UNIT-I INTRODUCTION

Security Trends - legal & ethical aspects of Security

Trends:-

→ Smart attackers
→ Spreading of mobile malwares.
→ Shortage of skill
→ IOT
→ Social n/w attacks
→ use of AI & machine learning.
→ complex Infrastructure

need for security:-

Data Security
computer Security
n/w Security
Internet Security.

Terminologies:-

cryptography → art encompassing principles & methods of transforming plaintext to unintelligeable message & then back into original form.

plaintext → original message

ciphertext → transformed message

key → critical information only known to sender & receiver

Encryption → process of plaintext → ciphertext

decryption → ciphertext → plaintext

Cryptanalysis → Study of principles and methods of transforming unintelligeable message to intelligeable

Security goals:-

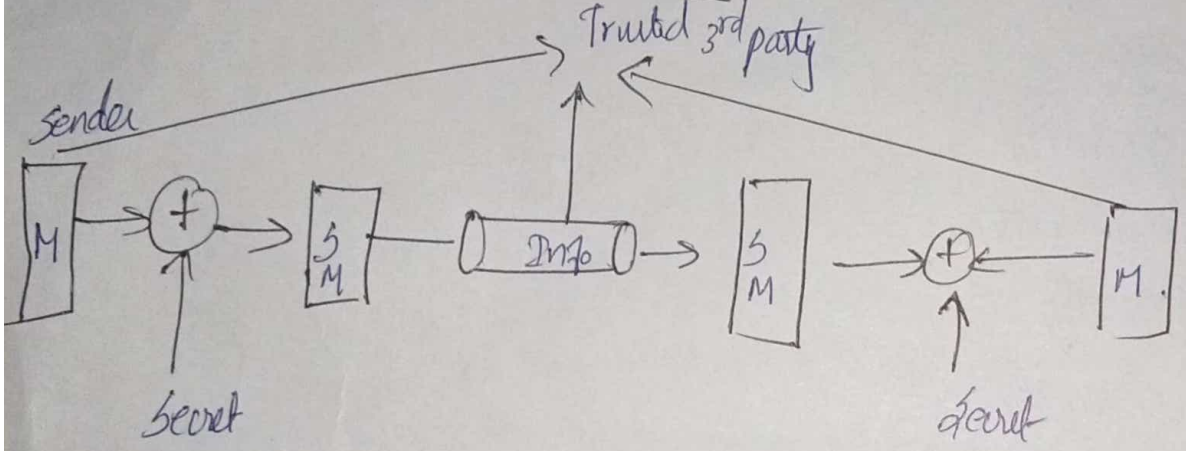
confidentiality

integrity

availability.

---

2) Model of n/w Security:-



* message is transformed from source to destination across internet.

2 components:-

* Security related transformation.
* Secret information

Basic tasks

algorithm

Secret information

methods for distribution

protocol

3) Security attacks:-

Asset → people, property, info.

Vulnerability → Security flaws in a system that allows an attack to be successful.

Threat → Anything that can exploit vulnerability, intentionally or accidentally, and obtain damage or destroy an asset.

Risk → The potential for loss, damage or destruction of an asset as a result of destroy assets.

control → proactive measure. (action, device, procedure, technique)

Types:-

* active attack
* Passive attack.

Active attack

some modification (or) false content

ex

Masquarade → entity pretendation

Replay → passive capture of data unit modification.

DoS → disruption of entire network

Passive Attacks:

Eavesdropping (or) monitoring of data

ex

Release of message contents

Traffic analysis.

Security Services

Authentication → determining whether someone or something is in fact, who or what it is declared to be

Access control → ability to limit and control.

Data confidentiality → concealment of information

integrity → stream of messages to a single.

non repudiation → prevents denying of transmission

OSI Security Architecture:

managers → way of organizing the task

OSI → Security attacks

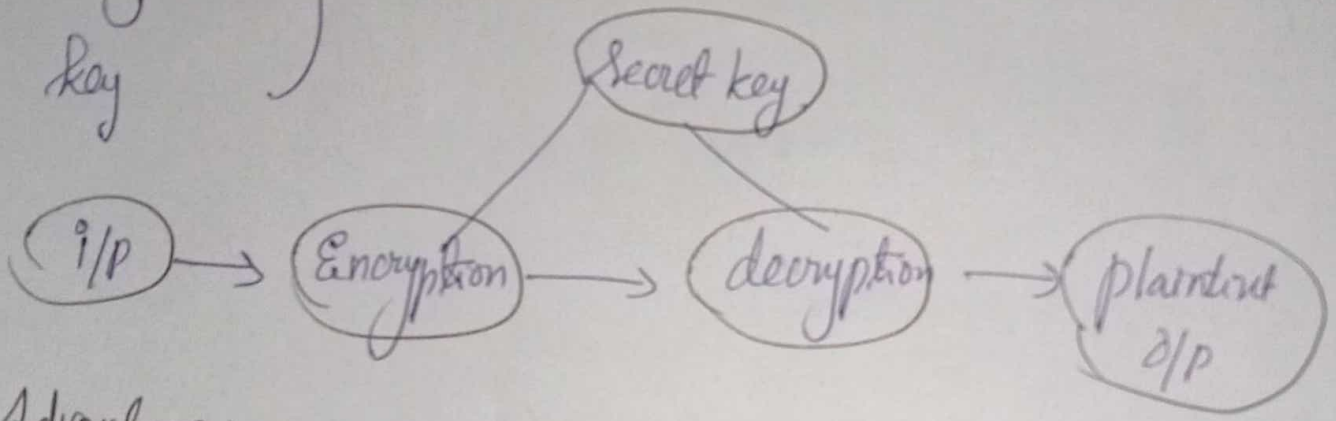
→ security mechanism

→ security services

Classical encryption techniques:-

plaintext
ciphertext
encryption
decryption
key

(already defined previously)



Advantages:-

* high rates of data throughput
* symmetric key
* stronger cipher.

Disadvantages:-

* key must remain secret
* many keypairs
* Digital Signature

Substitution techniques:-

→ changes characters.

Caeser cipher:-

* each alphabets are replaced by another alphabet some places down.

$$C = E(3, P) = (P+3) \bmod 26$$

general,

$$C = E(k, P) = (P+k) \bmod 26$$

decryption:-

$$P = D(k, C) = (C-k) \bmod 26$$

Ex Hello world, key = 3

C.T = KHOOR ZRUOG

Demerits:-

* encryption & decryption algorithm are known
* only 25 keys.

Monoalphabetic ciphers:-

* Substitution technique
* any letter can be substituted, as long as each other has unique substitute letter and vice versa

ex

P.T: hello

C.T: acggk

Demerits:-

* easy to break.

Play fair ciphers:-

* 5x5 Matrix
* key word: Monarchy is a keyword.

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

Matrix is constructed by filling the letters in key and then remaining letters to be filled

3 cases:-

→ Plaintext fall in same row of the matrix each are replaced by right letter.

