in several areas of cryptography. A finite field is simply a field with a finite no. of elements. It can be shown that the order of a finite field (no. of elements in the field) must be a power of a prime p^n, where n is a positive integer.
- Finite fields of order p can be defined using arithmetic mod p.
- Finite fields of order p^n, for $n>1$ can be defined using arithmetic over polynomials.
- Groups, Rings & Fields

→ A group G, $\{G, \cdot\}$, is a set of elements with binary operation, ·, that associates to each ordered pair (a,b) of elements in G an element $(a.b)$ in G such that closure, associate, identity element & inverse element axioms are followed.

→ A ring R, $\{R, +, \times\}$, is a set of elements with two binary operations, + & × such that for all a, b, c in R, {the abelian group (commutative law following group), closure under multiplication, associativity of multiplication, distributive laws, commutative law of multiplication, integral domain} axioms are followed.

→ A field F, $\{F, +, \times\}$ is a set of elements with binary operations, addition & multiplication such that all a, b, c in F follows axioms of ring & field as well as multiplicative inverse axiom.

- Modular Arithmetic

$$a = qn + r$$

Modulus: $a = [a/n] \times n + (a \bmod n)$

Ex:- $a=11 \quad n=7$ then $11 = 1\times 7 + 4$ $\quad r=4, q=1$

$a=-11 \quad n=7$ then $-11 = (-2)\times 7 + 3$ $\quad r=3, q=-2$

Congruent Modulo n: (Two integer a & b) if $(a \bmod n) = (b \bmod n)$

This can be written as $a \equiv b \pmod n$

Ex:- $73 \equiv 4 \pmod{23}$; $21 \equiv -9 \pmod{10}$

If $b|a$, b is divisor of a.

① $a \equiv b \pmod n$ if $n|(a-b)$

② $a \equiv b \pmod n \Rightarrow b \equiv a \pmod n$

③ $a \equiv b \pmod n$, $b \equiv c \pmod n$ $\Rightarrow$ $a \equiv c \pmod n$

- Modular Arithmetic operations

① $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$

Punk

① $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$

③ $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

→ Finding the smallest nonnegative integer to which k is congruent modulo n is called reducing k modulo n.

Date __/__/____

- Euclidean Algo

→ Greatest Common Divisor

Defⁿ: $\gcd(a,b) = \max[k \text{ such that } k|a \text{ \& } k|b]$

Ex: ~~gcd (a,b (60,24) = gcd (60, 24) = 12~~

In general $\gcd(a,b) = \gcd(|a|,|b|)$

Ex:- $\gcd(60,24) = \gcd(60,-24) = 12$

→ Find gcd

$\gcd(a,b) = \gcd(b, a \bmod b)$

Ex:- $\gcd(55,22) = \gcd(22, 55 \bmod 22) = \gcd(22,11) = 11$

→ Euclid (a,b)

1.) $A \leftarrow a$; $B \leftarrow b$

2.) if $B = 0$ return $A = \gcd(a,b)$

3.) $R = A \bmod B$

4.) $A \leftarrow B$

5.) $B \leftarrow R$

6.) goto 2

Ex:- $\gcd(26,12)$ ⁂ $26 = 1 \times 16 + 10$ ~~gcd (16,6)~~ $\gcd(16,10)$

$16 = 1 \times 10 + 6$ $\gcd(10,6)$

$10 = 1 \times 6 + 4$ $\gcd(6,4)$

$6 = 1 \times 4 + 2$ $\gcd(4,2)$

$4 = 2 \times 2 + 0$ $\gcd(2,0)$

$\therefore \gcd(26,12) = 2$

- Finite fields of form GF(p)

Simplest GF(2)

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

w	−w	w^{-1}
0	0	–
1	1	1

Simplest GF(3)

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

×	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

w	−w	w^{-1}
0	0	–
1	2	1
2	1	

- find gcd of polynomial

$\gcd[a(x), b(x)] = \gcd[b(x), a(x) \bmod b(x)]$

Punk

Date __/__/____

* Number Theory

- A prime number is an integer that can only be divided without remainder by +ve & -ve values of itself & 1. Prime no. plays critical role in cryptography.
- Two theorams that play important roles in public-key cryptography are Fermat's theoram & Euler's theoram.
- An important requirement in a number of cryptographic algorithms is the ability to choose a large prime number. An area of ongoing research is the development of efficient algorithms for determing if a randomly chosen large integer is a prime number.
- Discrete logarithms are fundamental to a number of public-key algorithms. Discrete logarithms are analogous to ordinary logarithms, but operate over modular arithmetic.
- Fermat's Theoram
  - → If p is prime & a is +ve integer not divisible by p then
    $$a^{p-1} \equiv 1 \pmod{p}$$
  - → Alternate Form:
    $$a^p \equiv a \pmod{p}$$

    Ex:- $3^5 \equiv 243 \equiv 3 \pmod{5}$ if $p=5$, $a=3$
    $= a \pmod{p}$
- Euler's Totient function

  $\phi(p) = p-1$ p,q prime nos.

  For $n=pq$
  $$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p-1) \times (q-1) \text{ if } p \neq q$$
- Euler's Theoram
  $$a^{\phi(n)} \equiv 1 \pmod{n}$$

  Ex:- $a=3$, $n=10$, $\phi(10)=4$ then $a^{\phi(n)} = 3^4 = 81 \equiv 1 \pmod{10}$
  $\equiv 1 \pmod{n}$
- Discrete logarithms
  $b \equiv a^i \pmod{p}$ where $0 \leq i \leq (p-1)$

  i is referred as discrete logarithm of number b for base a (mod p) & can denote this value as $d\log_{a,p}(b)$

Punk

Date __/__/____

* The OSI Security Architecture (Imp)

→ The OSI (Open System Interconnection) security architecture provides a systematic framework for defining security attacks, mechanisms and services.

→ "Security attacks" are classified as either passive attacks, which include unauthorized reading of a message of file and traffic analysis; and active attacks such as modification of messages or files, and denial of service.

→ A "Security Mechanism" is any process (or a device incorporating such a process) that is designed to detect, prevent or recover from a security attack. Examples are encryption algorithms, digital signatures and authentication protocols.

→ "Security Services" includes authenication, access control, data confidentiality, data integrity, nonrepudiation, and availability.

- Security Attacks

Types

(i) Active Attack: An active attack attempts to alter system resources or affect their operation. Here modification of original message is done. This attack can't be prevented easily. Active attacks are of three types:

(a) Interruption (b) Modification (c) fabrication (DOS attack)

(ii) Passive Attack: In this, the attackers aims to obtain the information. The attacker doesn't pretend or perform any modification in data, i.e., they are harder to detect. There are two types of passive attacks

(a) Release of message content to others

(b) Traffic Analysis

Hacker try to analyse message using a pattern that provide some clues regarding the communication.

Punk

Date __/__/____

*Security Services

(i) Authentication: Assurance that communicating entity is the one that it claims to be.

(A) Peer Entity Authentication: Used in association with a logical connection to provide confidence in the identity of the entities connected

(B) Data Origin Authentication: In a connectionless transfer, provides assurance that the source of received data is as claimed.

(ii) Access Control: Prevention of unauthorized use of a resource

(iii) Data confidentiality: Protection of data from unauthorized disclosure

(A) Connection Confidentiality: Protection of all user data on a connection

(B) Connectionless Confidentiality: Protection of all user data on single data block

(C) Selective-field Confidentiality: Confidentiality of selected fields within the user data on a connection or in a single data block.

(D) Traffic Flow Confidentiality: Protection of the information that might be derived from observation of traffic flows.

(iv) Data Integrity: The assurance that data received are exactly as sent by an authorized entity

(A) Connection Integrity with Recovery: Provides integrity of all user data on a connection with recovery options in case of modification detection

(B) Connection Integrity without Recovery: Provides integrity of all data with only detection not recovery.

(C) Selective-field Connection Integrity: Provides integrity of selected fields within user data of data block transferred over a connection

(D) Connectionless Integrity: Provides for integrity of a single connectionless data block

(E) Selective-field Connectionless Integrity: Provides integrity of selected field within a single connectionless data block.

(v) Non Repudiation: Provides protection against denial by one of the entities involved in all or part of the communication

(A) Nonrepudiation, origin: Proof that message was sent by specified party

(B) Nonrepudiation, Destination: Proof that the message was received by the specified party.

Punk

Date __/__/____

★

– Security Mechanisms

(i) Specific Security Mechanisms: May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

(A) Encipherment: The use of mathematical algorithms to transform data into a form that is not readily intelligible.

(B) Digital Signature: Data appended to a data unit that allows a recipient of data unit to prove the source & integrity of data unit & protect against forgery.

(C) Access Control: variety of mechanisms that enforce access rights to resources.

(D) Data Integrity: variety of mechanisms to assure integrity of data.

(E) Authentication Exchange: Mechanisms intended to ensure identity of an entity by means of information exchange.

(F) Traffic Padding: Insertion of bits into gaps in a data stream ~~that~~ to frustrate traffic analysis attempts.

(G) Routing Control: Enables ~~particular~~ selection of a particular physically secure routes for certain data & allows routing changes, especially when a breach of security is suspected.

(H) Notarization: The use of a trusted third party to assure certain properties of a data exchange.

(ii) Pervassive Security Mechanisms: Mechanisms that are not specific to any particular OST security service or protocol layer.

(A) Trusted Functionality: That which is perceived to be correct wrt some criteria (such as security policy etc.)

(B) Security Label: The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

(C) Event Detection: Detection of security-relevant events.

(D) Security Audit Trail: Data collected & potentially used to facilitate a security audit, which is an independent review & examination of system records & activit

(E) Security Recovery: Deals with requests from mechanisms, such as event handling & management functions, & takes recovery actions.

Punk

Date __/__/____

* Classical Encryption Techniques

- Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using same key. It is also known as conventional encryption.
- Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algo. Using the same key & a decryption algo, the plain text is recovered from ciphertext.
- The two types of attack on an encryption algorithm are cryptanalysis, based on properties of the encryption algorithm, & bruteforce, which involves trying all possible keys.
- Traditional (precomputer) symmetric ciphers use substitution and/or transposition techniques. Substitution techniques map plaintext elements (character, bits) into ciphertext elements. Transposition techniques systematically transpose the positions of plaintext elements.
- Rotor machines are sophisticated precomputer hardware devices that use substitution techniques.
- Steganography is a technique for hiding a secret message within a larger one in such a way that others can't discern the presence or contents of the hidden message.
- A symmetric cipher model has five ingredients:
  (i) Encryption algorithm (ii) Decryption Algorithm
  (iii) Plain Text (iv) Cipher Text
  (v) Secret Key
- Cryptography systems are characterized along three independent dimensions:
  (i) The type of operations used for transforming plaintext to ciphertext
  (ii) The no. of keys used
  (iii) The way in which plaintext is processed.

— Substitution Techniques

(i) Caesar Cipher

Encryption : $C = E(k, p) = (p + k) \bmod 26$

Decryption : $p = D(k, C) = (C - k) \bmod 26$

Punk

Date __/__/____

Ex:- Plain Text: HelloWorld

k=3

H+3 = K E+3 = H

Cipher Text: KHOORZRUOG

. If Z+3 then Z+3 = C. (Rotate A)

→ Drawback: Only 25 keys to try

(ii) Monoalphabetic Cipher

No. of keys = 26!

Problem: If the plain text is non compressed English then analyst can exploit regularities of language.

'E' is the most frequently used character in encryption. If the character most frequently used in cipher can be replaced with 'E' & similarly a pattern that correspond to plain text may be generated

(iii) Playfair Cipher

→ A matrix is constructed by filling the letters of keyword from left to right, top to bottom in matrix omitting repeated characters in keyword. Remaining matrix filled with remaining letters in alphabetic order. I & J count as one letter

→ Using the matrix, plain text is encrypted.

→ Algorithm:

① Choose Keyword (Ex:- "Playfair Encryption")

② Enter characters of keyword in 5X5 matrix row-wise left to right

③ Fill remaing spaces in matrix with rest of English alphabets

④ Combine I & J in same cell.

Encryption process:

① Broke the plain text in group of 2 letters

② If two letters same in plain text add 'x' b/w them
If only 1 letter left at the end of pairing add 'x' to it to form a pair

Punk

Date __/__/____

③ If both alphabet in pair appear in same row of matrix, replace them with alphabets to their immediate right & if it is last alphabet in row, replace with the first alphabet of the same row

④ If the alphabet pair appear in same column of matrix, replace them with their immediate below & if last in column, replace with first of same column

⑤ If alphabet are not in same row or column, replace with alphabet in same row but at the other pair of corners (exchange column no for both)

Ex:-

P	L	A	Y	F
I/J	R	E	N	C
T	O	B	D	G
H	K	M	Q	S
U	V	W	X	Z

Ex:- BALLON

BA LX LO NX

↓ ↓ ↓ ↓

ME YV RK DY

Same Colum — diff r & c

Plain Text: Ballon

Cipher Text: MEYVRKDY

(iv) Hill Cipher

Encryption: $C = KP \bmod 26 = E(K, P)$

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

for $m = 3$ } if $m=4$ then 4×4 & so on

Decryption: $P = D(K, P) = K^{-1}C \bmod 26$

Ex:- $K = \begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix}$ $K^{-1} = \begin{pmatrix} 9 & 2 \\ 1 & 15 \end{pmatrix}$

A → 0

Z → 25

Plain Text = HMORE

$\text{MOON} = \begin{pmatrix} 12 \\ 14 \end{pmatrix}$ $R = \begin{pmatrix} 17 \\ 4 \end{pmatrix}$

O E

$\begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}\begin{pmatrix} 12 \\ 14 \end{pmatrix}_{\bmod 26} = \begin{pmatrix} 172 \\ 246 \end{pmatrix} \Rightarrow \begin{pmatrix} 16 \\ 12 \end{pmatrix} = \begin{matrix} q \\ m \end{matrix}$

$\begin{pmatrix} 5 & 8 \\ 17 & 3 \end{pmatrix}\begin{pmatrix} 17 \\ 4 \end{pmatrix}_{\bmod 26} = \begin{pmatrix} 117 \\ 301 \end{pmatrix} \Rightarrow \begin{pmatrix} 13 \\ 15 \end{pmatrix} = \begin{matrix} n \\ p \end{matrix}$

Cipher Text = QMNP

Punk

Date __/__/____

~~Length of key = length of text~~

(v) One Time Pad : Same cipher text but different key results different result

- security entirely due to randomness
- Difficulty:- Requires millions of random characters on a regular basis
  Problem of key distribution & protection

Ex- Cipher Text : ANKYODKYUR

Key 1 : pxlmvmsydo

Result 1 : mr mustard

Key 2 : mfugpmiydg

Result 2 : ms scarlet

- Transposition Technique

(i) Railfence Technique

Plain Text written down as sequence of diagonals & then read off as a sequence of rows.

Ex:- Plain Text: meet me after the toga party

m e e t m e a f t e r t h e t o g a p a r t y

Cipher Text: mematrhtgpryetefeteoaat

(ii) A more complex scheme is to write message row by row & read message off, column by column but permutate column order. Order of column becomes key.

Key: 4 3 1 2 5 6 7

Plain Text: a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z

Cipher Text: ttnaaptmtsuoaodwcoixknlypetz

* Block Ciphers

- A block cipher is an encryption/decryption scheme in which a block of plaintext is treated as a whole and used to produce a cipher block of equal length
- A stream cipher is one that encrypts a digital data stream

Punk

Date __/__/____

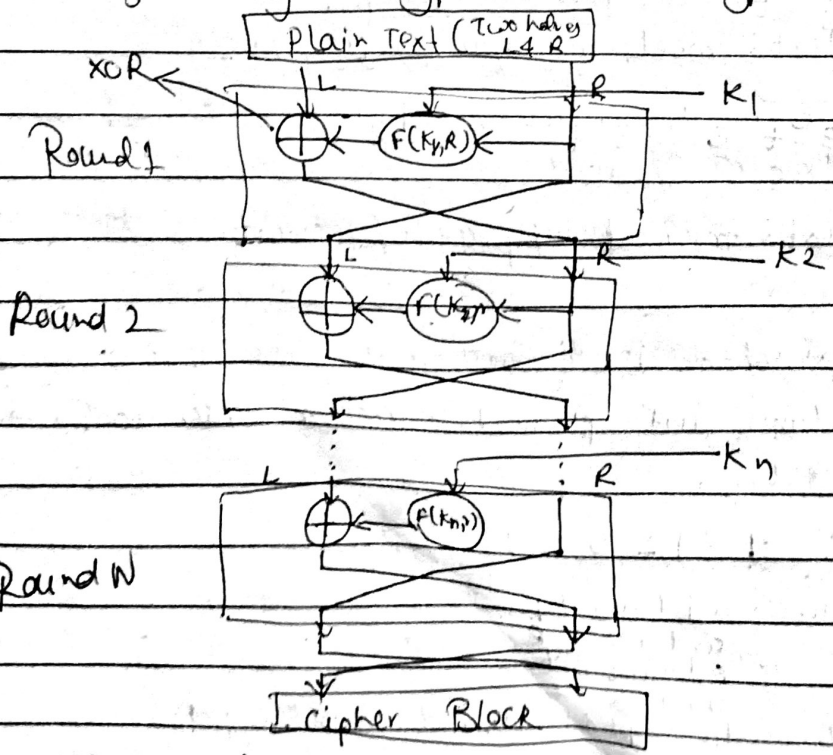
one byte at a time. Ex of classical steam ciphers are the autokeyed Vigenère cipher & the Verman cipher.