→ Plain text fall in same column are replaced by the letter beneath

→ otherwise, each plaintext is replaced by the letter that lies in its own row and the column occupied by other letter.

P.T= meet me.

C.T= clkl kl.

Strength:-

* advanced method

* 676 diagrams possible.

* frequency analysis is much difficult

Polyalphabetic ciphers:-

different monoalphabetic substitutions as one proceeds through the plaintext message.

Features:-

* set of monoalphabetic rules.
* key for rule

Vigenere cipher:-

→ 26 caeser ciphers with 0 to 25.

ex a=0, b=1, ... z=25

Given a key x, plaintext y, the cipher text is at the intersection of the row labeled x and column labeled y.

* In this case cipher text is v.

ex

key = decept

P.T = weare

C.T = zICVT

Vigenere table

key	P.T							
		a	b	c	d	...	y	z
	a	A	B	C	D	...	Y	Z
	b	B	...			...	Z	A
	c	C				...	A	B
	:	:	:	:	:	:	:	:
	z	Z	A	B	C	...	X	Y

Strength:-

* multiple ciphertext letters for each plaintext letter
* letter frequency information is obscured

One Time pad ciphers:-

* message as sequence of 0's and 1's.
* This can be accomplished by writing all numbers in binary.

$$C_i = P_i \oplus k_i$$

* cipher text is generated by performing bitwise X-OR

decryption,

$$P_i = C_i \oplus k_i$$

eg P.T: 0010
key: 1010
C.T: 1000

Advantages:-

* completely unbreakable

Disadvantages:-

* very long key
* danger to reuse key

Transposition Techniques:-

* Some sort of permutation on plaintext

Rail fence:-

plain text is written down as sequence of diagonals and then read off as a sequence of rows.

ex P.T: meet at school house

m e a s h o h u s e
e t t c o l o

C.T : measholuetfcolos.

if the key size is 3,

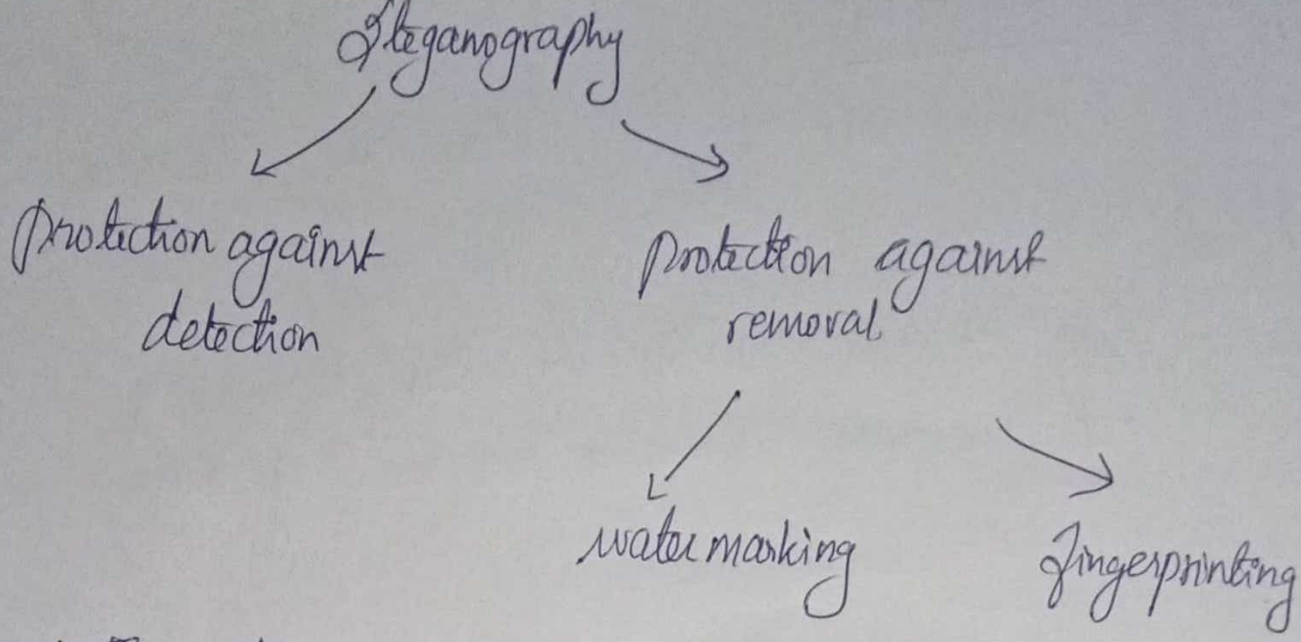
m a h h e
e t t c o l o s
e s o u

C.T: mahheettcolosesou

Steganography:-

* Steganography → Greek word "to hidden in plain sight"

* art and science of communicating in such a way that the presence of mesage cannot be detected



* The other major area of steganography is copyright marking, where the message to be inserted is used to assert copyright over a document

* data confidentiality.

Requirements :-

* Integrity of hidden information
* Stego object must remain unchanged
* In watermarking, changes in the stego object must have no effect on the watermark.

Techniques:-

* genome steganography
* hiding in text
* hiding in disk space

* hiding data in s/w and circuitry
* Information hiding in images
* hiding in n/w packets.

Cryptanalysis:-

* Process of trying to break any cipher text message to obtain the original plaintext itself
* breaking of codes.
* Brute force attack → tries every possible key.

Attacks:-

* cipher text only attacks
* known plaintext attacks
* chosen plaintext attacks
* Chosen ciphertext attacks.
* Chosen text attacks

Cryptology:-

"Cryptology is the art and science that deals with both cryptography and cryptanalysis"

Cryptanalysis attack types:

* known plaintext attack
* Chosen plaintext Analysis
* ciphertext only analysis
* Man in the middle attack
* adaptive chosen plaintext attack.

Unit - II.

Symmetric Key Cryptography.

Modular Arithmetic.

*) Much of modern number theory and many pratical problem are connected with modular arithmetic. In arithmetic modulo N, we are concerned with arithmetic on the integers, where we identify all numbers which differ by an exact multiple of N. That is,

$$x = y \bmod N \quad \text{if } x = y + mN.$$

*) This identification divides all the integer into N equivalence classes. We usually denote these by their "simplest" member, that is, the numbers 0, 1 ... N-1

Theorem:

$\equiv n$ is an equivalence relations on the integer An equivalence class consists of these integers which have the same remainder on division by n. The equivalence classes are also known as congurence classes modulo n.

Definition:

The set of all integers congruent to a modulo n is called the residue class [a]

Example:

w	-w	w^{-1}
0	0	---
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

Additive and Multiple inverse modulo 7

Addition Modulo 7.

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

Modu

To find $11^{13} \bmod 53$.

Soln:

$13 = 8+4+1$ so $11^{13} = 11^{8+4+1} = 11^8 * 11^4 * 11^1$

We can compute successive square of 11 to obtain, $11, 11^2, 11^4, 11^8$ and the multiply together $11^1 * 11^4 * 11^8$ to get the answer 11^{13}.

Because we are working mod 53, we will "take mods" at every stag of the calculation

Thus we have,

$11 \bmod 53 = 11$

$11^2 = 121$, $121 \bmod 53 = 121 - 2*53 = 15$.

$11^4 = (11^2)^2 = 15^2 \bmod 53 = 225 \bmod 53 = 225 - 4*53 = 13$

$11^8 = (11^4)^2 = 13^2 \bmod 53 = 169 \bmod 53 = 169 - 3*53 = 10$

Therefore $11^{13} \bmod 53 = 11 * 13 * 10 = 1430 \bmod 53 = 1430 = 26*53 + 52$

The answer is $11^{13} \bmod 53 = 52$.

Polynomial Polyarithmetic:

*) A polynomial is an expression of the form $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ for some non-negative integer n and where the coefficient

$a_0, a_1, a_2, \ldots a_n$, are drawn from some designated set S. The 'S' is called the coefficient set

*) when $a_n \neq 0$, then it is called as polynomial of degree n. A Zeroth - degree polynomial is called a constant polynomial.

*) If the value of $a_n = 1$ then the polynomial is said to be monic. If $n = 0$ then we simply have a constant known as a constant polynomial.

*) A non - constant polynomial is irreducible or prime, if it cannot be factorized as a product of polynomials of lower degree.

Addition of two polynomial

$$f(x) = a_2x^2 + a_1x + a_0$$

$$g(x) = b_1x + b_0.$$

$$f(x) + g(x) = a_2x^2 + (a_1+b_1)x + (a_0+b_0)$$

Subtraction of two polynomial.

$$f(x) = a_2x^2 + a_1x + a_0$$

$$g(x) = b_3x^3 + b_0$$

$$f(x) - g(x) = -b_3x^3 + a_2x^2 + a_1x + (a_0 - b_0)$$

Multiplication of two polynomials.

$$f(x) = a_2x^2 + a_1x + a_0$$

$$g(x) = b_1x + b_0.$$

$$f(x) \times g(x) = a_2b_1x^3 + (a_2b_0 + a_1b_1)x^2 + (a_1b_0 + a_0b_1)x + a_0b_0.$$

Finite fields.

A field is a set of elements on which two arithmetic operations i.e. addition and multiplication, have been defined and which has properties of abstract algebra arithmetic, such as closure, associative, commutativity, distributivity and having both additive and multiplicate inverses.

Properties:

*) It can be shown that finite fields have order p^n, where p is a prime.

*) It can be shown that for each prime p and each positive integer 'n' where, there is, upto isomorphism, a unique finite field of order p^n.

*) Let $GF(p^n)$ represent in a finite field of order p^n. GF stands for Galois field.

Groups:

*) A group G is non-empty set together with a binary operation $(*)$ such that the following three properties are satisfied:

Associative: $(a*b)*c = a*(b*c)$ for all $a, b, c \in G$

Identity: There is an element $e \in G$ such that $e*a = e*a$. For all $a \in G$.

Inverse: For each element $a \in G$, there is an element $b \in G$ such that $a*b = b*a = e$.

*) Order of a group G is the no. of elements it contains. Order of an element $g \in G$ is the smallest positive integer n such that $g^n = e$.

Properties of group:

*) For all $g \in G$, $g^0 = e$.

*) For all $n, m \geq 1$, $g \in G$.

1. $g^n = g^{n-1} * g$.
2. $g^n * g^m = g^{n+m}$.
3. $(g^n)^{-1} = g^{-n} = (g^{-1})^n$
4. $(g^m)^n = g^{mn}$.

If G is a group and for all $a, b \in G$ we have $a * b = b * a$ then G is called an Abelian group.

Ring with unity:

*) A ring R is a non empty set with two binary operations, addition and multiplication such that for all $a, b, c \in R$:

1. R is an abelian group under addition.
2. $a(bc) = (ab)c$ (associatively)
3. $a(b+c) = ab + ac = (b+c)a$ (commutatively)

*) A unit is a non-zero element of a commutative Ring with unity that has a multiplicative inverse.

*) A zero-Divisor is a non-zero element $a \in R$, R is a commutative Ring, such that there is a non-zero element $b \in R$ with $ab = 0$.

*) An internal domain is a commutative Ring with unity and no-zero-divisors.

RC4:

*) RC4 is an encryption algorithm that was created by Ronald Rivest of RSA Security. It is used in WEP and WPA, which are encryption protocal commonly used on wireless routers. The algorithm is based on the use of a random permulation.

*) RC4 was oginally very widely used due to its simplicity and speed. Typically 16 byte key are used for strong encryption, but shoter key length are also widely used due to restrictions.

*) Uses a variable length key from 1 to 256 bytes to initialize a 256-byte state table The state table is used for subsequent generation of pseudo-random bytes.

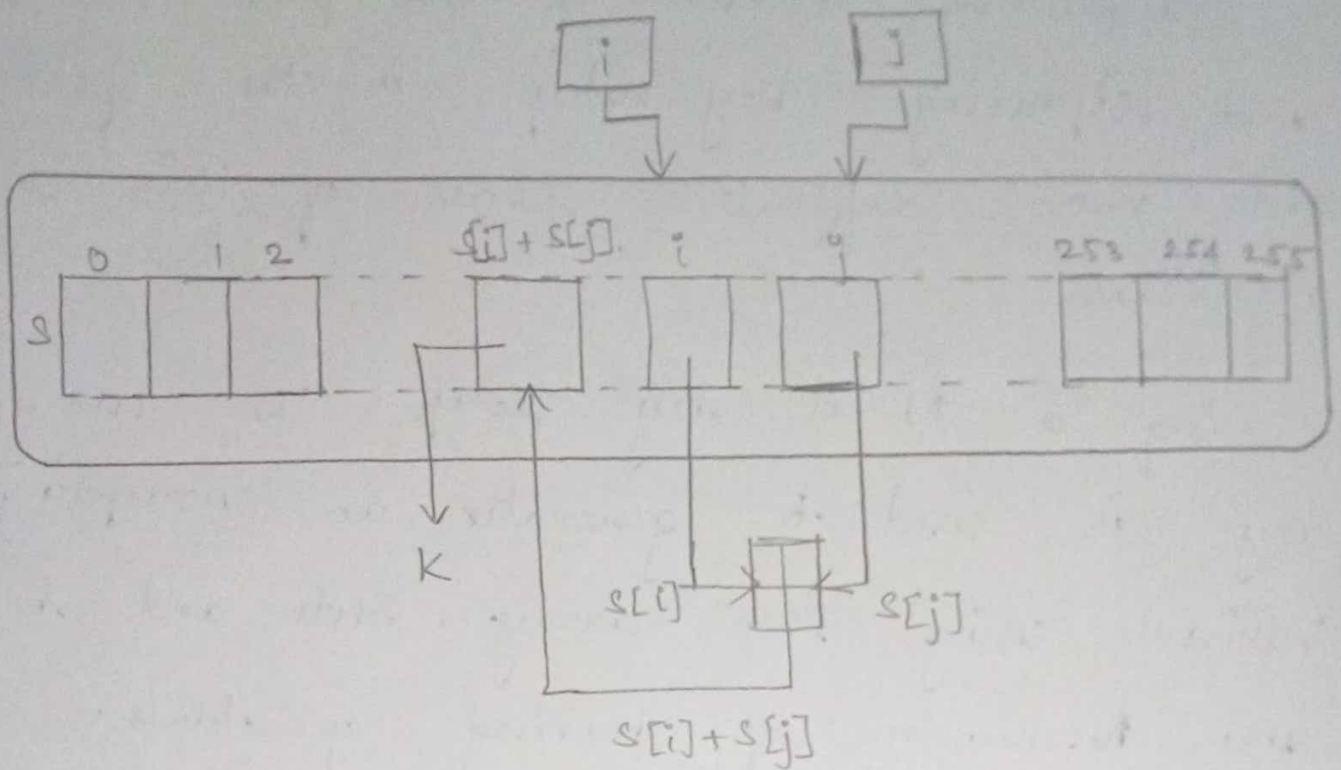
*) The key is often limited to 40 bits, because of expost restrictions but it is sometimes used as a 128 bit key. It has the capability of using keys between 1 & 2048 bits