- A block cipher is one in which a block of plain text is treated as a whole & used to produce a cipher text block of equal length.
- Many block ciphers have a Feistel structure. Such a structure consists of a no. of identical rounds of processing. In each round, a substitution is performed on one half of the data being processed, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round.

- Fiestel Cipher

→ It is a design model from which many different block ciphers are derived. DES is just one example of a Fiestel Cipher.

→ A cryptographic system based on Fiestel Cipher structure uses the same algorithm for encryption and decryption.

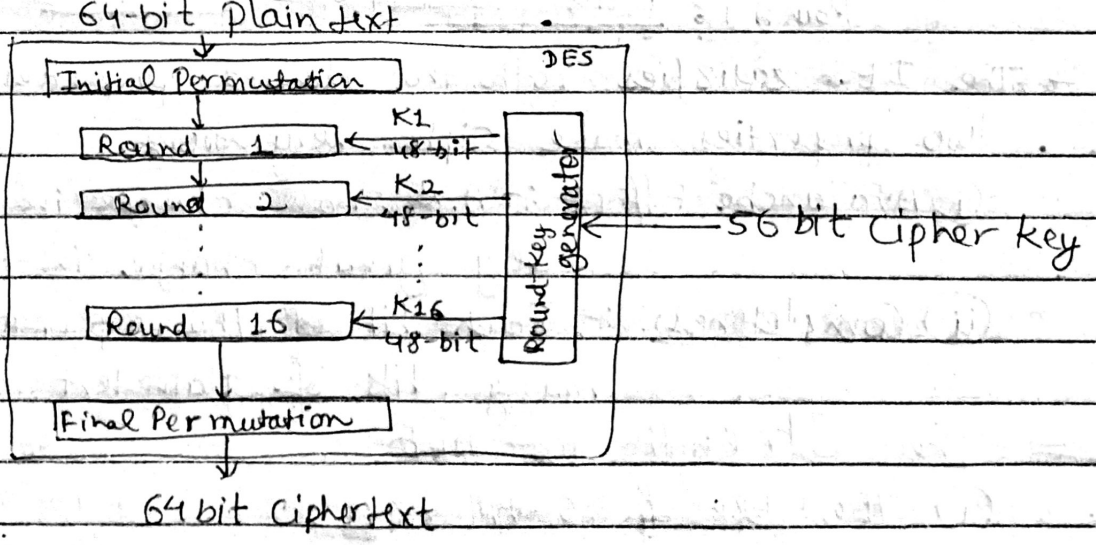


→ The essence of the approach is to develop a block cipher with a key length of k bits & a block length of n bits, allowing a total of 2^k possible transformations.

Date __/__/____

* The DES Algorithm (Imp)

- Data Encryption Standard is an outdated symmetric-key method of data encryption. It works by using the same key to encrypt & decrypt the message, so both the sender and the receiver must know & use the same private key. It is a symmetric-key block cipher published by NSIT
- The main parts of the algorithm are:
  - (i) Fractioning of the text into 64-bit (8 octet) blocks
  - (ii) Initial permutation of the block
  - (iii) Breakdown of the blocks into 2 parts: left & (L) & right (R)
  - (iv) Permutation & Substitution steps repeated 16 times
  - (v) Rejoining of the left & rights & then inverse initial permutation
- DES is an implementation of Fiesel Cipher & uses 16 round Fiesel structure with block size of 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits since 8 bits of 64 bits are not used by DES encryption algorithm (function as check bits only).



- Basic Principle

→ It is a symmetric encryption system that uses 64-bit blocks, 8 bit of which are used for parity check.

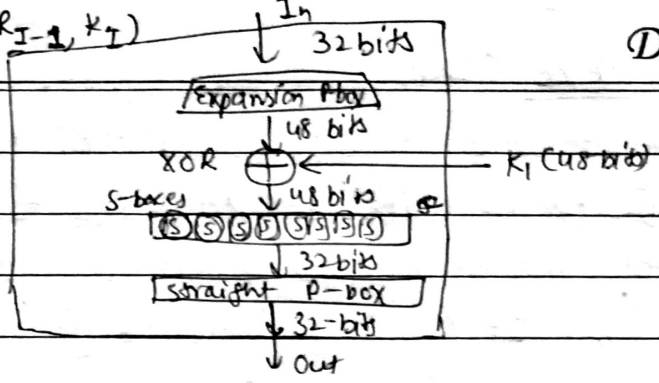
→ Each of the key parity bits is used to check one of the key's octet by odd parity, i.e., each of the parity bit is adjusted to have an odd number of '1's in the octet it belongs to.

→ The key therefore has a "useful" length of 56-bits.

Punk

- Round function

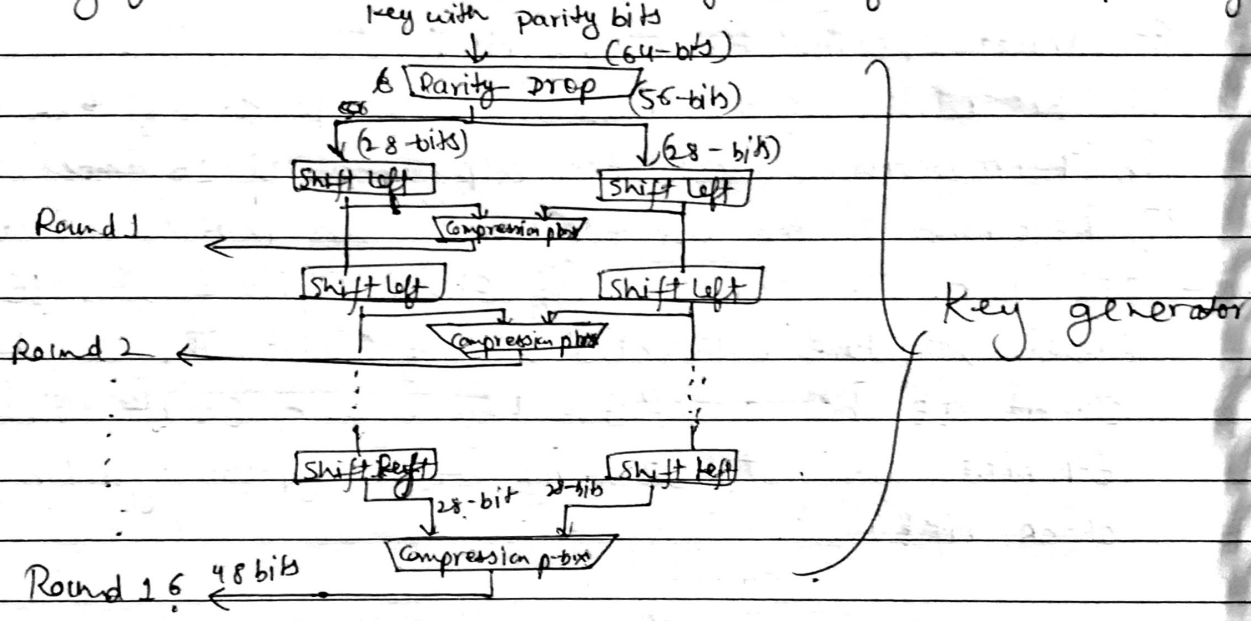
DES function applies a 48-bit key to the rightmost 32 bits to produce 32-bit output



Date __/__/____

- Key Generation

Round-key generator creates 16 48-bit keys out of a 56-bit cipher key.



- The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

(i) Avalanche Effect :- A small change in plain text results in very great change in the cipher-text

(ii) Completeness :- Each bit of the cipher text depends on many bits of plain text

- The Strength of DES

(i) The Use of 56-bit keys

→ There are 2^{56} keys (approx 7.2×10^{16} keys)

→ Broken in July 1998 by Electronic Frontier Foundation (EFF) using a special purpose "DES cracker" machine that was built less than $250,000

→ If the text message has been compressed before encryption then recognition is more difficult.

(ii) Nature of DES Algorithm

Punk

Date __/__/____

→ Cryptoanalysis is possible by exploiting the characteristics of DES (the eight substitution tables or s-boxes used in each iteration).
→ Despite this, no one has so far succeeded in discovering the supposed fatal weaknesses in the s-boxes.

(iii) Timing Attacks

→ In essence, a timing attack is one in which info about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various cipher texts.
→ A timing attack exploit the fact that an encryption or decryption algorithm often takes slightly different amount of time on different inputs. [HEVI99] reports on an approach that yields the Hamming weight (no. of bits equal to one) of the secret key.

* The AES Cipher (Imp)

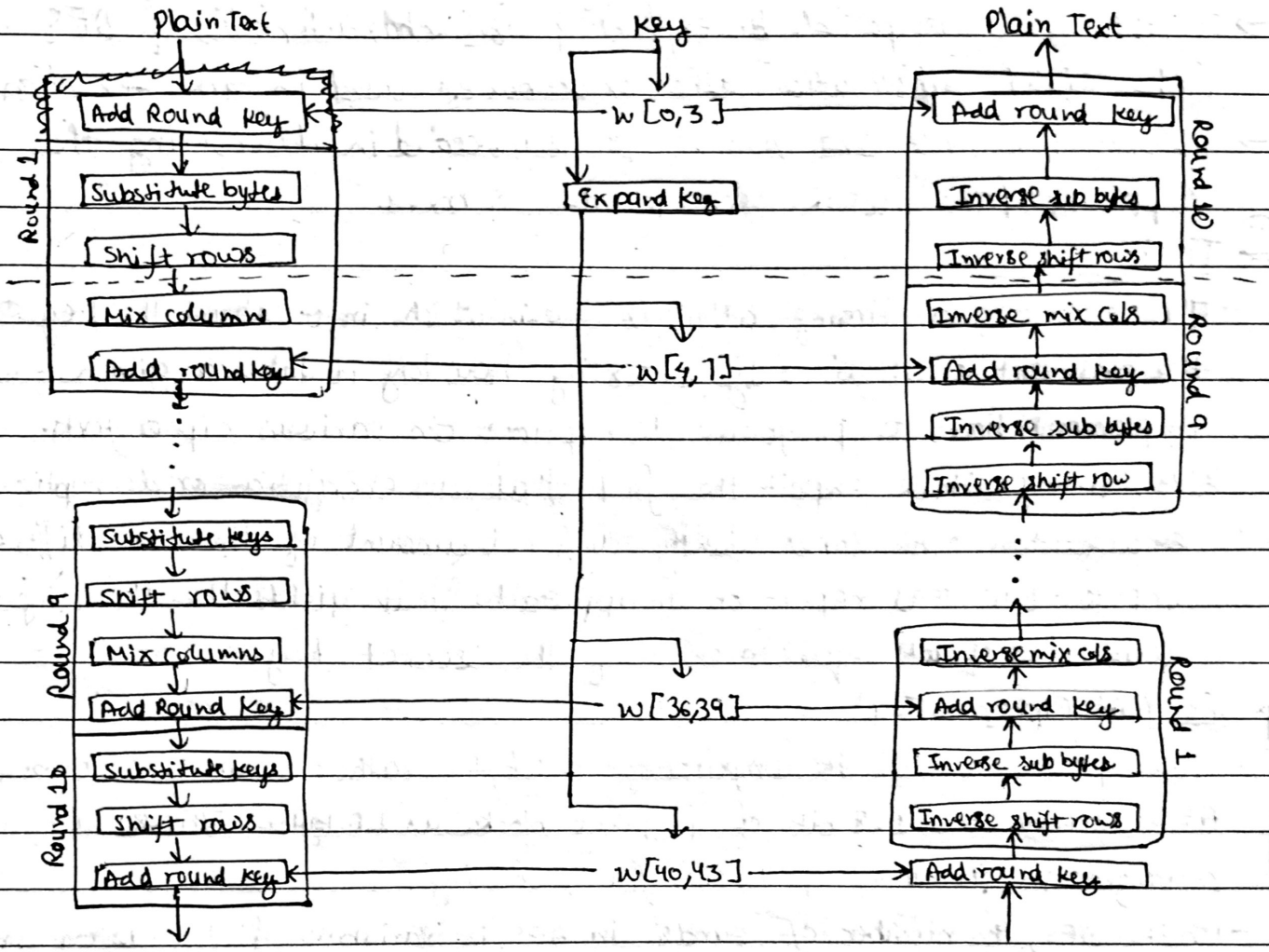
- AES performs all its computations on bytes rather than bits. Hence AES treats the 128 bits of plaintext block as 16 bytes. These 16 bytes are arranged in 4 columns & 4 rows for processing.
- Unlike DES, the number of rounds in AES is variable & depends on length of key.
- AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys & 14 rounds for 256-bit keys
- AES security is assured only if correctly implemented & good key management is employed.
- The key that is provided as input is expanded into an array of forty-four 32-bit words, w[i]. Four distinct words (128 bits) serve as a round key for each round.
- Four different stages are performed on each round, one of permutation & three of substitution:

(i) Substitute bytes: Uses an S-box to perform a byte-by-byte transformation
(ii) Shift Rows: A simple permutation
(iii) MixColumns: A substitution that makes use of arithmetic over $GF(2^8)$
(iv) AddRoundKey: A simple bitwise XOR of current block with a portion of the expanded key.

Punk

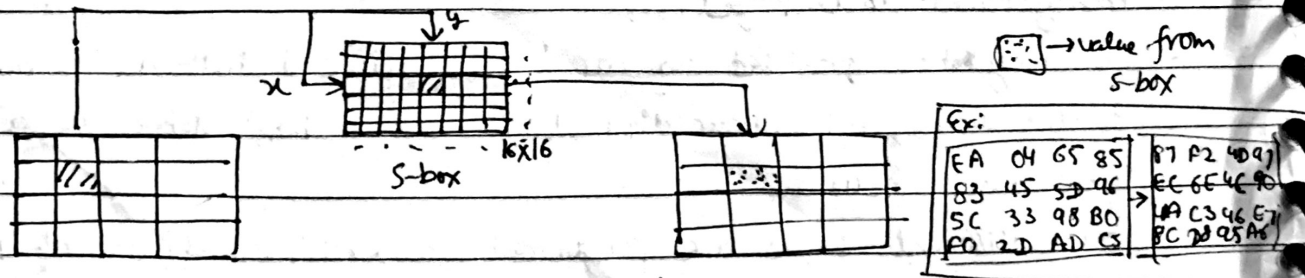
- Each stage is easily reversible. The decryption algorithm makes use of expanded key in reverse order.
- The final round of both encryption & decryption consists of only three stages which is required to make cipher reversible.

Date __/__/____



(a) Encryption (b) Decryption

- Substitute bytes Transformation : Using S-box (16 X 16 matrix)



- Shift Rows Transformation : For encryption, 1st row same
  2nd row shift left by 1-byte
  3rd row shift left by 2-byte
  4th row shift left by 3-byte
  } circular shift

Punk

Ex:-

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

Shift rows →

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Date __/__/____

- **MixColumns Transformation** : Operates on each column individually. Each byte is mapped into a new value that is a function of all four bytes in that matrix. Transformation can be defined by following matrix transformation on state:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

Individual additions & multiplications are performed in $GF(2^8)$

$$\begin{cases} s'_{0,j} = (2 \cdot s_{0,j}) \oplus (3 \cdot s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\ s'_{1,j} = (s_{0,j}) \oplus (2 \cdot s_{1,j}) \oplus (3 \cdot s_{2,j}) \oplus s_{3,j} \\ s'_{2,j} = (s_{0,j}) \oplus s_{1,j} \oplus (2 \cdot s_{2,j}) \oplus (3 \cdot s_{3,j}) \\ s'_{3,j} = (3 \cdot s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \cdot s_{3,j}) \end{cases}$$

Ex:-

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

- **Add Round Key Transformation**: 128 bits of state are bitwise XORed with 128 bits of round key.

Ex:-

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$ Round Key

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

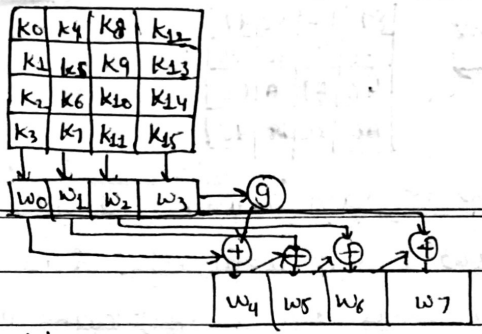
$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D2

- **AES Key Expansion**

```
KeyExpansion (byte key[16], word w[44])
{
    word temp;
    for (i=0; i<4; i++)  w[i] = (Key[4*i], Key[4*i+1], key[4*i+2], key[4*i+3]);
    for (i=4; i<44; i++)
    {
        temp = w[i-1];
        if (i mod 4 = 0)  temp = subWord (RotWord (temp)) ⊕ Rcon[i/4];
        w[i] = w[i-4] = temp
    }
}
```

Punk



RotWord → rotate word left by 1
SubWord → substitute word

Date __/__/____

Ex:- 10th Round Key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then first 4 bytes (1st column) of round key for round 9 are calculated as follows

i(decimal)	temp	After rotword	After subword	Rcon(9)	After XOR with RCON	w[i-4]	w[i]=temp ⊕ w[i-4]
36	7F8D292F	8D292F7F	5DA515D2	1B000000	46A515D2	EAD27321	AC7766F3

** Asymmetric cryptography also known as public key encryption, uses public & private keys to encrypt & decrypt data. Either of the keys can be used to encrypt the message & opposite key used to decrypt the message.