The algorithm works in two phase, key setup and ciphering. key setup is the first and most difficult phase of this encryption algorithm.

* During a N-bit key setup the encryption key is used to generate an encrypting variable using two array, state and key and N: number of mixing operations.

* Once the encrypting variable is produced from key setup, it enters the ciphering phase, where it is XORed with the plain text message to create an encrypted message.

* The permutation is initialized with a variable key length key, typically between 40 and 256 bits, using the key-Scheduling Algorithm. Then the stream of bits is generated by a pseudo-random generation algorithm.



Strengths:

*) The difficulty of knowing where any value is in table.

*) A particular RC4 algorithm key can be used only once.

*) Encryption is about 10 times faster than DES.

Weakness:

*) The algorithm is Valunerable.

*) Once in every 256 keys can be a weak key

Block cipher principles:

*) A block cipher operates on block of data.

*) Algorithm breaks the plaintext into blocks and operates on each block independently.

*) Usually 2^n is the size of each block.

*) Security of block ciphers depends on the design of the encryption functions.

*) Software implementations of block ciphers run faster than software implementation of stream ciphers.

*) Error in transmitting one block generally donot affect another block.

*) Each block is enciphered independently, using the same key, identical plaintext block produce identical cipher text blocks.

*) Algorithm grabs the first 16-bytes of data, encrypts them using the key table.

*) After first block algorithm takes next block.

*) The key table does not chang from block to block.

Plain text = 277 bytes.

Block size = 16 bytes = $277/16$ = 14 blocks plus 3 bytes

*) Algorithm encrypts 14 blocks and 3 bytes remain

*) Fa encrypting last 3 bytes data padding is used

*) Extra bytes are added to make the last box size to 16 bytes.

*) To avoid having these kinds of copies in the cipher text, feedback modes are used.

*) Before plaintext block is enciphered, that block is XORed with proceeding cipher text block.

*) Taking E_K to be the encipherment algorithm with key K and I to be the initialization vector, the cipher text block chaining

Blowfish:

*) It's a symetric block cipher.

*) It can be used as drop in replacement for DES or IDEA.

Subkey and S-box generation:

Blowfish makes use of the key that ranges from 32 to 448 bits. That key is used to generate 18 - 32 bits subkeys and four 8 × 32 S-boxes containg a total of 1024 32-bits entries.

The total for subkeys and S-boxes is 1042 32-bit values, 4168 bytes.

The keys are stored in

$$K_1, K_2, \ldots K_j, \quad j = 1, 2, \ldots, 14$$

The sub keys are stored in P-array:

$$P_1, P_2, \cdots, P_{16}.$$

There are 4 S-boxes,

$$S_{1,0}, S_{2,1}, \cdots, S_{1,255}$$

Comments about AES structure:

*) AES structure is not feistal structure.

*) The key that is provided as i/p is expanded into array of forty-four 32-bit words.

*) 4 different stages are used., one of permutation and three of substitution.

*) Only the AddRoundKey stage make use of the key.

*) Each stage is easily reversible.

*) The description algorithm makes use of the expanded key in reverse order.

Application of AES:

*) AES can be used anywhere, Symetric key cryptography is needed.

*) Banking System use AES-128 and AES-256 to secure online banking.

1) Assume $C = E_{K2}(E_{K1}(P))$

2). Given the plaintext P and ciphertext C.

3). Encrypt P using all possible keys K_1

4) Decrypt C using all possible key K_2

Triple DES:

*) Triple DES is simply another mode of DES operation. It takes three 64 bits key for an overall key length of 192 bits.

*) The procedure for encryption is exactly the same as regular DES, but it is repeated three times.

*) Triple DES uses 2 or 3 keys.

*) The data is encrypted with the first key (K_1), decrypted with second key (K_2), finally encrypted again with the third key (K_3).

*) Brute force search impossible on triple DES

~~Middl~~

*) Meet - in - middle attacks need 256 plaintext - Ciphertext pair keys.

*) This means the effective key strength for triple DES is actually 168 - bits because each of three keys contains 8 parity bits that are not used during the encryption process.

Triple DES with two keys.

*) In triple DES with two keys share there are only two key K_1 used by first & third stage and K_2 used in second stage.

*) First plain text encrypted with key K_1 then o/p of step 1 is decrypted with K_2 and final the o/p second step is encrypted again with K1.

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

$$P = D(K_1, E(K_2, D[K_1, C])).$$

technique is.

$$C_0 = E_K(m_0 \oplus I)$$

$$C_i = E_K(m_i \oplus C_{i-1}) \quad \text{for } i > 0.$$

Advantages:

*) High diffusion.

*) Immunity to insertation of symbols.

Disadvantages:

*) Slowness of encryption.

*) Error propagation.

AES:

Advanced Encryption Standard (AES) is a Symetric block key 'cipher'

Evaluation criteria:

NIST evaluation criteria for AES are

*) Security.

*) Cost

*) Algorithm and Implementation characteristics.

AES cipher:

*) AES is a Non-fiestal cipher that encrypts and decrypts a data block of 128-bits.

*) The key size can be 128, 192 & 256 bits. It depends on no. of. rounds.

*) The no. of. Rounds: 10 rounds for 128-bits, 12 rounds for 192 bits and 14 rounds for 256 bits.

Characteristics:

*) Resistance against all known attacks.

*) Speed and code compactness on a wide range of platform.

*) Design Simplicity.

*) The input to the encryption and decryption algorithm is a single 128-bit block. The block is represented as a row of matrix of 16-bytes.

*) AES use several rounds in which each round is made of several stages.

Simple DES:

*) Takes an 8-bit block plaintext, a 10 bit-key and produces an 8-bit block of cipher text.

*) Decryption takes the 8-bit block of cipher text, the same 10-bit key and produces the original 8-bit block of plaintext.

*) It was designed as a test block cipher for learning about modern cryptanalysis technique such as linear cryptanalysis differential cryptanalysis and linear – differential cryptanalysis.

*) A input block to be encrypted is subjected to an initial permutation IP. Then, it is applied to two rounds of key dependent computation.

→ plaintext - $b_1 b_2 b_3 b_4 b_5 b_6 b_7$

→ key = $K_1 K_2 K_3 K_4 K_5 K_6 K_7 K_8 K_9 K_{10}$

Subkey generation:

First produce two subkeys K_1 and K_2:

$$K_1 = P8(LS_1(P10(key)))$$

$$K_2 = P8(LS_2(LS_1(P10(key))))$$

where P8, P10, LS_1, LS_2 are bit substitution operators.

For example; P10 takes 10 bits and returns the same 10 bits in a different order.

$$P10(k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}) = k_3k_5k_2k_7k_4k_{10}k_1k_9k_8k_6$$

Encryption:

The plaintext is split into 8 bit blocks, each block is encrypted seperately. Given a plaintext block, the cipher text is defined using two subkeys K_1 and K_2, as follows

$$ciphertext = IP^{-1}(f_{K_2}(SW(f_{K_1}(IP(plaintext)))))$$

where:

Initial Permutation (IP): 8 bits to 8 bits.

2	6	3	1	4	8	5	7

IP^{-1} :

4	1	3	5	7	2	8	6

Switch (SW): 8 bits to 8 bits.

5	6	7	8	1	2	3	4

and $f_k()$ is computed as follows.

We write ~~xor~~ XOR as $+$

$$f_k(L, R) = (L + F_k(R) + R)$$

$$F_K(R) = P4(S0(lhs(EP(R)+k)), S1(rhs(EP(R)+k))$$

4 bits to 8 bits.

4	1	2	3	2	3	4	1

P4

2	4	3	1

lhs

1	2	3	4

rhs.

5	6	7	8

Data Encryption Standard:

*) DES encryption is a symmetric key block cipher published by the National Institute of Standards and Technology.

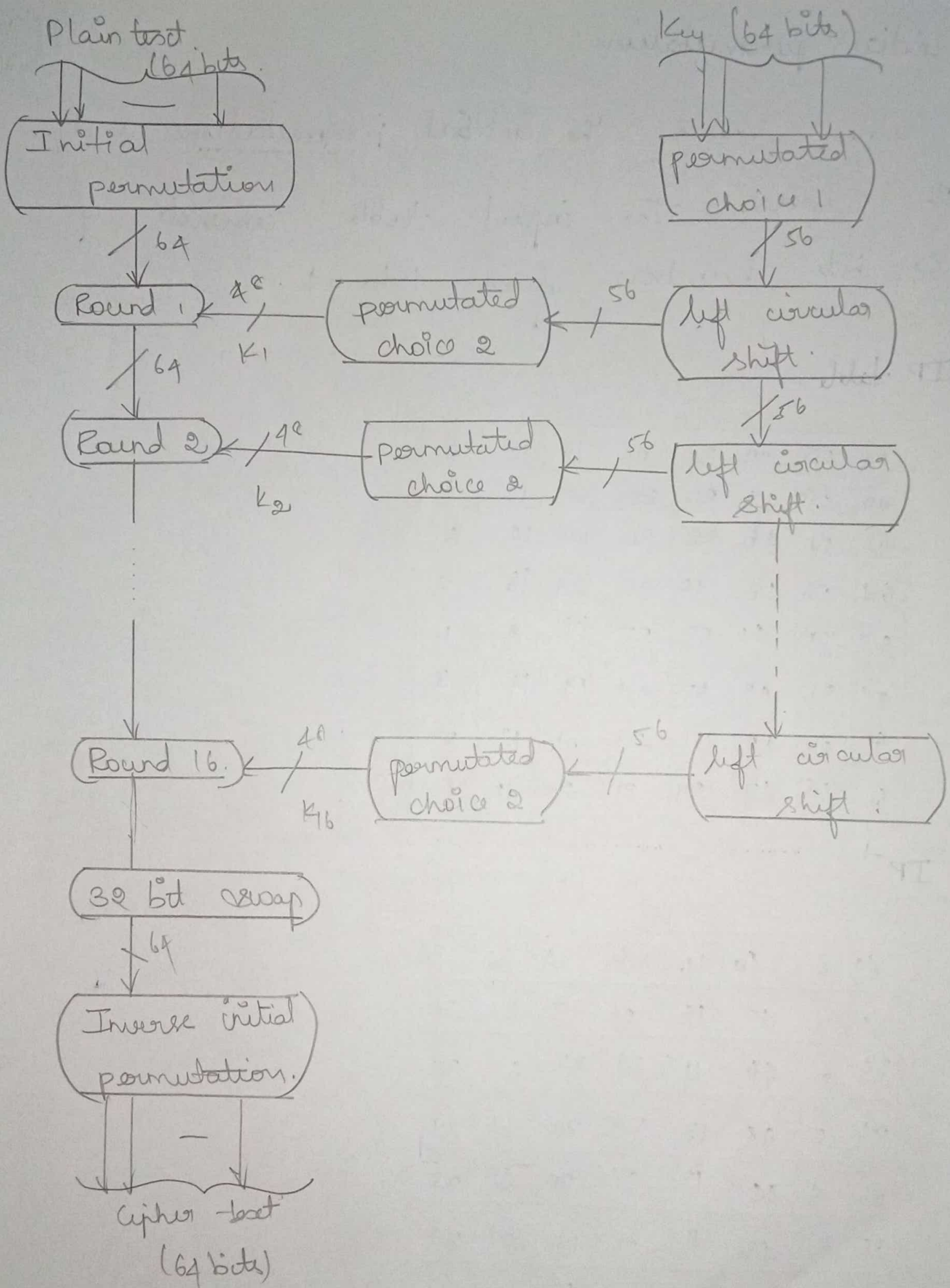
*) It encrypts data in 64-bit blocks. The same algorithm key is used for both encryption and decryption.

*) Key size is 56-bit.

*) The encryption process is made of 2 permutations.

*) DES uses both transposition and Substitution and for that reason is sometimes refferred to as a product cipher.

*) The cipher consists of 16 rounds or iterations. Each round uses a seperate key of 48-bits.



DES encryption Algorithm.

Initial permutation:

Table shows the initial permutation and its reverse. The input table consists of 64-bit numbers from 1 to 64.

IP table.