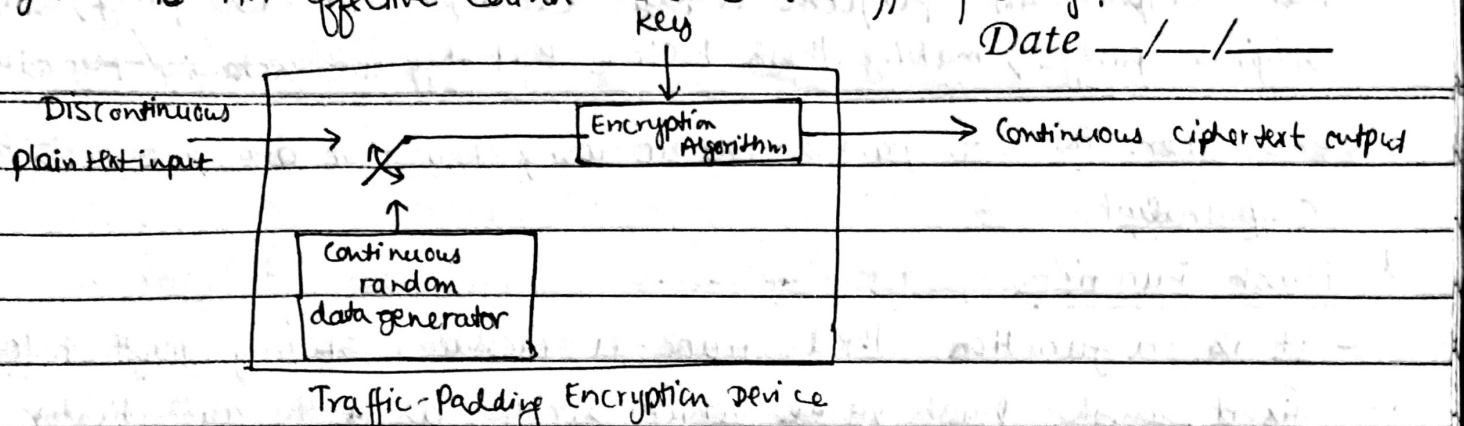
* Traffic Confidentiality

- The following types of information can be derived from a traffic analysis attack:
  (i) Identities of partners
  (ii) How frequently the partners are communicating
  (iii) Message pattern, message length, or quantity of messages that suggest important information is being exchanged
  (iv) The events that correlate with special conversation between particular partners
- Another concern related to traffic is the use of traffic patterns to create a covert channel. A covert channel is a means of communication in a fashion unintended by the designers of the communication facility. Typically, the channel is used to transfer information in a way that violates a security policy.
- Link Encryption Approach
  → with use of link encryption, network-layer headers are encrypted, reducing opportunity for traffic analysis. However, it is still possible

Punk

in those circumstances for an attacker to assess amount of traffic on network & to observe amount of traffic entering & leaving each end system. An effective countermeasure is traffic padding.

Date __/__/____



Traffic-Padding Encryption Device

- End-to-End Encryption Approach

→ If only end-to-end encryption is employed, then measures available to defender are more limited. One technique is to pad out data units to a uniform length either at transport or application level.

→ In addition, null messages can be inserted randomly into the stream. These tactics deny an opponent knowledge about amount of data exchanged between end users & obscure the underlying traffic pattern.

* Cryptoanalysis

- It refers to the study of ciphers, ciphertext or cryptosystems with a view to finding weakness in them that will permit retrieval of the plaintext from the ciphertext, without necessarities.
- Cryptographers develops algorithms to encrypt sensitive information & protect critical information. Ex:- In military
- Knowing the key or algorithm, below are some of the most common types of attacks:

(i) Known-plaintext analysis: with this procedure, the cryptoanalyst has knowledge of a portion of the plaintext from ciphertext

(ii) Chosen-plaintext analysis: The cryptoanalyst is able to have any plaintext encrypted with a key & obtain the resulting ciphertext, but the key can't be analyzed itself. Here entire ciphertext is compared with original plaintext (to obtain ciphertext).

(iii) Ciphertext only analysis: The cryptoanalyst has no knowledge of plaintext & must work only from the ciphertext. This requires accurate guesswork as to how a message could be worded.

(iv) Man-in-the middle attack: It differs from other by involving tricking individuals into surrending their keys.

Punk

Here cryptanalyst places him in the communication channel b/w two parties who wish to exchange their keys for secure communication. Then cryptanalyst perform a key exchange with each party, with the original party, making them believe that they ~~are exchanging key with~~ Date each other. The two parties end up using keys that are known to the cryptanalyst.

* Hash Function

- It is a function that maps a message of any length into a fixed length hash value, which serves as the authenticator
- It is useful in almost all information security applications.
- It is a mathematical function that converts a numerical input value into another compressed numerical value. Value returned by a hash function is called msg digest.

Punk

Date __/__/____

Unit-II

* Linear Cryptanalysis and Differential Cryptanalysis

- Differential cryptanalysis is a form of cryptanalysis applicable primarily to block cipher. It is the study of how differences in two inputs can affect the output.
- Linear cryptanalysis is based on finding linear approximations to describe the transformations performed in the cipher.
- Differential Cryptanalysis Attack
  → The idea behind it to observe behaviour of pairs of text blocks evolving along each round of cipher, instead of observing the evolution of a single text block.
- Linear Cryptanalysis Attack
  → For a cipher with n-bit plaintext & cipher text blocks & an m-bit key, let the plaintext block be labeled $P[1], \ldots P[n]$, the cipher text block $C[1], \ldots C[n]$, & the key $K[1], \ldots, K[m]$. Then define

$$A[i,j,\ldots,k] = A[i] \oplus A[j] \oplus \ldots \oplus A[k]$$

The objective of linear cryptanalysis is to find an effective linear equation of the form:

$$P[\alpha_1, \alpha_2, \ldots, \alpha_a] \oplus C[\beta_1, \beta_2, \ldots, \beta_b] = K[\gamma_1, \gamma_2, \ldots, \gamma_c]$$

that holds with probability $p \neq 0.5$

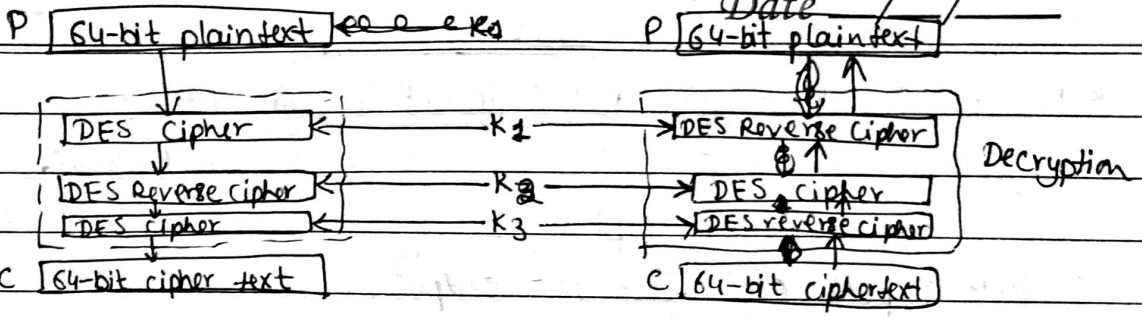
* Triple DES

- There are two variants of Triple DES known as 3-key Triple DES (3TDES) & 2-key Triple DES (2TDES)
- 3TDES
  → Before using 3TDES, user first generate and distribute a 3TDES key K, which consists of three different DES keys K1, K2 & K3. This means that the actual 3TDES key has a length $3 \times 56 = 168$ bits.
  → The encryption-decryption process is as follows:
  (i) Encrypt the plain text blocks using single DES with key K1.
  (ii) Now decrypt the output of step (i) using single DES with key K2.
  (iii) Finally, encrypt the output of step (ii) using single DES with key K3.

Punk

(iv) The output of step (iii) is the ciphertext.

(v) Decryption of a ciphertext is a reverse process. User first decrypt using K_3, then encrypt with K_2, & finally decrypt with K_1.



Encryption: $C = E(K_3, D(K_2, E(K_1, P)))$

Decryption: $P = D(K_1, E(K_2, D(K_3, C)))$

* TDES more secure than DES but slower than DES.

- 2 TDES

Encryption: $C = E(K_1, D(K_2, E(K_1, P)))$

Decryption: $P = D(K_1, E(K_2, D(K_1, C)))$

Expected running time of the attack is of the order of

$$(2^{56})\ \frac{2^{64}}{n} = 2^{120 - \log_2 N}$$

* Evaluation of AES

- The three criteria were as follows: (NIST 1997)

(i) Security :- Refers to effort required to cryptanalyze an algorithm

(A) Actual security ⇒ Compared to other submitted algorithm

(B) Randomness ⇒ Extent to which O/P indistinguishable from a random permutation of I/P block

(C) Soundness ⇒ of Mathematical basis for algorithm security

(D) Other security factors

(ii) Cost :- High computation efficiency

(A) Licensing Requirements → worldwide, non-exclusive, royalty-free basis

(B) Computational Efficiency

(C) Memory Requirements

(iii) Algorithm & implementation characteristics :- Includes variety of considerations, including flexibility, hardware & software suitability etc.

(A) Flexibility (B) Hardware & Software Suitability

(C) Simplicity

Punk

Date __/__/____

- Final NIST Evaluation of Rijndael (AES) (2000):

(i) General Security ⇒ No known security attacks

(ii) Software Implementation ⇒ Performs encryption & decryption very well across variety of platforms

(iii) Restricted-Space Environments ⇒ very well suited for restricted space environments where either encryption or decryption is implemented (not both)

(iv) Hardware Implementations ⇒ Highest throughput of any of the finalists for feedback modes & second highest for non-feedback modes.

(v) Attacks on Implementation ⇒ Operations used by Rijndael are easiest to defend against power & timing attacks.

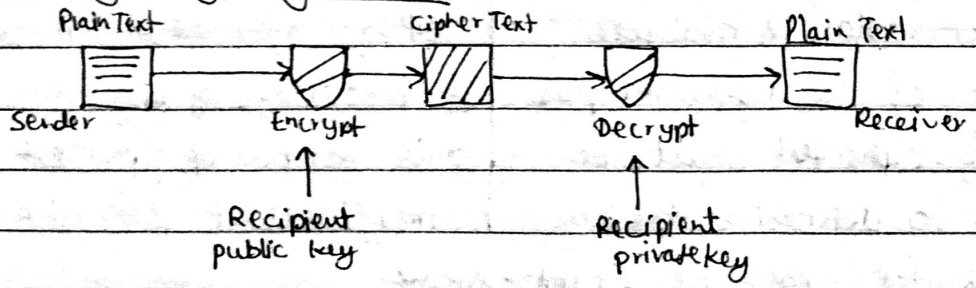
(vi) Encryption vs Decryption ⇒ Encryption & decryption func^n differ. Implementation of both encryption & decryption takes about 60% more space than the implementation of encryption alone.

(vii) Key Agility ⇒ Rijndel supports on-the-fly subkey computation for encryption. This places a slight resource burden on key agility.

(viii) Other versability & flexibility ⇒ Rijndel fully supports block sizes & key sizes of 128 bits, 192 bits & 256 bits in any combination

(ix) Potential for Instruction-Level Parallelism ⇒ Rijndel has an excellent potential for parallelism for a single block encryption

* Public Key Cryptography (Asymmetric Encryption)



- Properties of public key encryption schemes:

(i) Different keys are used for encryption & decryption. ~~This is a property~~

(ii) Each receiver possess a unique decryption key called private key

(iii) Receiver needs to publish an encryption key, called public key.

(iv) Some assurance of authenticaty of a public key is needed in this scheme to avoid spoofing by adversary as the receiver. Generally, this type

Punk

of cryptosystem involves trusted third party which certifies that a particular public key belongs to a specific person or entity only.

(v) Encryption algorithm is complex enough to prohibit attacker from deducing the plaintext from ciphertext & the encryption (public) Date: __/__/____

(vi) Though private & public keys are related mathematically, it is not feasible to calculate the private key from the public key.

* - Asymmetric encryption can be used for confidentiality, authentication, or both.

- The most widely used ~~prime~~ public-key cryptosystem is RSA. The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

- A public-key encryption scheme has six ingredients :-

(i) Plain Text (ii) Encryption Algorithm (iii) Private & Public Keys (iv) Cipher Text (v) Decryption algorithm

Conventional Encryption	Public-Key Encryption
Needed to work:	Needed to work:
1. Same algo with same key used for encryption & decryption	1. One algo for encryption & decryption with a pair of keys.
2. Sender & receiver must share algo & key	2. Sender & receiver must each have one of the matched pair of keys (not same one).
Needed for security:	Needed for security:
1. key must be kept secret	1. One of the keys must be kept secret
2. It must be impossible or atleast impractical to decipher a message if no other information is available	2. It must be impossible or at least impractical to decipher a message if no other information is available
3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key	3. Knowledge of algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine other key.

- Applications for Public-Key Cryptosystems

(i) Encryption / Decryption

(ii) Digital Signature

(iii) Key Exchange

- Ex of Public-key cryptosystems are RSA, Elliptic Curve, Diffie-Hellman & DSS.

Punk

* The RSA Algorithm (Imp)

- Encryption & decryption use modular exponential.
- ~~Discovered~~ Invented by Ron Rivest, Adi Shamir & Len Adleman & hence RSA

Date __/__/____

- **Algorithm:-**

Generate RSA modulus:
(i) choose two large primes, p & q such that $p \neq q$
(ii) Calculate $n = p * q$ (n is RSA modulus)

Find derived Number(e):
(iii) Choose number e. (public key) such that e is not factor of $(p-1)*(q-1)$ or e & $(p-1)(q-1)$ are co-prime & $1 < e < (p-1)(q-1)$

Form public key:
(iv) pair (n, e) form RSA public key & is made public

Generate private key:
(v) Choose number d (private key) such that ~~$(d*e) \bmod (p-1)(q-1)$~~
$(d*e) \bmod (p-1)(q-1) = 1$ or $ed = 1 \bmod (p-1)(q-1)$

(vi) (Encryption) Cipher Text, $C = (P)^e \bmod n$ P = plain Text

(vii) (Decryption) Plain Text, $P = (C)^d \bmod n$

- Ex:- ① Say $p = 7$ & $q = 11$

② $n = p * q = 77$

③ $(p-1)*(q-1) = 6 \times 10 = 60$

④ Let $e = 13$

⑤ $(D \times E) \bmod (p-1)(q-1) = 1 \Rightarrow (D \times 13) \bmod 60 = 1$

$D = 37$

⑥ $C = (P)^{13} \bmod 77$
let $P = 5$
$C = (5)^{13} \bmod 77$
$= 26$

⑦ $P = (C)^d \bmod n$
$P = (26)^{37} \bmod 77$
$= 5$

- **Security of RSA:**

→ Four approaches for attacking: (i) Brute force (ii) Mathematical attacks (iii) Timing Attacks (iv) Chosen cipher text attacks

→ Although timing attacks is a serious threat, there are simple counter measures such as constant exponentiation time, random delay & blinding (multiply cipher text by random no. before performing expanentiation)

→ Factoring problem for mathematical attacks.
If $e < n$ & $d < n^{1/4}$, d can be determined easily

Punk

~~* Block Ciphers & Stream Ciphers (Cover before DES)~~

* Block Cipher Modes of Operation

Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using same key	- Secure transmission of single values (eg an encryption key)
Cipher Block Chaining (CBC)	I/P to encryption algo is XOR of next 64 bits of plain text & the preceding 64 bits of cipher text	- General purpose block-oriented transmission - Authentication
Cipher Feedback (CFB)	I/P is process j bits at a time. Preceding ciphertext used as I/P to encryption algo to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext	- General-purpose stream oriented transmission - Authentication
Output Feedback (OFB)	Similar to CFB, except that the I/P to encryption algo is preceding DES O/P	- Stream oriented transmission over noisy channel (eg satellite communication)
Counter (CTR)	Each block of plaintext is XORed with an encryption counter. The counter is incremented for each subsequent block	- General purpose block oriented transmission - Useful for high-speed requirements

* Key Management

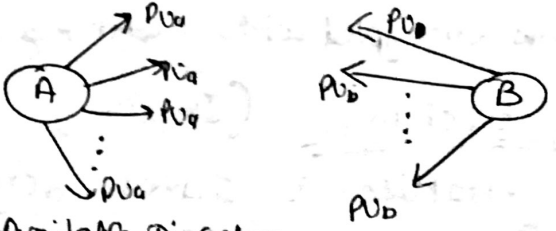
- Public key encryption schemes are secure only if the authenticity of the public key is assured. A public-key certificate scheme provides the necessary security.
- A simple public-key algorithm is Diffie-Hellman key exchange. This protocol enables two users to establish a secret key using a public-key scheme based on discrete logarithms. The protocol is secure only if the authenticity of two participants can be established.
- Elliptic curve arithmetic can be used to develop a variety of elliptic curve cryptography (ECC) schemes, including key exchange, encryption, & digital signature.
- For purposes of ECC, Elliptic curve arithmetic involves the use of an elliptic curve equation defined over a finite field. The coefficients & variables in the equation are elements of a finite field. Schemes using Z_p & $GF(2^m)$ have been developed.
- Distribution of public keys