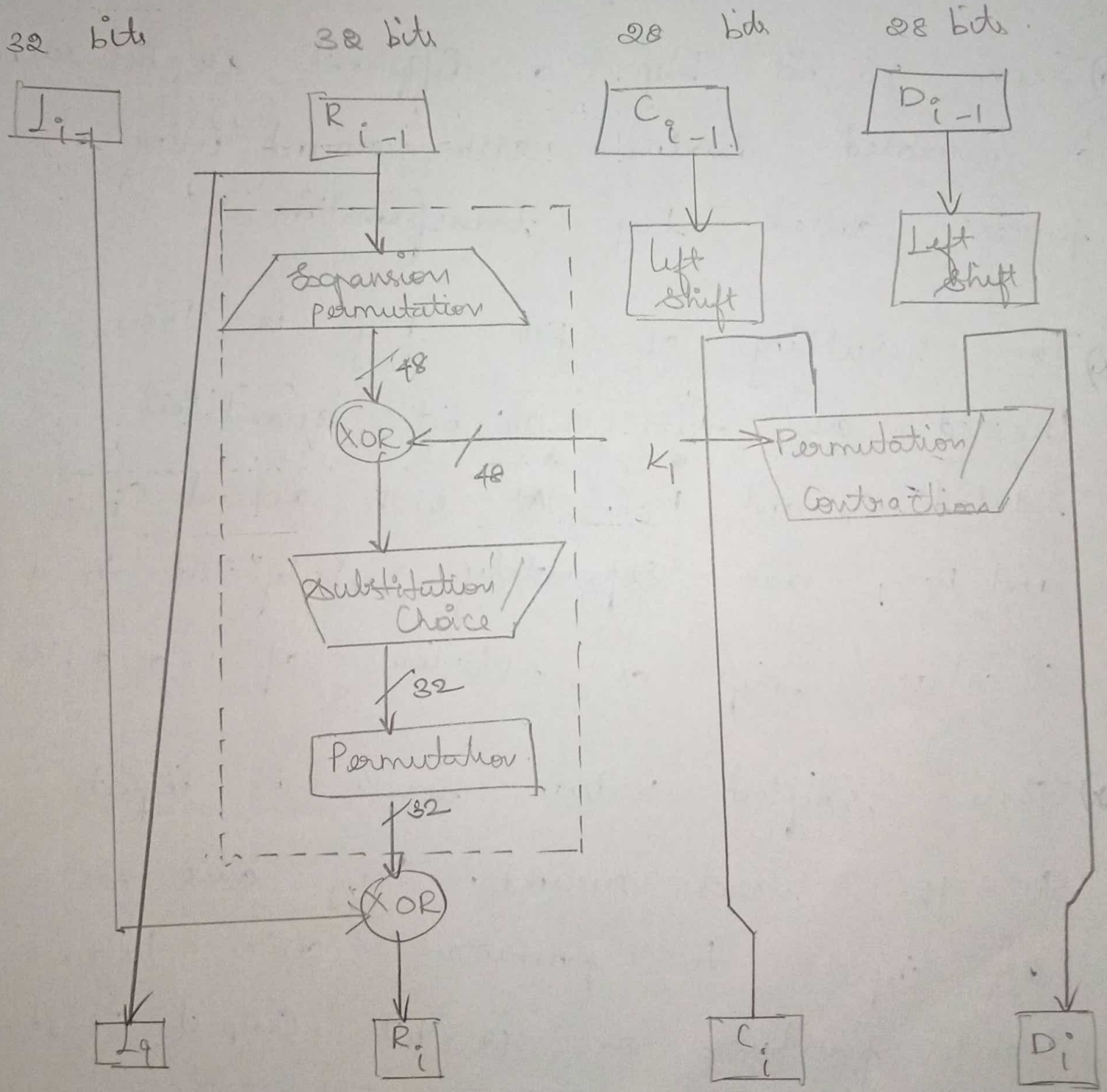
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Single Round DES:

The left and right halves of each 64-bits intermidiate value are treated as seperate 32-bit quantities, lables L and R.



Key Generation:

*) 64 - bit key is used as input algorithm The initial 64 bit key is transformed into 56 - bit key by discarding every 8th bit of initial key.

*) From 56 bit key, a different 48 - bit subkey is generated during each round using a process called key transformation.

*) The resulting 56 - bit key is then treated as two 28 - bit quantities, labeled C_0 and D_0. At each round C_{i-1} and D_{i-1} are separately subjected to a circular shift or rotation of 1 or 2 bits.

*) These shifted value serve as input to the next round. They also serve as input to permuted choice two, which produces as 48 - bit output that serves as i/p to the function.

Advantages of DES:

*) As 56-bit key are used there are 70 quadrillion possible key value. Hence specific cannot be identified easily.

*) The security of the DES algorithm resides in the key.

Disadvantages:

*) There is a possibility that the key is intercepted.

*) The design of s-box makes its susceptible to linear cryptanlysis attack

---

S-box Design criteria:

*) No o/p bit of any s-box should be too close a linear function of the i/p bits

*) Each row of an s-box should be include all the 16 possible o/p combination.

*) If a i/p to an sbox differ in the exactly one bit, the op must differ in at least two bits.

*) If a inputs to an s-box differ in the 2 middle bits exactly, the outputs must differ atleast two bits.

*) For any non zero 6-bits difference b/w inputs, no more than 8 of the 32 pairs of inputs.

Double DES:

*) Using two encryption stages and two keys

A) The plain to cipher text as follows.

$$C = E_{K_2}(E_{K_1}(P))$$ where K_1 and K_2 are key

B) cipher to plaintext;