→ four schemes

① Public Announcement of Public Keys:

Problem: Anyone can forge such a public announcement

Punk

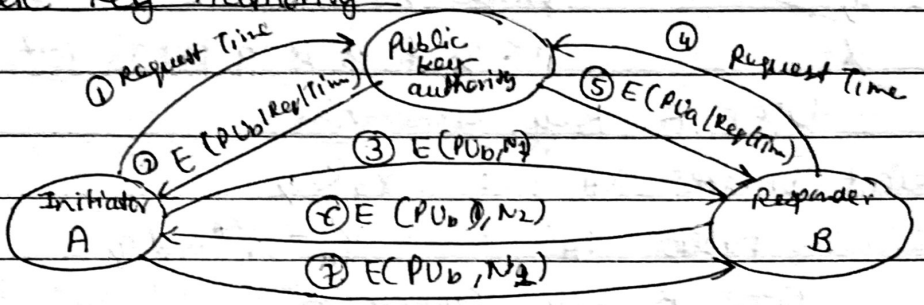


Date __/__/____

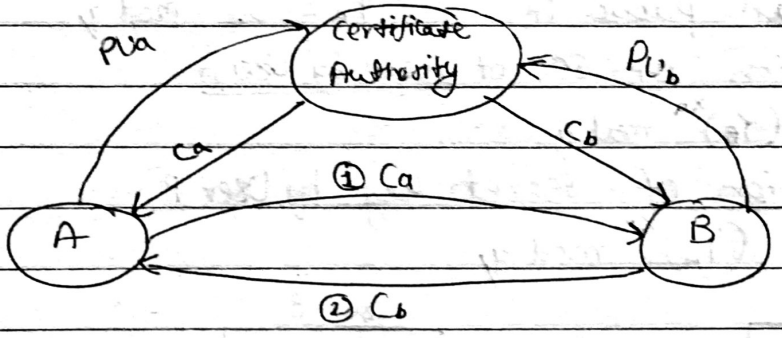
(ii) Publicly Available Directory

Problem: Tamper with records available kept by authority

(iii) Public Key Authority

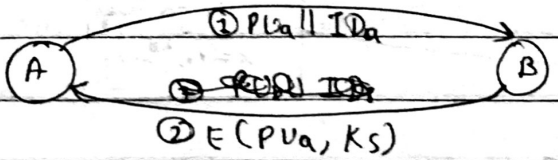


(iv) Public-Key Certificates

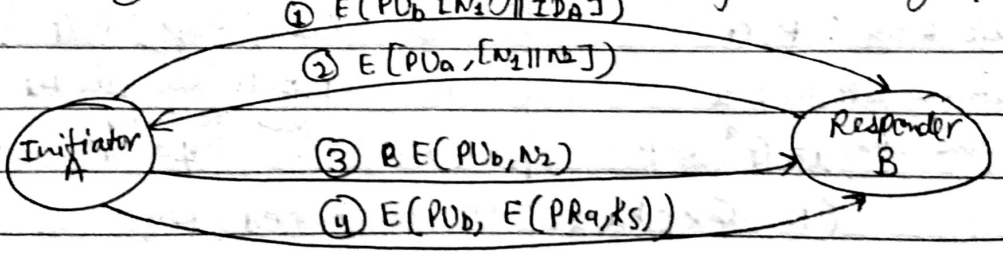


- Distribution of Secret Keys Using Public Key Cryptography

(i) Simple Secret Key Distribution problem: Man-in-the-Middle Attack



(ii) Secret Key Distribution with Confidentiality & Exchange



(iii) A hybrid scheme : Retains use of Key distribution center (KDC) that shared a secret master key with each user of

Punk

distributes secret session keys encrypted with master key.

* Diffie-Hellman Key Exchange (Imp)

- It is being used to establish a shared secret that can be

Date __/__/____

used for secret communications while exchanging data over a public network using the elliptic curve to generate points & get the secret key using the parameters.

- Algo:

(1) Global public Elements

q — prime no

α — $\alpha < q$ & α is a prime root of q

(2) User A Key Generation

(i) select private X_A : $X_A < q$

(ii) Calculate public Y_A : $Y_A = \alpha^{X_A} \bmod q$

(3) User B Key Generation

(i) select private X_B : $X_B < q$

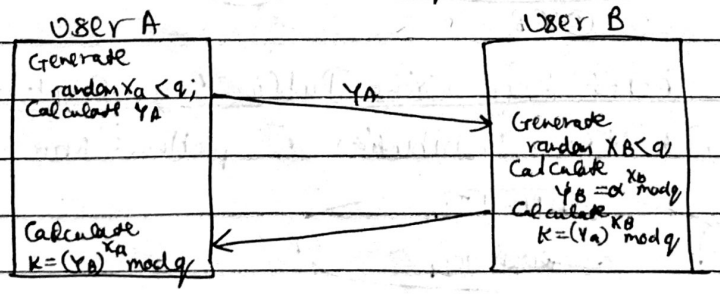
(ii) Calculate public Y_B : $Y_B = \alpha^{X_B} \bmod q$

(4) Calculation of secret key by User A

$K = (Y_B)^{X_A} \bmod q$

(5) Calculation of secret key by User B

$K = (Y_A)^{X_B} \bmod q$



- Step by Step Explanation

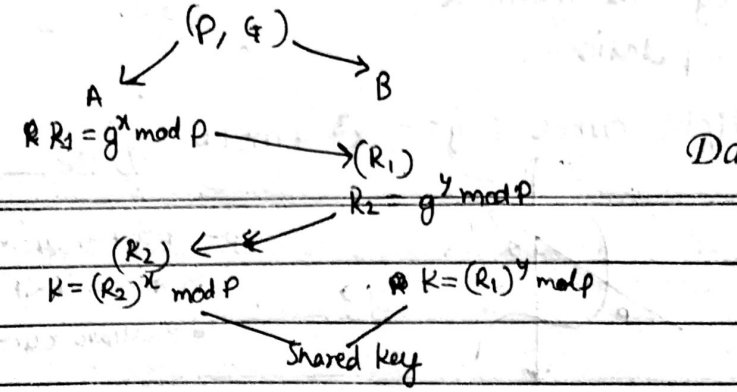
A	B
① public keys : P, G	① Public keys : P, G
② Private key selected : a	② Private key selected : b
③ key generated, $x = G^a \bmod P$	③ key generated $y = G^b \bmod P$
④ Exchange of generated keys	
⑤ key received = y	⑤ key received = x
⑥ Generate secret key $k_a = y^a \bmod P$	⑥ Generate secret key $k_b = x^b \bmod P$

$k_a = k_b$

Users now have a symmetric secret key to encrypt

Punk

Process:



Date __/__/____

Ex:-

A

B

$g = 7, p = 23$

$x = 3$

$y = 6$

① $R_1 = 7^3 \bmod 23$

$= 21$

$R_2 = 7^6 \bmod 23 = 4$

$= 4$

$K = (4)^3 \bmod 23$

$= 18$

$K = (21)^6 \bmod 23$

$= 18$

Shared key

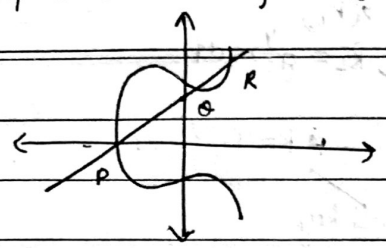
* Elliptic Curve Cryptography (ECC)

- ECC is a term used to describe a suite of cryptographic tools and protocols whose security is based on special versions of the discrete logarithm problem & doesn't use numbers modulo p.
- ECC is based on sets of numbers that are ~~based~~ associated with mathematical objects called elliptic curves. There are rules for adding & computing multiples of these numbers, just as there are for numbers modulo p.
- ECC includes a variant of many cryptographic schemes that were initially designed for modular numbers such as ElGamal encryption and Digital Signature Algorithm.
- It is believed that the discrete logarithms problems is much harder when applied to points on an elliptic curves. This prompts switching from numbers modulo p to points on an elliptic curve. Also an equivalent security level can be obtained with shorter keys if we use elliptic curve-based variants.
- The shorter keys results in two benefits:

Punk

(i) Ease of key management
(ii) Efficient computation

- Equation of elliptic curve : $y^2 = x^3 + ax + b$ Date __/__/____



p -> point on curve
n -> max limit (this should be prime no)
E -> Elliptic curve

- ~~Encryption~~: ECC Diffie-Hellman Key Exchange

$E_q(a, b)$ → elliptic curve with parameters a, b & q
q is prime or an integer of form 2^m
G → point on elliptic curve whose order is large value n

G

A
Select private n_A
$n_A < n$
Calculate public $P_A = n_A \times G$

B
Select private n_B
Calculate $P_B = n_B \times G$

(P_A) →
← (P_B)

$K = n_A \times P_B$
$K = n_B \times P_A$

Shared Key

Ex:- $E_p(0, -4) : y^2 = x^3 - 4$
$p = 211$ & $G = (2, 2)$

$n_A = 121$
$P_A = 121(2, 2) = (115, 48)$

$n_B = 203$
$P_B = 203(2, 2) = (130, 203)$

(115, 48) →
← (130, 203)

$K = 121(130, 203) = (161, 69)$
$K = 203(115, 48) = (161, 69)$

Shared Key

∴ The security of ECC depends on how difficult it is to determine k given kP & P. This is referred to as the elliptic curve logarithm problem. The fastest known technique for taking the elliptic curve logarithm is known as Pollard Rho method.

Punk

Date __/__/____

*Message Authentication

- Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that the data received are exactly as sent by and that the purported identity of the sender is valid.
- Symmetric encryption provides authentication among those who share the secret key. Encryption of a message by the sender's private key also provides a form of authentication.
- The two most common cryptographic techniques for message authentication are a message authentication code (MAC) & a secure hash function.
- A MAC is an algorithm that requires use of secret key. A MAC takes a variable-length message & a secret key as input & produces an authentication code. A recipient in possession of the secret key can generate an authentication code to verify integrity of message.
- A hash function maps a variable-length message into a fixed length hash value, or message digest. For message authentication, a secure hash function must be obtained in some fashion with a secret key.
- Authentication Requirements

→ In context of communications across a network, the following attacks can be identified:

(i) Disclosure: Release of message contents to any person or process not possessing the appropriate cryptographic key.

(ii) Traffic Analysis: Discovery of pattern of traffic between parties

(iii) Masquerade: Insertion of messages into the network from a fraudlent source.

(iv) Content Modification: Changes to the contents of a message, including insertion, deletion, ~~reordering~~ transposition & modification

(v) Timing Modification: Delay or replay of messages

(vi) Sequence Modification: Any modification to a sequence of messages between parties, including insertion, deletion, & reordering

(vii) Source Repudiation: Denial of transmission of message by source

(viii) Destination Repudiation: Denial of receipt of message by destination

→ Message authentication is a procedure to verify that

Punk

received messages come from the alleged source & have not been altered
→ Message authentication may also verify sequencing & timeliness.
→ A digital signature is an authentication technique that also includes measures to counter repudiation by the source. Date __/__/____

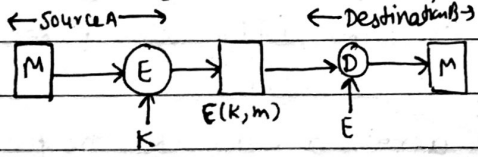
– Authentication Functions (Imp)

→ Any msg authentication or digital signature mechanism has two levels of functionality.
At lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message.
This lower-level func^n is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

→ Types of function that may be used to produce an authenticator:
(i) Message Encryption: Ciphertext of entire message serves as its authenticator
(ii) Message Authentication Code: A function of the message and a secret key that produces a fixed length value that serves as the authenticator
(iii) Hash function: A function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

→ Message Encryption:

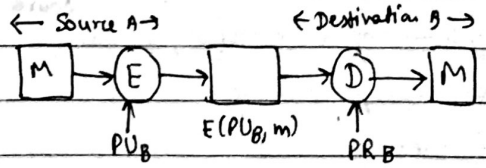
(a)



⇒ Does not provide signature
[↳ Receiver could forge message
↳ sender could deny message]

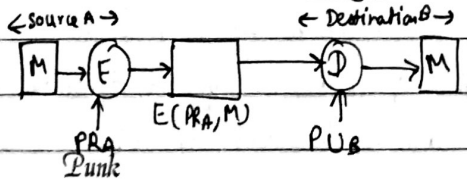
Symmetric Encryption: Confidentiality & authentication
A→B: E(K, M)
⇒ provides confidentiality
[↳ only A & B share K]
⇒ provides a degree of authentication
[↳ could come only from A
↳ Has not been altered in transit
↳ Requires some formatting / redudancy]

(b) Public-key Encryption: Confidentiality



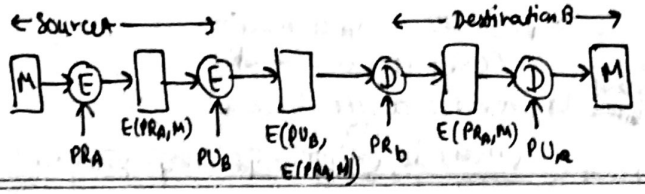
A→B: E(PUB, M)
⇒ provides confidentiality
[only B has PRB to decrypt]
⇒ provides no authentication
[↳ Any party could use UB to encrypt message and claim to be A]

(c) Public-Key Encryption: Authentication & Signature



A→B: E(PRA, M)
⇒ provides authentication & signature
[↳ only A has PRA to encrypt
↳ Has not been altered in transit
↳ Requires some formatting / redudancy
↳ Any party can use PUB to verify signature]

ⓓ Public-key Encryption: Confidentiality, Authentication & Signature



A → B: E(PU_B, E(PR_A, M))

⇒ provides confidentiality because of PU_b

⇒ provides authentication & signature because of PR_A

Date __/__/____

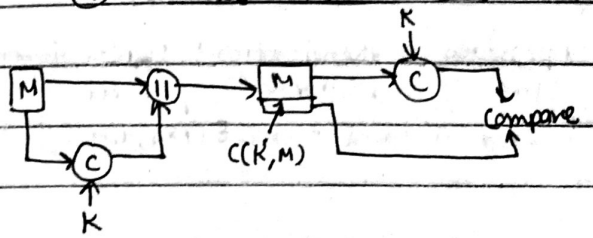
→ Message Authentication Code (MAC)/ Cryptographic checksum:

⇒ when A has a message to send to B, it calculates MAC as a function of the message & key : MAC = C(K, M) & transmitted with message.

⇒ The received MAC is compared to calculated MAC to check if message tempered or not.

⇒ Uses:

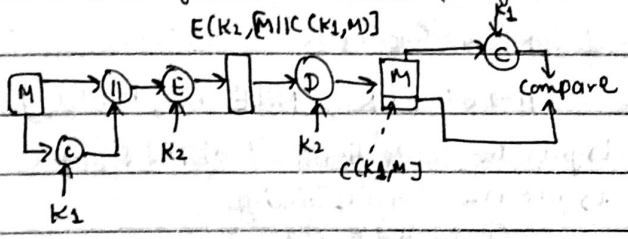
ⓐ Message Authentication



A → B: M || C(K, M)

↳ provides authentication (only A & B share K)

ⓑ Message authentication & confidentiality; authentication tied to plaintext

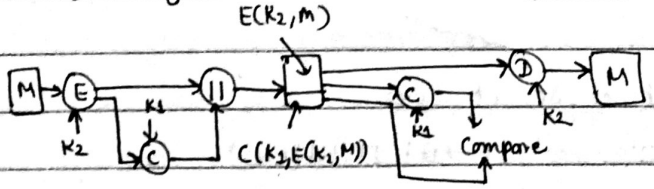


A → B: E(K2, [M||C(K1,M)])

↳ provides authentication (only A & B share K1)

↳ provides confidentiality (only A & B share K2)

ⓒ Message authentication & confidentiality; authentication tied to plaintext



A → B: E(K2,M) || C(K1, E(K2,M))

↳ provides authentication (using K1)

↳ provides confidentiality (using K2)

→ Hash Function:

⇒ Similar to MAC, i.e., one way hash function takes variable length input & a fixed-length output, referred to as Hash code [H(M)].

⇒ Unlike MAC, a hash code does not use a key but is a function only of the input message. The hash code is also referred to as message digest or hash value.

⇒ Uses:

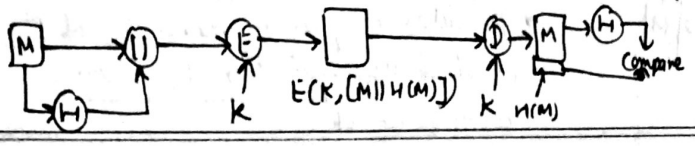
Punk

(A) Encrypt message plus Hash code
(Symmetric Encryption)

$A \rightarrow B: E(K, [M \| H(M)])$

↳ provides confidentiality (only A & B share K)

↳ provides authentication (H(M) is cryptographically protected)



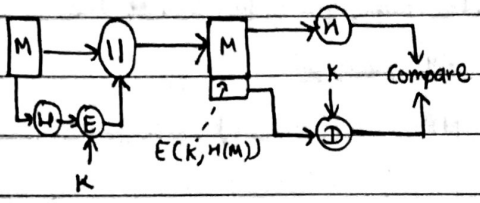
Date

(B) Encrypt hash code - shared secret key
(Symmetric Encryption)

$A \rightarrow B: M \| E(K, H(M))$

↳ provides authentication

(H(M) is cryptographically protected)

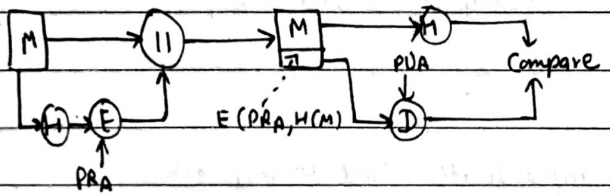


(C) Encrypt Hash Code - Sender's Private Key
(Public key Encryption)

$A \rightarrow B: M \| E(PR_A, H(M))$

↳ provides authentication & digital signature

(H(M) is cryptographically protected
only A could create $E(PR_A, H(M))$)

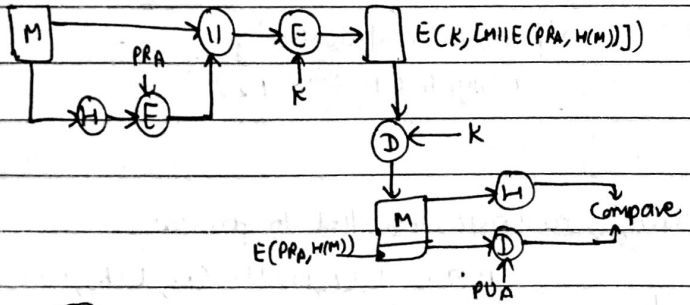


(D) Encrypt Hash Code - Sender Private key & Symmetric Encryption
(Public key + Symmetric Encryption)

$A \rightarrow B: E(K, [M \| E(PR_A, H(M))])$

↳ provides authentication & digital signature

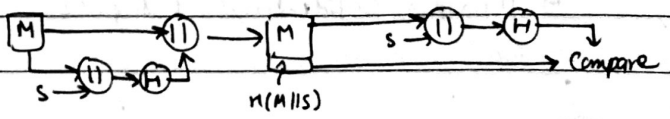
↳ provides confidentiality (only A & B share K)



(E) Compute Hash Code of Message Plus Secret Value

$A \rightarrow B: M \| H(M \| S)$

↳ provides authentication (only A & B share S)

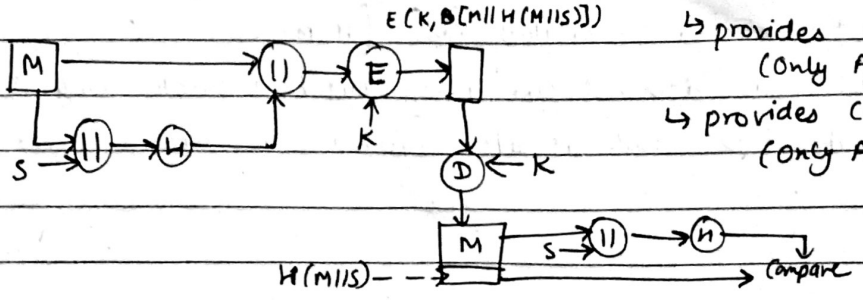


(F) Encrypt result of (E)

$A \rightarrow B: E(K, [M \| H(M \| S)])$

↳ provides authentication (only A & B share S)

↳ provides confidentiality (only A & B share K)



Punk

* Digital Signatures

- A digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.

Date __/__/____

- Digital signature is a cryptographic value that is calculated from the data & a secret key known only to the ~~sender~~ signer.
- Signing a hash is more efficient that signing entire data.
- Uses: (i) Message Authentication
  (ii) Data Integrity
  (iii) Non-repudiation
- Encryption with Digital Signature
  → By combining digital signatures with encryption scheme, message authentication & repudiation can be assured.
  → There are two possibilites : sign-then-encrypt & encrypt-then-sign
  → Crypto ~~based~~ system based on sign-then-encrypt can be exploited by receiver to spoof identity of sender & send that data to third party.
  → Process of encrypt-then-sign is more reliable & widely adopted as first signature is verified using sender's public key. After ensuring validity of the signature, then retrieve the data through encryption using private key.

* Cyber Forensics

- Also known as computer forensics
- It's the application of investigation & analysies techniques to gather & preserve evidence from a particular computer device in a way, i.e., suitable for presentation in a court of law
- The goal of computer forensics is the perform a structured investigation while maintaining a documented chain of evidence to find out exactly what happened on a computing device & who was responsible for it.
- Ethical hacking is the key to strengthening network security & it's one of the most desired skill for any IT security professional. They have knowledge in problem-solving strategies for security breaches & can collect & analyze data to monitor & interpret weakness.

Punk

* Comparison chart for AES & DES

DES	AES
(i) Divides data block into 2 halves.	(i) Entire data block is processed as a single matrix. Date __/__/____
(ii) Work on feistal cipher structure	(ii) Work on substitution & permutation principle
(iii) plaintext is of 64 bits	(iii) plain text can be of 128, 192 & 256 bits
(iv) has smaller key compared to AES	(iv) has larger key size
(v) 16 rounds	(v) variable rounds 10 rounds for 128 bit 12 rounds for 192 bit 14 rounds for 256 bit
(vi) less secure	(vi) larger secret key & hence secure
(vii) Speed is slower	(vii) AES is faster

(Part of Unit I & II both)

* Additive Cipher

→ It is simplest code cipher. Mathematically it can be expressed

$$c = (p + a) \bmod 26$$

p ≡ position of letter, a ≡ key, c ≡ new position of letter

* Affine Cipher (Monoalphbetic Cipher)

→ Here each letter is mapped to its numeric equivalent, encrypted using simple mathematical function & converted back to letters

Mathematically represented as $(ap + b) \bmod 26$, b ≡ magnitude of shift

* Kerberos Protocol

→ It is a computer network authentication protocol which works on the basis of "tickets" to allow nodes communicate over a non secure network to prove their identity to one another in a secure manner

→ client-server model

→ symmetric key model

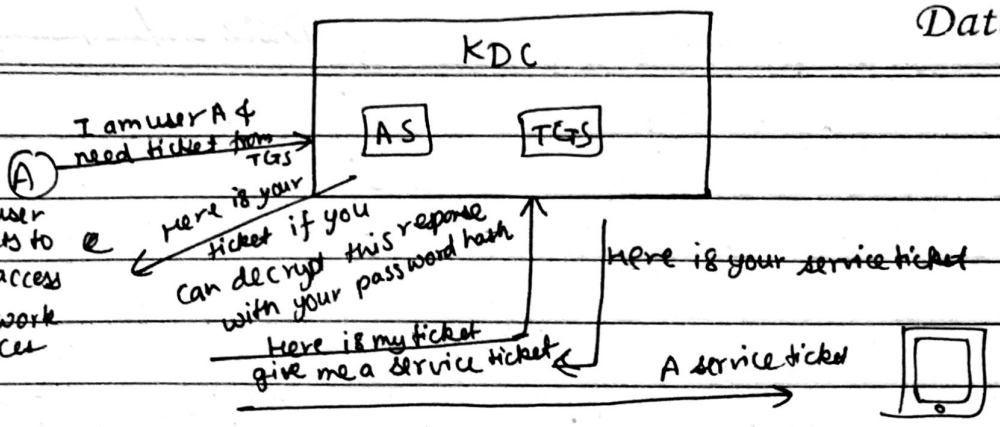
→ Requires a trusted 3rd party ⇒ key distribution centre (KDC)

(KDC) is database of secret key.

Punk

→ There are two types of KDC
(i) Authentication server (AS)
(ii) Ticket Granting Server (TGS)

Date __/__/____



* Vernam Cipher

→ Implemented by using 1-time pad (key)
→ Length of I/P cipher text equal to length of original plain text
→ Algo:
(i) Write each plaintext alphabet as number (A=0, Z=25) or (A=1, Z=26)
(ii) Same for 1-time pad
(iii) Add plaintext alphabet no. to one time pad no.
(iv) If sum ≥ 26, subtract 26 from it.
(v) Convert each no. of sum to alphabet

Ex:- H E L L O
7 4 11 11 14] Plain Text

X M C K L
23 12 2 10 11] key

Plain Text + key =

30	16	13	21	25
-26				
4	16	13	21	25
E	Q	N	V	Z] Cipher Text

Cipher Text - key =

4	16	13	21	25
-23	12	2	10	11
-19	4	11	11	14
+26				
7	4	11	11	14
H	E	L	L	O] Plain Text

Punk

Unit-III

Date __/__/____

* Buffer Flow Attack

- In computer security & programming, a buffer overflow is an anomaly where a program, while writing data to a buffer overruns the buffer's boundary & overwrites adjacent memory locations.

or

- A buffer overflow occurs when a program or process attempts to write more data to a fixed length block of memory or buffer, than the buffer is allocated to hold
- Buffers are areas of memory set aside to hold data, often while moving it from one section of program to another, or between programs.
- Buffer overflows are commonly associated with C-based languages, which do not perform any kind of array boundary checking. When a buffer overflow occurs in a program, it will crash or become unstable.
- A buffer overflow occurs when data written to a buffer also corrupts data values in memory addresses adjacent to destination buffer due to insufficient bounds checking. This can occur when copying data from one buffer to another without first checking that the data fits within the destination buffer.
- Heap Based: It is difficult to execute & least common of two. Attack an application by flooding the memory space reserved for a program.
- Stack Based: More common among attackers. It exploits application & program by using stack memory space used to store user input
- These can result in erratic program behaviour, including memory access errors, incorrect results, a crash or a breach of system security. Thus, Buffer flow attacks are the basis of many S/w vulnerabilities & can be maliciously exploited.
- The most reliable way to avoid or prevent buffer overflows is to use automatic protection at the language level.

* Distributed Denial of Services (DDOS)

- DDOS attacks make computer system inaccessible by flooding servers, networks or even end user systems with useless traffic so that legitimate users can no longer gain access to those resources. The flooding of incoming messages, connection requests or malformed packets to the target system forces it to slow down or even crash & shutdown.

Punk

- DOS attack is an attempt to prevent legitimate users of a service from using that service.
- In DDOS attack, an attacker is able to recruit a no. of hosts throughout the Internet to simultaneously or in a coordinated fashion launch an attack upon the target by sending useless packets to the target.

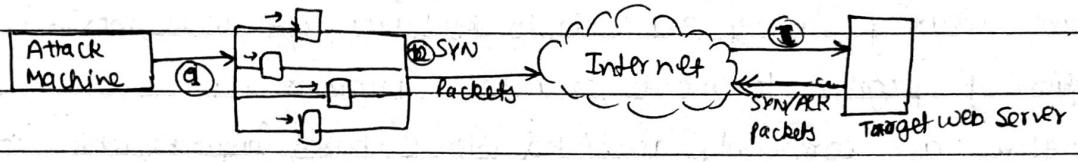
Date __/__/____

- Types of DDOS:

• A DDOS attack attempts to consume the target's resources so that it cannot provide service.

1. Internet host resource attack
2. Attack that consumes data transmission resources

1. Internet Resource Attack (SYN Flood Attack)

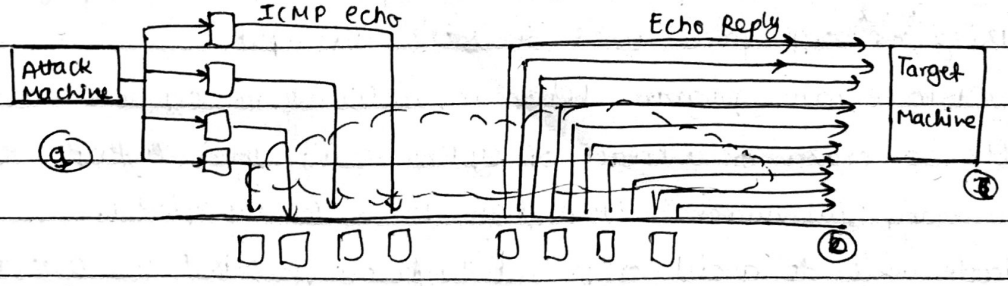


(a) The attacker takes control of multiple hosts over the Internet, instructing them to contact the target server.

(b) Slave hosts begin sending TCP/IP SYN (synchronize/initialization) packets with erroneous return IP address information, to target.

(c) Each SYN packet is request to open TCP connection.
For each such packet, the web server responds with an SYN/ACK packet. The web server becomes bogged down as more traffic floods in. It results that legitimate connections are denied while the victim machine is waiting to complete bogus connections.

2. Attack that consumes data transmission resources



(a) The attacker takes control of multiple hosts over internet, instructing them to send ICMP echo packets with target's spoofed IP address to group of hosts that acts as reflectors.

(b) Nodes at bounce site receive multiple spoofed requests & respond by sending echo reply packets to target site.

(c) The target's router is flooded with packets from bounce site leaving no data transmission capacity for legitmate traffic.

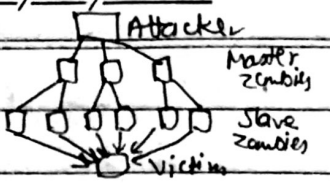
Punk

• Direct DDOS

→ The attacker implants zombie s/w on a no. of sites distributed throughout the Internet.

→ Direct DDOS involves two level of zombie machines: master zombies & slave zombies.

Date __/__/____



→ The attacker coordinates & triggers the master zombies, which in turn coordinate & trigger the slave zombies.

The two level of zombies makes it more difficult to trace attack back to its source.

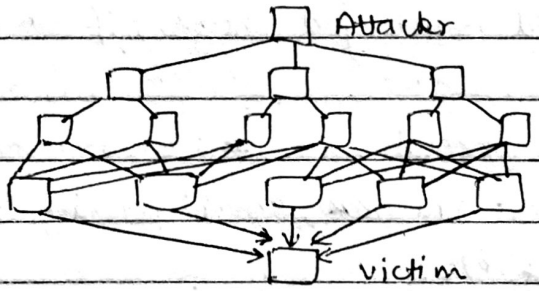
• Reflector DDOS

→ It adds another layer of machines known as reflectors

→ The slave zombies constructs packets requiring a response. These packets are sent to uninfected machines, i.e., reflectors.

→ The reflectors respond with packets directed at the target machine.

→ A reflector DDOS attack can easily involve more machines & more traffic than a direct DDOS attack & hence can be more damaging.



Master zombies
Slave zombies
Reflectors

— To construct an attacking network, the attacker first scans out a no. of vulnerable machine & infects them. Then, the zombie s/w that is installed in infected machine repeats the same scanning process, until a large distributed network of infected machines is created.

• Scanning strategies

→ Random - Randomly hits any machine in network producing high volume of Internet traffic.

→ Hit-List - Attacker compiles a list of vulnerable machines. This can be a slow process. Once the list is compiled attacker begins infecting machines on list. Each list machine on the list is given a portion of list to scan.

→ Topological - Info contained on an infected victim is used to find more hosts to scan

→ Local subnet - Host looks for target in its own local network. The host uses the subnet address to find other hosts in the network.

Punk

- DDOS Countermeasures:

① Attack Prevention & Preemption (Before attack)

Date __/__/____

- It enables victim to endure the attack attempts without denying service to legitimate clients.
- It includes enforcing policies for resource consumption & providing backup resources available on demand.
- It modifies systems & protocols on Internet to reduce the possibility of DDOS attacks.

② Attack Detection & Filtering (during the attack)

- It attempts to detect the attack as it begins & respond immediately. It minimizes the impact of attack on target.
- It involves looking for suspicious pattern of behaviour & filter out packets that are likely to be part of attack.

③ Attack Source Trace back & identification (After Attack)

- It attempts to identify the source of attack as a step to prevent future attacks. However, it doesn't yield results fast enough, if at all, to mitigate an ongoing task.

- These attacks can be minimized through some core information security practice like solid patch management practices, email phishing testing, user awareness, proactive n/w monitoring etc.

* Weak Authentication

- "Authentication" refers to the process of proving an identity to an application or system. It's the task of demonstrating that you are who you claim to be.
- weak authentication describes any scenario in which the strength of the authentication mechanism is relatively weak compared to value of the assets being protected.
- It also describes scenarios in which the authentication mechanism is flawed or vulnerable.
- For avoiding weak authentication, following practices should be done:
  - adopting a strong password policy
  - using two factor authentication
  - keeping authentication token secure.

* Substitution Boxes (S-Box)

- S-box is a basic component of symmetric key algorithm which performs substitution. In block cipher, they are typically used to depict rel^n ship b/w the key & ciphertext.