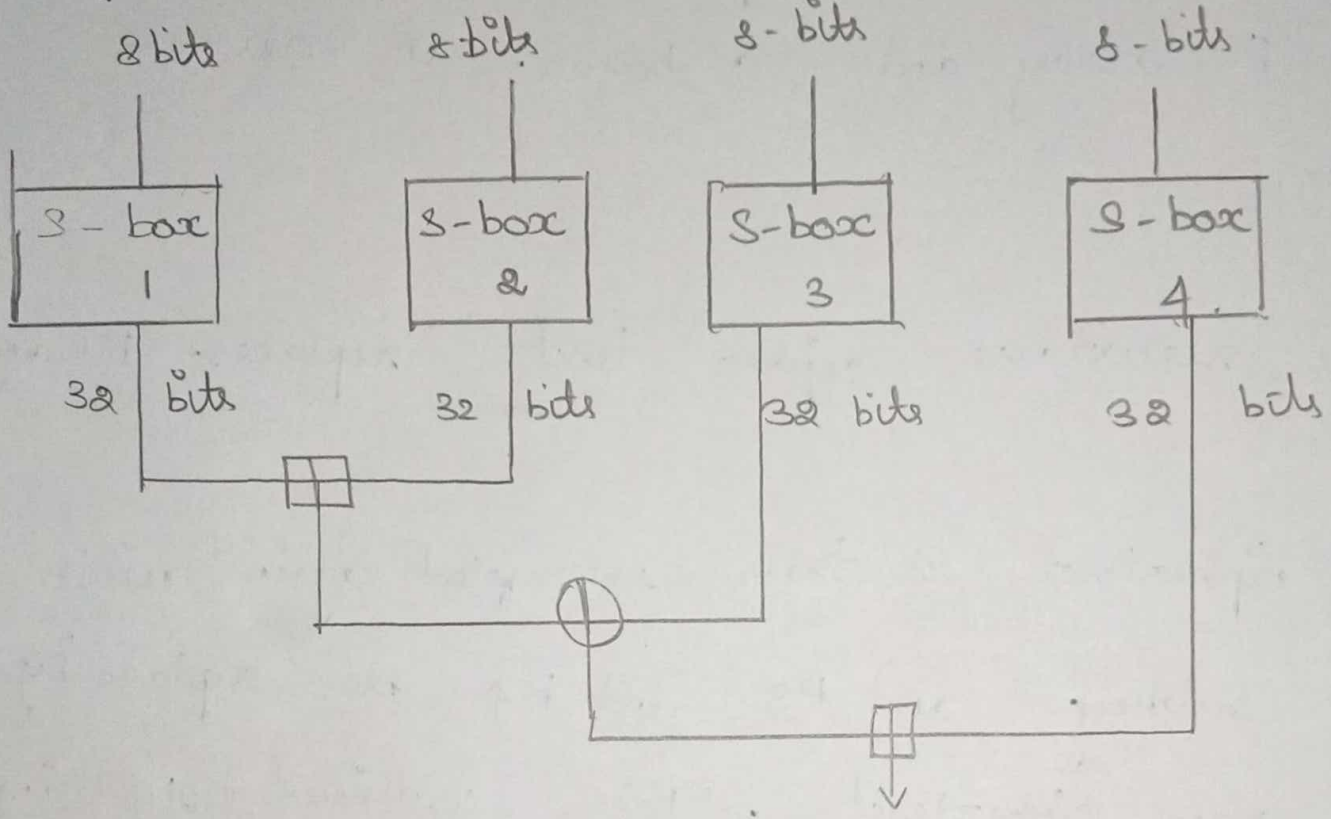
$$P = D_{K_1}(D_{K_2}(C))$$

*) Double DES suffers from Meet-in-the middle attack.

$S_{2,0}, S_{2,1}, \cdots S_{2,255}$

$S_{3,0}, S_{3,1}, \cdots, S_{3,255}$

Encryption.



* The function splits the 32 bits i/p into 4 eighth - bit quarters, and uses the quarters as i/p into S-boxes.

* The outputs are added modulo 232 and xored to produce the final 32-bit output.

*) Since Blowfish is a feistal network, it can be inverted simply by XORing P17 and P18 to the ciphertext block, then using the P-array and S-boxes in reverse order.

*) The resultant cipher text replaces P1 and P2.

*) The ciphertext is then encrypted again with new subkey and P3 and P4 are replace by the new ciphertext. This continues replacing the entire P-array and all S-box entries.

*) In all the Blowfish encryption algorithm will run 521 times to generate all the subkeys - About 4 KB of data is processed.

Unit - III

Public Key Cryptography

Mathematics of Asymmetric Key Cryptography

Primes:

* A prime number is an integer that can only be divided without remainder by positive and negative values of itself and 1.

* Any integer $a > 1$ can be factored in a unique way as

$$a = p_1^{a_1} p_2^{a_2} \ldots\ldots p_t^{a_t}$$

where $p_1 < p_2 < \ldots < p_t$ are prime numbers and where each a_i is a positive integer. This is known as the fundamental theorem of arithmetic.

* If P is the set of all prime numbers then any positive integer a can be written uniquely in the following form:

$$a = \prod_{p \in P} p^{a_p} \quad \text{where each } a_p \geq 0.$$

Relatively Prime Numbers:

* Two integers a and b are relatively prime if

$$\gcd(a, b) = 1.$$

* The integers $a_1, a_2, \ldots, a_n$ are pairwise relatively prime if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

* The method for calculating the number of relatively prime number less than a given number involves prime factorization, which can be reviewed in positive integral divisors.

1) Find the exponential prime factorization of the number.

2) Taking each term separately, change the term to 2 numbers:

a) Subtract 1 from the base for the first number.

b) Subtract 1 from the exponent and evaluate the expression for the second number.

3) Multiply all the numbers together found in step 2.

Example: How many numbers less than 20 are relatively prime to 20?

* The prime factorization of 20 is: $2^2 \times 5^1$

* Taking 2^2 first, we get : $2-1=1$ and $2^2-1=2$.

* Taking 5^1 we get : $5-1=4$ and $5^1-1=1$.

* Multiplying all of them together we get : (1)(2)(4)(1) or 8.

* The answer is 8. The numbers which are relatively prime are 1, 3, 7, 9, 11, 13, 17 and 19. So indeed there are 8.

Primality Testing :

Two properties of prime numbers :

1) If p is prime and a is a positive integer less than p, then $a^2 \bmod p = 1$ if an only if either $a \bmod p = 1$ or $\bmod p = -1$ and $p = p-1$. By the rules of modular arithmetic $(a \bmod p)(a \bmod p) = a^2 \bmod p$.

Thus if either $a \bmod p = 1$ or $a \bmod p = -1$ then $a^2 \bmod p = 1$. Conversely, if $a^2 \bmod p = 1$, then $(a \bmod p)^2 = 1$ which is true only for $a \bmod p = 1$ or $a \bmod p = -1$.

2) Let p be a prime number greater that 2. We can then write $p-1 = 2^k q$, with $k>0$, q odd. Let a be any integer in the range $1<a<p-1$. Then one of the following conditions is true.

a) a^q is congruent to 1 modulo p. That is a^q mod p = 1 or equivalently $a^q \equiv 1 \pmod{p}$.

b) One of the numbers $a^q, a^{2q}, a^{4q}, \ldots a^{2^{k-1}q}$ is congruent to -1 modulo p. That is, there is some number j in the range $(1 \le j \le k)$ such that $a^{2^{j-1}q} \bmod p = -1 \bmod p = p-1$ or equivalently, $a^{2^{j-1}q} = -1 \pmod{p}$.

Greatest Common Divisor:

* A positive integer d is called the greatest common divisor of the non zero integers a and b if

i) d is a divisor of both a and b, and

ii) Any divisor of both a and b is also a divisor of d.

* We will use the notation gcd (a, b), or simply (a, b) for the greatest common divisor of a and b.

* Greatest Common Divisor gcd (a, b) is the largest number that divides both a and b.

* If a and b share no common factors, they are called relatively prime.

Example: Find gcd (1403, 1081).

Solution:

$1403 = 1081(1) + 322$

$1081 = 322(3) + 115$

$322 = 115(2) + 92$

$115 = 92(1) + 23$

$92 = 23(4) + 0$

The last non-zero remainder is 23,

so, gcd (1403, 1081) = 23.