Punk

- S-box takes 'm' no. of input bits & transforms them into 'n' no. of bits where 'n' is not necessarily equal to 'm'.
- A mxn s-box can be implemented as lookup table with 2^m words of n bits each.
- The main criteria of a good s-box are:
  - It should have balanced components functions.
  - The non-linearity of its components func^n should be high.
  - The non-zero linear combinations of its component func^{n}s should be balanced & highly non linear.
  - It should have a high algebric degree.
- Ideal S-box properties
  - All linear combinations of s-box columns are bent.
  - All entries in the s-box XOR table are 0 or 2.
  - The set of weights of rows has a binomial distribution with mean $m/2$.
  - The set of weights of all pairs of rows has a binomial distribution with mean $m/2$.
  - The column each have hamming weight $2(n-1)$.
- Approaches of S-box design:
  - Random: Use some pseudorandom no. generation or some table of random digits to generate the enteries in the S-boxes.
  - Random with testing: Choose S-box entries randomly, then test the results against various criteria & throw away those that do not pass.
  - Human-made: Manual approach with only simple mathematics to support it.
  - Math-made: Generate S-boxes acc^n to mathematical principles. S-boxes can be constructed that offer proven security against linear & differential cryptanalysis, together with good diffusion.

Hash functions

- A hash func^n is a func^n which takes on input (or message) & returns a fixed size alphanumeric string known as hash value. A hash value 'h' is generated by a func^n 'H' as:

$h = H(M)$ where h = hash value (msg digest / checksum)
M = variable-length message
$H(M)$ = Hash func^n

- The hash value is appended to the msg at the source when the msg is assumed to be correct. The receiver authenticates the msg by recomputing the hash value. It acts as a 'fingerprint' of a file, msg or other block of data.

Punk

- A hash func^n must have the following properties:

(1.) H can be applied to a block of data of any size.

(2.) H produces fixed length output.

(3.) H(x) is relatively easy to compute for any given x. Date __/__/____

(4.) One way property - For any given value h, it is computationally infeasible to find x such that H(x) = h.

(5.) Weak collision Resistance - For any given block x, it is computationally infeasible to find $y \neq x$ such that H(x) = H(y).

(6.) Strong Collision Resistance - It is computationally infeasible to find any pair (x,y) such that H(x) = H(y), i.e., two different msgs can't have same hash values.

- The input (msg, file etc.) is viewed as a sequence of n-bit blocks. The I/P is processed one block at a time in an iterative fashion to produce n-bit hash func^n.

- One of the simplest hash func^n is the bit-by-bit XOR of every block

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

$C_i \equiv i^{th}$ bit of the hash code $m \equiv$ no. of n bit blocks in input

$b_{ij} \equiv i^{th}$ bit in j^{th} block $\oplus \equiv$ XOR operation

* Security of Hash Functions

- There are 3 main characteristics which are looked for in a hash func^n:

- Resistance to preimages: Given x, it shall be hard to find m such that h(m) = x.
- Resistance to second preimages: Given m & h(m), it shall be hard to find m' (distant from m) such that h(m) = h(m')
- Resistance to collisions: It shall be hard to find m & m' distinct from each other such that h(m) = h(m')

(1) Brute Force Attack

→ The strength of hash func^n against brute-force attacks depends solely on the length of the hash code produced by the algorithm.

→ For a hash code of length n, the level of efforts required are:

One way	2^n
weak collision resistance	2^n
strong collision resistance	$2^{n/2}$

(2) Cryptanalysis

→ Cryptanalysis attacks on hash func^n algo to exploit some property

Punk

of algorithm to perform some attack other than exhaustive search.

→ Secure Hash Func^n Structure

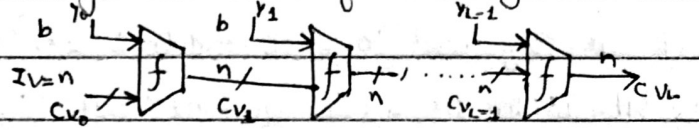
Hash func^n takes an I/P msg. & partitions it into L fixed sized blocks of b bits each. (If needed, final block is padded to b bits)

Date __/__/__

Hash also involves repeated use of compression func^n, f, that takes 2 inputs & produces n bit output [n bit I/P from previous step, chaining variable; n bit block]

At the start of hashing, the chaining variable has a fixed value that is specified as a part of the algorithm.

The final value of chaining variable is the hash value



CV_0 = IV = Initial n-bit value

$CV_i = f(CV_{i-1}, Y_{i-1})$

$H(M) = CV_L$

→ Cryptanalysis of hash func^{n}s focuses on the internal structure of f & is based on attempts to find efficient techniques for producing collisions for single execution of f. Once that is done, attacker saves the value of fixed IV.

* Secure Hash Algorithm (SHA)

- The SHAs are finally family of cryptographic hash func^{n}s developed by National Institute of Standards & Technology (NIST). It comprises of 4 SHA, i.e., SHA-0, SHA-1, SHA-2 & SHA-3

- SHA-0 → It is a 160 bit hash func^n, was published by NIST in 1993. It had few weaknesses & did not become very popular & was replaced by SHA-1.

- SHA-1 → It is a 160 bit hash func^n & is employed in several widely used applications & protocols including secure socket layer (SSL). Security weaknesses were discovered in SHA-1 & it is no longer used ~~No successful attacks~~ after 2010.

- SHA-2 → It has 4 variants: SHA-224, SHA-256, SHA-384, SHA-512 depending upon no. of bits in their hash value. It is a strong hash func^n & uses best design of SHA-1. No successful attacks have yet been reported on SHA-2 hash func^n.

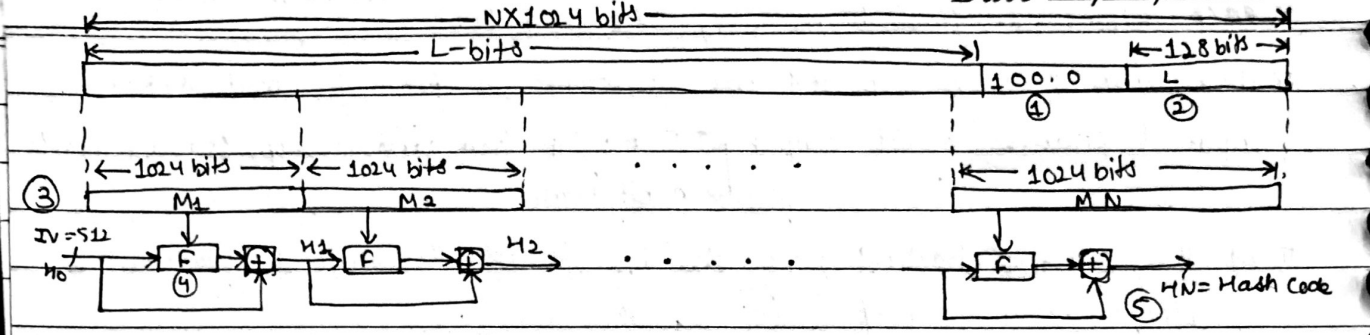
- SHA-3 → A hash func^n known as 'Keccak' is chosen by NSIT as the new SHA-3 standard. Keccak offers many benefits such as efficient performance & good resistance for attacks. It supports same hash lengths as SHA-2 & its internal structure differs significantly from rest of the SHA-family.

Punk

- SHA-512 logic

- This algorithm takes an I/P a msg with a max length of less than 2^{128} bits & produces O/P a 512-bit message digest. The input is processed in 1024-bit blocks.

Date __/__/____



Step1: Append Padding Bits

The msg is padded so that its length is congruent to 896 module 1024. Padding is always added, no. of padding bits is in the range of 1 to 1024. Single 1 bit is followed by necessary no. of a bits.

Step2: Append Length

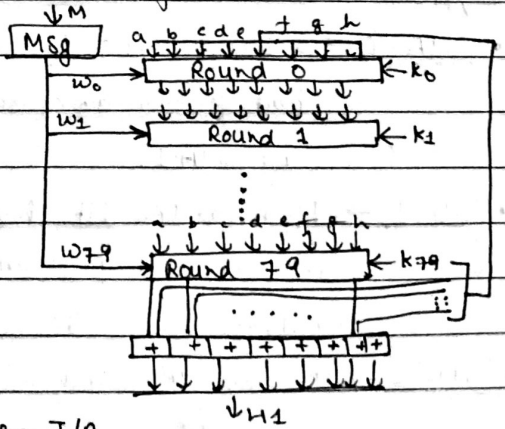
A block of 128 bits is appended to the msg. This block is treated as an unsigned 128 bit integer & contains length of msg before padding.

Step3: Initialize Hash Buffer

A 512-bit buffer is used to hold intermediate & final result of hash func^n. The buffer can be represented as 8 64-bit registers.

Step4: Processing message in 1024 bits blocks

The module F consists of 80 rounds & works as follows:-



(a) Each round takes I/P the 512-bit buffer value abcdefgh & updates the content of buffer

(b) Each round t makes use of 64-bit value W_t derived from the current 1024 bit block being processed.

(c) Each round also makes use of an addictive constant K_t $(0 \le t \le 79)$

(d) O/P of the 80th round is added to the I/P of 1st round to produce H_i.

Addition is done independently for each of the 8 words in the buffer

$H_0 = IV$
$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$
$MD = H_N$

$I_V \rightarrow$ initial value of buffer
$abcdefgh_i \rightarrow$ O/P of last buffer
$N \rightarrow$ no. of blocks in msg
$MD \rightarrow$ Final digest value

Step5: Output

After all 1024-bit blocks have been produced, the O/P from the N^{th} stage is 512-bit msg digest

Punk

- Message Digest (MD)

- The MD family comprises of hash func^{n}s MD 2, MD 4, MD5 & MD 6. MD5 digests have been widely used to provide assurance about integrity of transferred file (ex- checksum for the files). It produces 128-bit hash value.

Date __/__/____

* Authentication Applications

- Kereberos is an authentication service designed for use in a distributed environment.
- Kerberos makes use of a trusted third-party authentication service that enables clients & servers to establish authenticated communication.
- X.0509 defines the format for public-key certificates. This format is widely used in a variety of applications.
- A public key infrastructure (PKI) is defined as the set of hardware, software, people, policies, & procedures needed to create, manage, store, distribute, & revoke digital certificates based on asymmetric cryptography.
- Typically, PKI implementations make use of X.509 certificates

* Kerberos

- It is protocol for authenticating service requests blw trusted hosts across an untrusted n/w such as Internet. Kerberos is built-in to all major OSes
- The problem that Kerberos address is:
  "Assume an open distributed environment in which clients wish to access services on servers that are distributed throughout the network"
- The servers should be able to restrict access to authorized users & should be able to authenticate requests for services. So, rather than building authentication protocols at each server, Kerberos provides a centralized authentication server whose func^n is to authenticate users to servers & servers to users. It relies exclusively on symmetric ~~authentication~~ encryption, making no use of public key encryption.

- Requirements of Kerberos

(1.) Secure: A n/w eavsdropper should not be able to obtain necessary info. Kerberos should be strong enough that no weak link can be found.

(2.) Reliable: Kerberos should be highly reliable, as are the services that rely on Kerberos for access control needs Kerberos for supported ~~services~~ services. It should emply a distributed server architecture with one system to backup another.

(3.) Transparent: User should not be aware that authentication is taking place, beyond the requirement to enter password.

(4.) Scalable: The system should be capable of supporting large no. of clients & servers.

Punk

— Working:

- To start the Kerberos authentication process, the initializing client sends a request to an authentication server for access to a service.
- The initial request is sent as plain text because no sensitive info is ~~stored~~ included in the request.

Date __/__/____

- The authentication server retrives the initiating client's private key, assuming the client's private username is in the KDC database.
- If client username is found, then authentication server generates a sessionkey & if not found, then client can't be authenticated.

once per user logon session

① C → A S $ID_c || ID_{tgs} || TS_1$

② AS → C $E(K_c, [K_{c,tgs} || ID_{tgs} || TS_2 || Lifetime_2 || Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} || ID_c || AD_c || ID_{tgs} || TS_2 || Lifetime_2])$

ⓐ Authentication service exchange to obtain ticket-granting ticket

once per type of service

③ C → TGS $ID_v || Ticket_{tgs} || Authenticator_c$

④ TGS → C $E(K_{c,tgs}, [K_{c,v} || ID_v || TS_4 || Ticket_v])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} || ID_c || AD_c || ID_{tgs} || TS_2 || Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} || ID_c || AD_c || ID_v || TS_4 || Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_c || AD_c || TS_3])$

ⓑ Ticket-granting system service exchange to obtain service-granting ticket

once per service session

⑤ C → V $Ticket_v || Authenticator_c$

⑥ V → C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$Ticket_v = E(K_v, [K_{c,v} || ID_c || AD_c || ID_v || TS_4 || Lifetime_4])$

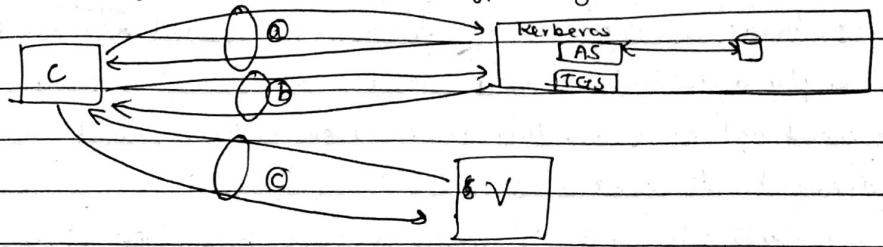
$Authenticator_c = E(K_{c,v}, [ID_c || AD_c || TS_5])$

ⓒ Client/Server Authentication Exchange to obtain service

C = Client AS = Authentication Server V = server ID_c = identifier of user C

ID_v = Identifier of V P_c = Password of user on C AD_c = network address of C

TGS = Ticket granting server K_v = secret encryption key shared by AS & V



Punk

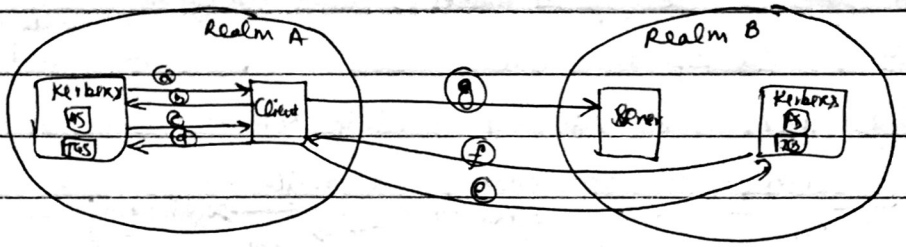
- Kerberos Realm : A full-service kerberos environment consisting of a kerberos server, a no. of clients, & a no. of application servers require the following:

① The kerberos server must have user ID & hashed passwords for all participating users in its database. All users are registered with the kerberos server

② The kerberos server must share a secret key with each server. All servers are registered with kerberos server.

Such an environment is referred to as a "Kerberos realm".

③ The kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two kerberos servers are registered with each other.



ⓐ Request Ticket for local TGS ⓑ Ticket for local TGS
ⓒ Request Ticket for remote TGS ⓓ Ticket for remote TGS
ⓔ Request ticket for remote server ⓕ Ticket for remote server
ⓖ Request remote service

- Difference b/w Kerberos V4 & V5 : Challenges in v4 overcome by v5:

① Environmental shortcomings:
ⓐ Encryption System Dependence ⓑ IP Dependence: ⓒ Msg byte ordering
ⓓ Ticket lifetime ⓔ Authentication forwarding ⓕ Inter realm authentication

② Technical Deficiencies:
ⓐ Double encryption ⓑ PCBC (Propagating Cipher Block Chaining) Encryption
ⓒ Session keys ⓓ Password attacks

* IP Security

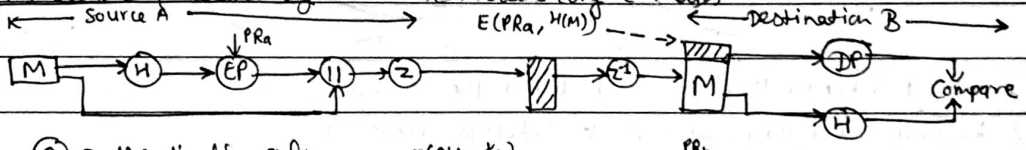
- IP security is a n/w protocol suite that authenticates & encrypts the packets of data sent over a n/w. IPsec includes protocols for establishing mutual authentication b/w agents at the beginning of the session & negotiation of crypto keys to use during the session.
- IPsec can protect data flows b/w a pair of hosts, b/w a security gateway or b/w n/w & host. IPsec is used for protecting communications over IP n/w.
- IPsec is an end-to-end security scheme operating in the Internet layer of the IP suite. IPsec can automatically secure applications at the IP layer.

Punk

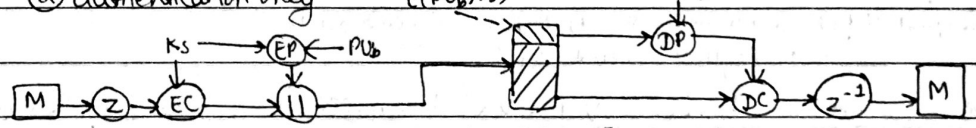
* PGP (Pretty Good Privacy)

- It is popular program used to encrypt & decrypt email over the Internet as well as authenticate msgs with digital signatures & encrypted stored files.
- In PGP, each user has an encryption key, i.e., publicly known & a private key, i.e., known only to user. You can encrypt a msg you send to someone else using their public key. When they receive it, they decrypt it using their private key.
- PGP comes in 2 public key versions - RSA & Diffie Hellman & PGP uses fastest encryption algo to encrypt the msg.
- PGP is an open-source freely available S/W package for email security. It provides

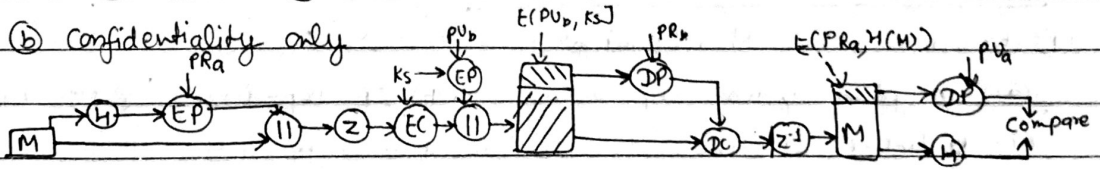
1. Authentication - through the use of digital signature (DSS/SHA or RSA/SHA)
2. Confidentiality - through the use of symmetric block encryption (CAST-128 or IDEA or 3-key Triple DES with Diffie Hellman or RSA)
3. Compression - through the zip algorithm
4. email compatibility - using the radix-64 encoding scheme
5. Segmentation & reassembly - to accomodate long e-mails



(a) authentication only



(b) Confidentiality only



(c) Confidentiality & authentication

PGP Cryptographic Functions

- PGP is used widely because:

1. It is freely available & runs on a variety of platforms
2. It is based on algorithms that are considered extremely secure
3. It has a wide range of ~~confidentiality~~ applicability
4. It is now on an Internet standards track.

(a) Authentication

- PGP provides digital signature service. The sequence is as follows:

1. Sender creates a msg
2. SHA-1 generates 160 bit hash code of the msg
3. Hash code is encrypted with RSA using sender's private key.
4. Receiver uses RSA with sender's public key to decrypt & recover the hash code.

Punk

⑤ Receiver generates a new hash code for the msg & compares with the decrypted one. If the two match, the msg is accepted as authentic.

- Because of RSA - receiver is assured that authenticated machine can generate digital signature.

Date __/__/____

- Because of SHA-1 - receiver is assured that no one else could generate a new msg with same hash code.
- Detached Signature - It may be stored & transmitted separately from the msg it signs.

(b) Confidentiality

- It is promised by encrypting msgs to be transmitted or stored locally as files. Encryption algorithm CAST-128 is used.

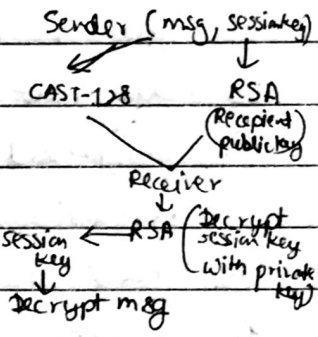
① The sender generates a msg & a random 128 bit no which is used as a session key for this msg only.

② Msg is encrypted using CAST-128 with the session key.

③ Session key is encrypted with RSA using recepient's public key.

④ Receiver uses RSA with its private key to decrypt & recover the session key.

⑤ The session key is used to decrypt msg.



(c) Authentication + Confidentiality

Signature is generated
↓
msg + sign encrypted using CAST-128
↓
session key is encrypted with RSA

Sender signs the msg with its own private key
Encrypts the msg with session key
Encrypts the session key with recepient's public key

Authentication

Sender (msg)
↓
SHA-1 - 160 bit hash code
↓
Hash code is encrypted using RSA sender's private key
↓
Receiver uses sender's public key to decrypt hash code
↓
Matches hash code

Confidentiality

Sender (msg, session key)
↓
Msg encrypted using sessing key CAST-128
↓
Session key is encrypted with RSA using recepient's public key
↓
Receiver use RSA with its private key to decrypt session key
↓
Session key is used to decrypt msg

(c) Compression

- PGP compresses the msg after applying the signature but before encryption. It saves space for both e-mail transmission & for file storage.
- Signature is created before compression because

One can store only the uncompressed msg together with the signature for future verification

Punk

#If signature is done after compression, then it would be necessary either to store a compressed msg or to recompress the msg when verification required.

- Msg encryption is applied after compression to strengthen the cryptographic security because compressed msg has less redundancy than original plain text & hence cryptanalysis is difficult.

Date / /

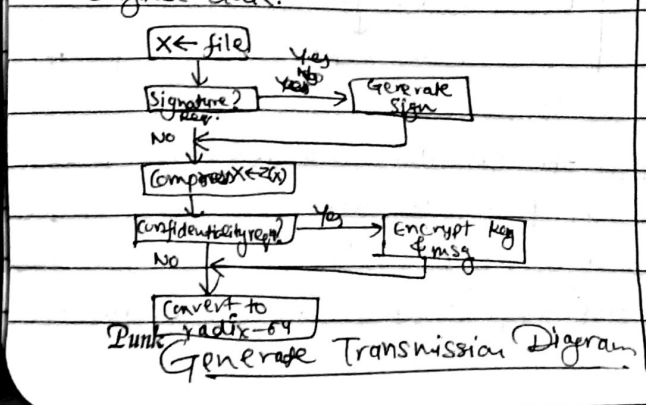
(d) Compatibility

- The resulting encrypted blocks consists of a stream of 8bit octets. Many e-mail systems only permit the use of blocks consisting of ASCII text.
- To accodomdate this restriction, PGP provides service of converting the 8-bit binary stream to a stream of ASCII characters.
- Radix-64 conversion technique is used for this conversion. 3 octets of binary data is mapped into 4 ASCII characters.

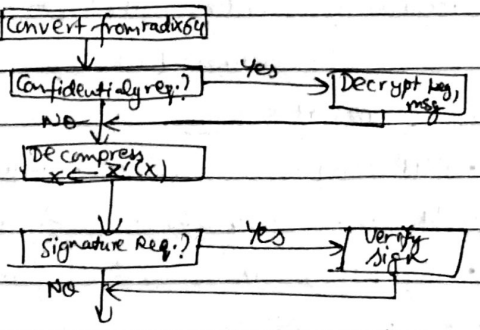
(1.) Signature is created using hash code of the uncompressed plain text.
(2.) Plain text signature is compressed.
(3.) Compressed text signature is encrypted & added with public-key-encrypted Symmetric encryption key.
(4.) Finally, the entire block is converted to radix-64 format.
(1.) On reception, the incoming block is converted back from radix-64 format to binary
(2.) Recover the session key & decrypts the msg.
(3.) ~~Decrypts~~ Decompress the msg
(4.) Recover the transmitted hash code & matches with its own calculated hash code.

(e) Segmentation

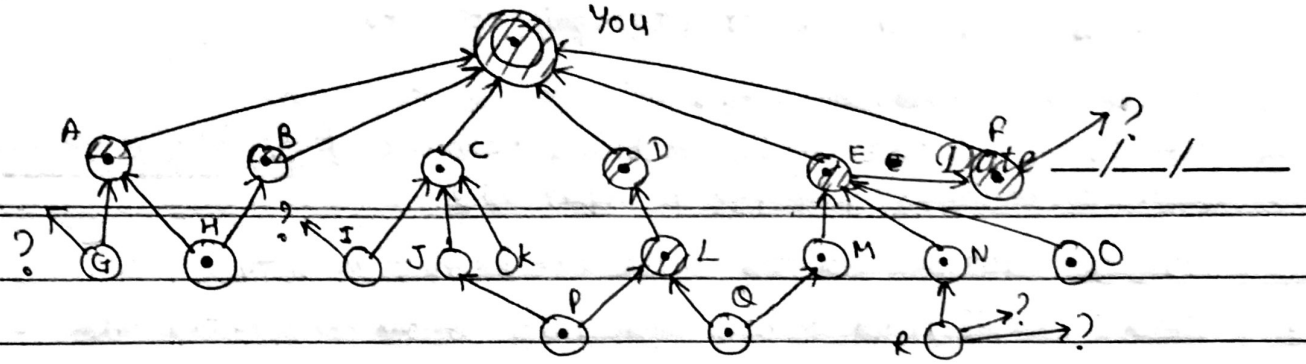
- Email facilities often are restricted to maximum message length (ex-max. length of 50,000 octets). Any msg longer than that must be broken up into smaller segments each of which is mailed separately.
- PGP automatically subdivides a msg that is too large into segments that are small enough to send via e-mail.
- Segmentation is done after all other processes, including the radix 64-conversion.
- At receiving end, PGP must strip off e-mail headers & reassemble the entire original block.



Punk Generate Transmission Diagram



Generic Reception Diagram



? ≡ unknown signatory

Ⓧ→Ⓨ ≡ X is signed by Y

◍ ≡ key's owner is trusted by you to sign keys

◓ ≡ key's owner is partly trusted by you to sign keys

⊙ ≡ key is deemed legitimate by you

PGP Trust Model Example

* S/MIME (Secure/Multipurpose Internet Mail Extension)

- It is a security enhancement to MIME Internet e-mail format standard.
- In terms of general functionality, S/MIME is very similar to PGP. Both offers ability to sign & encrypt msgs.
- Functions of S/MIME

① Enveloped Data: It consists of encrypted content & the encryption keys for one or more recepients.

② Signed Data: Digital signature is formed by taking msg digest of the content to be signed & then encrypting with signer's private key.

Content signature are encoded using base 64 encoding. A signed msg can only be viewed by recepient with S/MIME capability.

③ Clear-Signed Data: only digital sign. is encoded with base 64 & hence recepients without S/MIME capability can be viewed the msg, although they can't verify the sign.

④ Signed & Enveloped Data: Signed only & encrypted only entities may be nested so that encrypted data may be signed & signed data / clear signed data may be encrypted.

* Side Channel Attack

- It is an attack based on the information gained from the physical implementation of cryptosystem, rather than brute force or theoretical weaknesses in the algorithm.
- Side channel attacks monitor Power consumption & electromagnetic emissions

Punk

while a device is performing cryptographic operations.

- These attacks are relatively simple & inexpensive to execute.
- Some side-channel attacks require technical knowledge of the internal operation of the system on which cryptography is implemented.

Date __/__/____

- Cache Attack : Attacker monitors cache accesses made by victim.
- Timing Attack : Attacks based on how much time various computations take
- Power Monitoring Attack: use varying power consumption by hardware.
- Electromagnetic attack: based on leaked electromagnetic radiation, which can provide plaintexts & other information
- Accoustic Attack: Exploits sound produced during computation

- Counter measures

(1.) Eliminate/reduce the release of information.

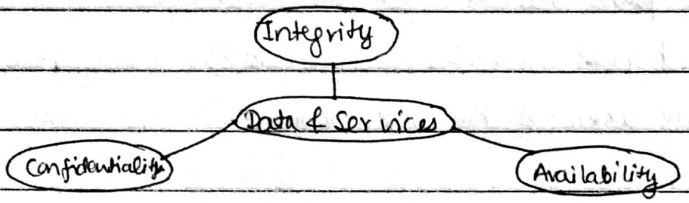
(2.) Eliminate rel^{n}ship b/w the leaked information & secret data, i.e., make the leaked info unrelated to the secret data through some form of randomization of the ciphertext that transforms the data in a way that can be undone after the cryptographic operation is completed.

Unit-IV

Date __/__/____

* Computer Security

- The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability & confidentiality of information system resources (i.e., h/w & s/w)
- Integrity: Assets can be modified by ~~authorized~~ authorized parties only
- Availability: Assets be ~~avaible~~ available to authorized parties
- Confidentiality: Requires information in a computer system only be accessible by authorized parties. Individuals set their own privacy requirements.



- Additional Requirements:
  - Authenticity: Requires that a computer system be able to verify the identity of a user
  - Accountability: Requires the detection & tracing of a security breach to a responsible party.

* Intruders

- Unauthorized intrusion into a computer system / network is one of the most serious threats to computer security. Three classes of intruders are:
  1. Masquerader: An individual who is not authorized to use the computer & who penetrates a system's access controls to exploit the system.
  2. Misfeasor: A legitimate user who accesses data, programs or resources for which such access is not authorized or who is authorized for such access but misuse his/her privileges.
  3. Clandestine User: An individual who ~~seizes~~ seizes supervisory control of the system & uses this control to access other's control.
- The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system.
- Intruders ~~acquire~~ acquire information that should have been protected.

Punk

- Intrusion Techniques

- Aim to gain access for increase privileges on a system
- Basic attack methodology
  → Information gathering → Initial access → Privilege Escalation → Covering tracks

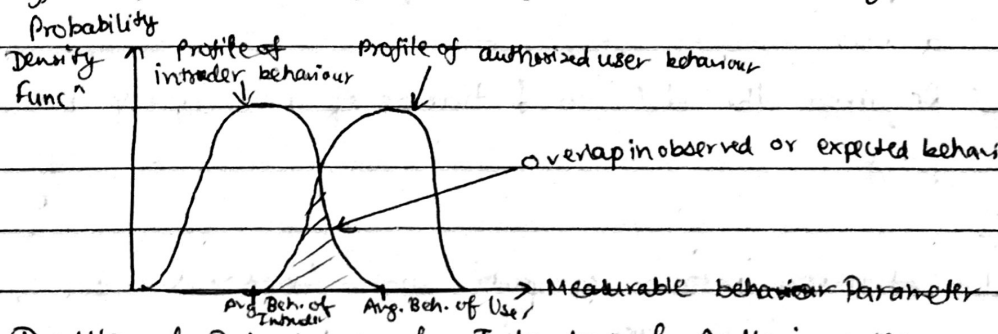
Date: __/__/____

- Key goal often is to acquire passwords
- Following techniques for learning passwords:

(1.) Try default passwords used with standard accounts that are shipped with the system. Many admins do not bother to change these defaults
(2.) Exhaustively try all short passwords (those of one to three characters)
(3.) Try words in the system's online dictionary or a list of likely passwords.
(4.) Collect information about users
(5.) Try user's phone numbers, Social security numbers & room numbers
(6.) Try all legitimate license plate numbers for this state
(7.) Use a Trojan Horse to bypass restrictions on access
(8.) Tap a line b/w a remote user & the host system

* Intrusion Detection

- Intrusion detection is based on the assumption that the behaviour of the intruder differs from that of a legitimate users in ways that can be quantified.



Profiles of Behaviours of Intruders & Authorized Users

- Approaches to Intrusion Detection

(1.) Statistical Anamoly Detection: It involves collection of data related to behaviour of legitimate users over a period of time. Statistical tests are applied to observe behaviour of users to find out illegitimate users.
  (a) Threshold Detection - Involves defining thresholds, independent of user, for the frequency of occurrence of various events.
  (b) Profile Based - A profile of activity of each user is developed & used to detect changes in the behaviour of individual accounts.
(2.) Rule-based Detection: Defines a set of rules that can be used to decide that a given behaviour is that of an intruder.
  (a) Anamoly Detection - Rules are developed to detect deviation from previous usage patterns

Punk

⑥ Penetration Identification :- An expert system approach that searches for suspicious behaviour.

* Intrusion Detection System (IDS)

Date __/__/____

- An IDS monitors n/w traffic & monitors for suspicious activities & alerts the system or n/w administrator. Its main func^s are to identify malicious activity, log information about this activity & report it.
- IDS is a device or S/w application that monitors n/w for or system for malicious activity or policy violations. If found any, detected activity is typically reported either to an admin or collected centrally using a security information & event management (SIEM) system.
- There is a wide spectrum of IDS varying from antivirus S/w to hierarchial system that monitors the traffic of an entire backbone n/w.

① Network Intrusion Detection System (NIDS)

→ NIDS are placed at a strategic points within the network to monitor traffic to & from all devices on the n/w. It would scan all the inbound & outbound traffic.

→ It performs an analysis of passing traffic on the entire subnet, & matches the traffic, i.e., passed on the subnets to the library of known attacks. Once an attack is identified, the alerts are sent to the admin.

→ Ex:- Opnet & NETsim

② Host Intrusion Detection System (HIDS)

→ HIDS run on individual hosts or devices on the n/w.

→ It monitors the inbound & outbound packets from the device only & will alert the user / admin if suspicious activity is detected.

→ It takes snapshot of the existing system file & matches it to previous snapshot. If modified, an alert is sent to the administrator.

③ Signature Based IDS

→ A signature based IDS will monitor packets on the n/w & compare them against a database of signatures or attributes from known malicious threats.

- Passive IDS
  - Simply detects & alerts the administrator
  - It is upto admin to take action to block the activity
- Reactive IDS
  - It will not only detect suspicious activity & alert the admin but will take pre-defined actions to respond to threat.

Punk

* Honeypots

- Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems. These are designed to:

Date __/__/____

① Divert an attacker from accessing critical systems
② Collect information about the attacker's activity
③ Encourage the attacker to stay on system long enough for admins to respond

- These systems are fabricated with information that appear valuable but a legitimate user of the system wouldn't access. Thus, any access to the honeypot is a suspect.
- The system is aided with sensitive monitors & event loggers that detect these accesses & collect information about the attacker's activity.
- Administrators have time to log & track the attacker without exposing productive systems.

* Intrusion Prevention System (IPS)

- Intrusion prevention system (IPS), also known as Intrusion Detection & Prevention System (IDPS), are network security / threat prevention technology that monitors network traffic & system activities for malicious activities.
- The main func^n of IPS are to identify malicious activity, log information about this activity, attempt to block / stop it & report it.
- IPS are considered as extensions of IDS because they both monitor traffic / system for malicious activities.
- The main differences are, unlike IDS, IPS are placed in-line & are able to actively prevent / block intrusions that are detected.
- IPS often sits behind the firewall & provides complementary layer of security. IPS can take actions such as

① Sending an alarm to administrator
② Dropping malicious packets
③ Blocking traffic from the source address
④ Resetting the connection

- Types of IPS are:

① Network Based IPS (NIPS) - It monitors the entire n/w for suspicious traffic by analyzing protocol activity
② Wireless IPS (WIPS) - It monitors a wireless n/w for suspicious traffic by analyzing wireless networking protocol
③ Network Behaviour Analysis (NBA) - It examines n/w traffic to identify threats that generate unusual traffic flows, such as DDOS attacks, certain forms of malware & policy variations.

Punk

④. Host Based IPS (HIPS) – It is an installed s/w package that monitors a single host for suspicious activity by analyzing events occuring within the host.

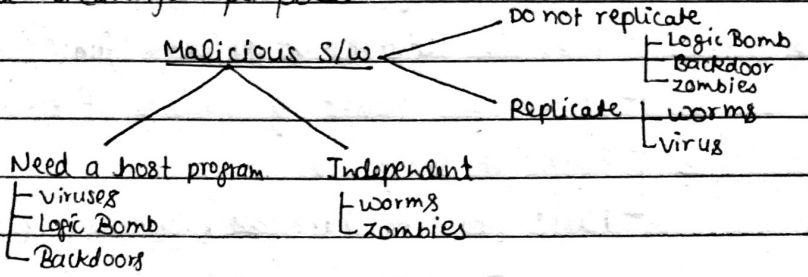
– Detection Methods:

Date __/__/____

①. Signature Based ②. Statistical anomaly based detection ③. Stateful protocol analysis detection

* Malicious Software

– Malicious s/w is a s/w that is intentionally included or inserted in a system for a harmful purpose.



– Virus is a piece of s/w that can 'infect' other programs by modifying them; the modification includes a copy of virus program which can then go on & infect other programs. (attach itself to another program & executes secretly when host runs).

– Worm is a program that can replicate itself & send copies from computer to computer across n/w connections. The worm may be activated to replicate & propagate again. In addition to propagation, they perform some unwanted func^n.

– Backdoor is a secret entry point into a program that allows someone to gain access without going through usual security access procedures. Programmers have used backdoors legitimately to debug & test programs.

– Logic Bomb is the code embedded in some legitimate program that is set to 'explode' when certain conditions are met (ex- a particular day of week, particular application running). Once triggered, bomb may alter or delete data or files, cause a machine halt or do some other damage.

– Zombie is a program that secretly takes over another Internet attached computer & then uses the computer to launch attacks that are difficult to trace to the zombie's creator. These are used in DOS attacks. The zombies are planted on hundred of computers & then used to attack the targeted web site.

– Trojan Horses is an apparently useful program that contains hidden code that when invoked performs some unwanted or harmful function. They can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly.

Punk

* Life cycle of virus

Date __/__/____

1. **Dormant Phase** — Virus is idle; It will eventually be activated by some event; Not all viruses have this stage.
2. **Propagation Phase** — Virus places identical copy of itself into other programs. Each infected program will now contain clone of the virus, which itself will propagate.
3. **Triggering Phase** — Virus is activated to perform func^n for which it was intended; It can be caused by variety of events
4. **Execution Phase** — Func^n is performed; Func^n may be harmless, such as msg on screen, or damaging, such as destruction of data files.

* Types of Viruses

1. **Parasitic Virus** : It attach itself to executable files & replicates
2. **Memory-resident**: Lodges in main memory as a part of resident system program, infects every program that executes.
3. **Boot Sector**: Infects master boot record & spreads when a system is booted from the disk containing virus
4. **Stealth** : A form of virus that is designed to hide itself from detection by antivirus s/w.
5. **Polymorphic**: Creates & copies during replication that are functionally equivalent but have different bit patterns.
6. **Metamorphic**: It rewrites itself completely at each iteration making it difficult to detect.

* Virus Structure

1. A virus can be prepended / postpended to an executable program, or can be embedded in some other fashion.
2. Key to operation - when infected program is invoked, it will first execute the virus code & then executes original code of program.

- An infected program begins with virus code & work as follows:

1. 1st line of code is a jump to main virus program
2. 2nd line is a special marker that is used by virus to determine whether the program is already infected or not
3. When program is invoked → control transfers to main virus program, which first seeks uninfected executable file & infects them
4. Next, virus may perform some damaging action
5. Finally the virus transfers control to the original program.

```
program V
{ goto main;  //transfers control to virus program
  1234567;
  subroutine infect-executable
  { }
  subroutine do-damage
  { }
  main
  { infect-executable;
    do-damage;
    goto next;
  }
}
```

The user is unlikely to notice any difference b/w the execution of infected & uninfected program.

* State of Worm Technology:

Date __/__/____

- Multiplatform
- Multiexploit
- Ultrafast spreading
- Polymorphic
- Transport vehicles
- Zero-day exploit

* Virus Countermeasures

1. Detection - once the infection has occured, determine that it has occured & locate the virus.
2. Identification - Identify the specific virus that has infected the program
3. Removal - Remove all traces of the virus from the infected program & restore it to its original state. Remove the virus from all infected systems so that the disease cannot spread further.

- If removal is not possible, then discard the infected program & reload a clean backup version.

- Four Generations of Antivirus

1. 1st Generation: Simple scanner requires virus signature to identify virus. These are limited to detection of known viruses.
2. 2nd Generation: Uses heuristic scanner which uses heuristic rules to search for virus infection. They look for fragments of code that are often associated with viruses. Integrity checking - a checksum can be appended to each program.
3. 3rd Generation: Memory resident programs that identify a virus by its action rather than its structure in an infected program. A small set of actions that indicate an infection needs to be identified.
4. 4th Generation: packages consisting of variety of antivirus techniques used in conjunction. Includes scanning & activity trap components. Include access control capability which limits the ability of viruses to ~~penerate~~ penetrate a system & then limits the ability of a virus to update files to pass an infection

* Firewall

- A firewall forms a barrier through which the traffic going in each direction must pass.
- It is designed to operate as a filter at the level of IP packets or may operate at higher layer protocol.
- A firewall security policy dictates which traffic is authorized to pass in each direction.

Punk

- Firewall Characteristics

① All traffic from inside to outside & vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local n/w / expect via the firewall.

② Only authorized traffic, as defined by security policy will be allowed to pass

③ The firewall itself is immune to penetration. This implies that use of trusted systems with a secure OS.

- Techniques used by Firewalls

① Service Control → Determines the type of Internet services that can be accessed. Firewall filters traffic on basis of IP address & TCP port no. (may provide proxy S/W).

② Direction Control → Determines the direction in which particular service requests may be initiated & allowed to flow through the firewall.

③ User Control → Controls access to a service acc^n to which the user is attempting to access it.

④ Behaviour Control → Controls how particular services are used. Ex- Firewall may filter e-mail to eliminate spam.

- Limitations of Firewalls

① cannot protect against attacks that bypass the firewall.

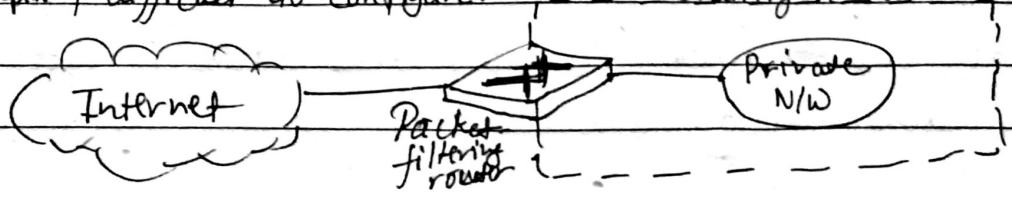
② doesnot protect against internal threats (disgruntled employee).

③ cannot protect against the transfer of virus-infected programs, impossible for firewall to scan all incoming files, emails for viruses.

* Types of Firewalls

① Packet-Filtering

- A packet filtering router applies set of rules to each incoming & outcoming IP packet & forwards or discards the packet.
- It allows or block packets based on criteria such as, source or destination IP addresses, protocol, source or destination port no. & various other parameters within IP header packet filler part
- Selection criteria : condition & pattern matching for decision making
- Action field : Specifies action to be taken if IP packet meets the criteria (Block/Allow)
- It works well for small n/w but when applied to larger n/w can become very complex & difficult to configure.

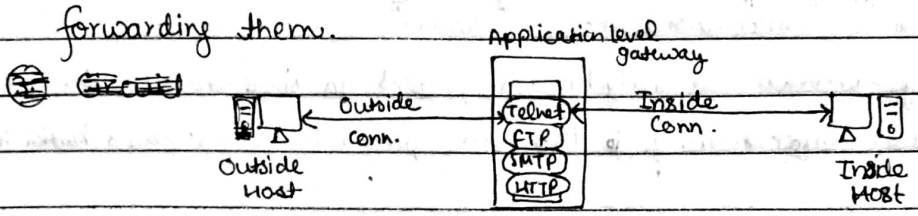


Punk

② Application-level Gateway (Proxy Servers)

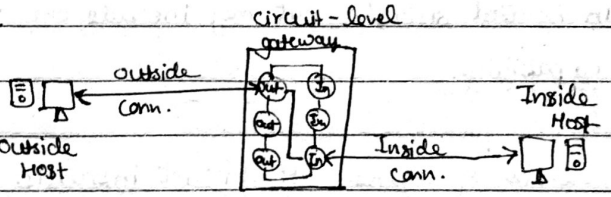
Date __/__/____

- It deals with network traffic by passing ~~through~~ all packets through a separate 'proxy' application that examines data at an application level.
- A proxy firewall doesn't allow direct connection b/w a trusted server or client & an untrusted host.
- They intercept incoming & outgoing packets, run proxies that copy & forwards information across gateway. Proxies examines & filters individual packets before forwarding them.



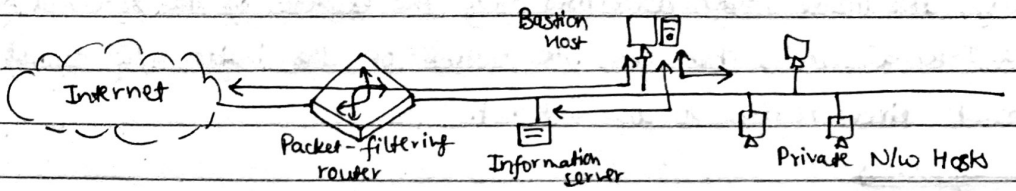
③ Circuit-Level Gateway

- It is an intermediate soln b/w packet filter & application gateway. It runs at transport layer & hence can act as proxy for applications.
- It does not permit end-to-end TCP connection across gateway. It sets up two TCP connections:
  - one b/w itself and TCP user on inner host
  - b/w itself & TCP user on outer host
- Then, it relays TCP segments from one n/w to the other without examining contents.



* Firewall configurations

① Screened host Firewall system (Single-Home Bastion Host)



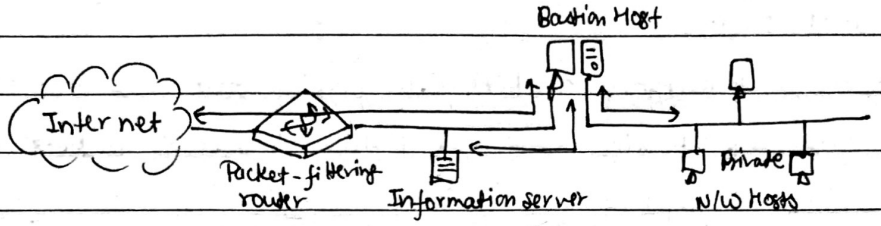
- In this configuration, the firewall consists of two systems: a packet-filtering router & a bastion host. Typically the router is configured so that
  - (a) for traffic from the Internet, only IP packets destined for bastion host are allowed in
  - (b) for traffic from the Internal n/w, only IP packets from bastion host are allowed out.
- The bastion host performs authentication & proxy func^n^s.
- This config. has greater security than simply a packet-filtering router or application level gateway alone.

Punk

- This config. also affords flexibility in providing direct Internet access.
- If the packet-filtering router is completely compromised, traffic could flow directly through the router b/w Internet & other hosts on private n/w.

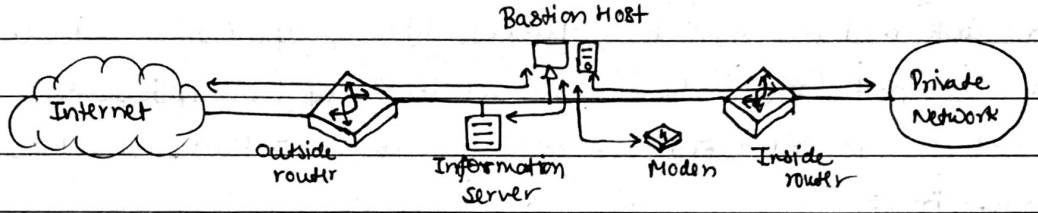
Date __/__/____

② Screened Host Firewall System (Dual-Homed Bastion Host)



- It physically prevents security breach present in single-homed bastion host.
- Advantages of dual layer security that were present in single-homed bastion host are present.

③ Screened Subnet Firewall System



- It is the most secure config. among these. In this config., two packet-filtering routers are used, one b/w bastion host & Internet and one b/w bastion host & internal n/w.
- This config. creates an isolated subn/w but may include one or more info. servers & modem for dial-in capability.
- Advantages:
  - There are now three levels of defense to thwart intruders.
  - The outside router advertises only the existence of the screened subnet to the Internet; therefore, the internal n/w is invisible to the Internet.
  - Similarly, the inside router advertises only the existence of the screened subnet to the Internal n/w; therefore, the systems on the inside n/w cannot construct direct routes to the Internet.

* Trusted Systems

- A Trusted system is a computer & OS that can be verified to implement a given security policy.
- The focus of trusted system is access control. A policy is implemented that dictates what objects may be accessed by what subjects.

Punk

- Data Access Control

• A general model of access control is that of an "access matrix". The basic elements of the model are:

Date __/__/____

→ Subject: An entity capable of accessing object

→ Object: Anything to which access is controlled

→ Access Right: The way in which an object is accessed by a subject. Ex :- read, write & execute

• An access ~~card~~ matrix may be decomposed by "columns", yielding "access control lists" for each object, an access list lists users & their permitted access rights.

• An access matrix decomposed by "rows", yields "capability tickets" which specifies authorized objects operations for a user.

	Program 1	.....	Segment A	Segment B
Process 1	Read Execute		Read Write	
Process 2				Read
⋮				

object →
subject ↓
Access rights

ⓐ Access Matrix

Access control list for Program 1:
Process 1 (Read, Execute)
Access control list for Segment A:
Process 1 (Read, Write)
Access control list for Segment B:
Process 2 (Read)

ⓑ Access control list

Capability list for Process 1:
Program 1 (Read, Execute)
Segment A (Read, Write)

Capability list for Process 2:
Segment B (Read)

ⓒ Capability list

Access Control Structure

- The concept of Trusted Systems

• Information can be organized into gross categories & users can be granted clearances to access certain categories of data.

• When multiple categories or levels of data are defined, the requirement is referred to as "multilevel security". A multilevel secure system must enforce following:

→ No read up: A subject can only read an object of less or equal security level. This is referred as the Simple Security Property.

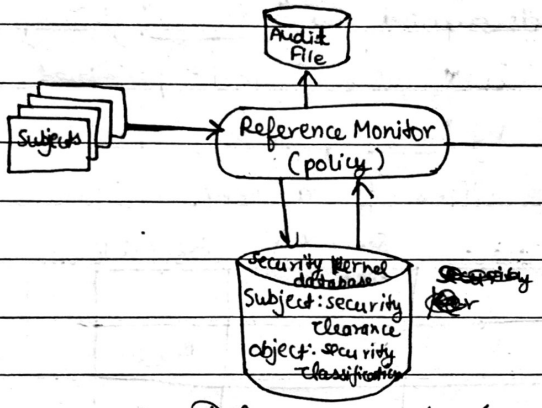
→ No write down: A subject can only write into an object of greater or equal security level. This is referred as the *-Property.

Punk

- These rules, if properly enforced, provide multilevel security.
- For data processing, the approach that has been taken & has been object of much research & development, is based on the "reference monitor" concept.

Date __/__/____

- The reference monitor is a controlling element in the h/w & O.S of a computer that regulates the access of subjects to objects on the basis of security parameters of the subject & object.



Reference Monitor Concept

- Important security events such as detected security violations & unauthorized changes to the security kernel database, are stored in the audit file.

- The reference monitor has access to a file, known as security kernel database, that lists the access privileges (security clearance) of each subject & the protection attributes (classification level) of each object.
- The reference monitor enforces security rules (no read up, no write down) & has the following properties:
  - → Complete mediation: The security rules are enforced on every access
  - → Isolation: The reference monitor & database are protected from unauthorized modification
  - → Verifiability: The reference monitor's correctness must be provable.
- A system that can provide such verification is referred to as a "trusted system".