* It is always possible to write gcd (a, b) as a linear combinations of a and b. That is, there exist integers x and y such that gcd (a, b) = ax + by (x or y may be negative).

Euler's Totient Function:

* Euler's totient function (also called the Phi function) ~~counts~~ counts the number of positive integers less than n that are coprime to n. That is, $\phi(n)$ is the number of $m \in N$ such that $1 \leq m < n$ and gcd (m, n) = 1.

* The totient function appears in many applications of elementary number theory, including Euler's theorem, primitive roots of unity, cyclotomic polynomials, and constructible numbers in geometry.

* Can you find some relationships between n and $\phi(n)$?

When n is a prime number, $\phi(n) = n-1$.

* But how about the composite numbers?

Example, $15 = 3 * 5$ and $\phi(15) = \phi(3) \times \phi(5) = 2 \times 4 = 8$. This is also true for 14, 12, 10 and 6.

* However, it does not hold for 4, 8, 9. For example, $9 = 3 \times 3$, but $\phi(9) = 6 \neq \phi(3) \times \phi(3) = 2 \times 2 = 4$. In fact, this multiplicative relationship is conditional: When m and n are coprime, $\phi(m \times n) = \phi(m) \times \phi(n)$.

Fermat's and Euler's Theorem:

Fermat's theorem:

If p is prime and a is a positive integer not divisible by p, then $a^{p-1} \equiv 1 \bmod (p)$.

Proof: Consider the set of positive integers less that p: $\{1, 2, \ldots p-1\}$ and multiply each element by a, modulo p, to get the set $X = \{a \bmod p, 2a \bmod p, \ldots (p-1)a \bmod p\}$. None of the elements of X is equal to zero because p does not divide a. Further more no two of the integers in X are equal.

Euler's Theorem :

Euler's theorem states that for every a and n that are relatively prime :

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Proof : Equation is true if n is prime because in that case $\phi(n) = (n-1)$ and Fermat's theorem holds. It also holds for any integer n. Recall that $\phi(n)$ is the number of positive integers less that n that are relatively prime to n. Consider the set of such integers, labelled as follows :

$$R = \{x_1, x_2, x_{\phi(n)}\}$$

That is, each element x_i of R is a unique positive integer less than n with $\gcd(x_i, n) = 1$. Now multiply each element by a, modulation n :

$$S = \{(ax_1 \bmod n), (a x_2 \bmod n) \ldots (a x_{\phi(n)} \bmod n)\}$$

The set S is a permutation of R.

Therefore,

$$\prod_{i=1}^{\phi(n)} (ax_i \bmod n) = \prod_{i=1}^{\phi(n)} x_i$$

$$\prod_{i=1}^{\phi(n)} a x_i = \prod_{i=i}^{\phi(n)} x_i \pmod{n}$$

$$a^{\phi(n)} \times \left[\prod_{i=1}^{\phi(n)} x_i \right] = \prod_{i=1}^{\phi(n)} x_i \pmod n$$

$$a^{\phi(n)+1} = a \pmod n.$$

Chinese Remainder Theorem:

* Find a number x such that have remainders of 1 when divided by 3, 2 when divided by 5 and 3 when divided by 7. ie)

$$x = 1 \bmod 3$$
$$x = 2 \bmod 5$$
$$x = 3 \bmod 7$$

* Integers can be represented by their residues modulo a set of pair-wise relatively prime moduli. For example: In Z_{10}, integer 8 can be represented by the residues of the 2 relatively prime factors of 10 (2 and 5) as a ~~tuple~~ tuple (0, 3).

* Let $M = m_1 * m_2 x \cdots x m_k$, where m_i's are pairwise relatively prime, ie) $\gcd(m_i, m_j) = 1, 1 \le i \ne j \le k$.

* Assertion.

i) $A \Leftrightarrow (a_1, a_2, \cdots, a_k)$, where $A \in Z_m$, $a_i \in Z_{m_i}$ and $a_i = A \bmod m_i$ for $1 \le i \le k$.

a) One to one correspondance (bijection) between Z_M and the cartesian product $Z_{m_1} \times Z_{m_2} \times \cdots \times Z_{m_k}$.

b) For every integer A such that $0 \leq A < M$, there is a unique k-tuple $(a_1, a_2, \ldots, a_k)$ with $0 \leq a_i < m_i$.

c) For every such k-tuple $(a_1, a_2, \ldots, a_k)$, there is a unique A in Z_M.

d) Transformation from A to $(a_1, a_2, \ldots, a_k)$ is unique.

e) Computing A from $(a_1, a_2, \ldots, a_k)$ is done as follows:

1) Let $M_i = M/m_i$ for $1 \leq i \leq k$; ie) $M_i = m_1 \times m_2 \times \cdots \times m_{i-1} \times \cdots \times m_k$.

2) Note that $M_i \equiv 0 \pmod{m_j}$ for all $j \neq i$.

3) Let $c_i = M_i \times (M_i^{-1} \bmod m_i)$ for $1 \leq i \leq k$.

4) Then $A \equiv (a_1 c_1 + a_2 c_2 + \cdots + a_k c_k) \bmod M$.

5) $\leftarrow a_i = A \bmod m_i$, since $c_j \equiv M_j \equiv 0 \pmod{m_j}$ if $j \neq i$ and $c_i \equiv 1 \pmod{m_i}$.

* Operations performed on the elements of Z_M can be equivalently performed on the corresponding k-tuples by performing the operation independently in each co-ordinate position.

Example:

$A \leftrightarrow (a_1, a_2, \cdots, a_k)$, $B \leftrightarrow (b_1, b_2, \cdots, b_k)$

$(A+B) \bmod M \leftrightarrow ((a_1+b_1) \bmod m_1, \cdots (a_k+b_k) \bmod m_k)$

$(A-B) \bmod M \leftrightarrow ((a_1-b_1) \bmod m_1, \cdots (a_k-b_k) \bmod m_k)$

$(A \times B) \bmod M \leftrightarrow ((a_1 \times b_1) \bmod m_1, \cdots (a_k \times b_k) \bmod m_k)$

* CRT provides a way to manipulate numbers mod M in terms of tuple of smaller numbers.

Chinese remainder theorem:

Suppose $\gcd(m, n) = 1$. given a and b, there exists exactly one solution $x \pmod{mn}$ to the simultaneously congruence under certain conditions.

$$x \equiv a \pmod{m}, \quad x \equiv b \pmod{n}.$$

Proof:

* There exist integers s, t such that $ms + nt = 1$. Then $ms \equiv 1 \pmod{n}$ and $nt \equiv 1 \pmod{m}$. Let $x = bms + ant$. Then $x \equiv ant \equiv a \pmod{m}$ and $x \equiv bms \equiv b \pmod{n}$, as desired.

* Suppose x_1 is another solution. Then $x \equiv x_1 \pmod{m}$ and $x \equiv x_1 \pmod{n}$, so $x - x_1$ is a multiple of both m and n.

Exponentiation and Logarithm:

The exponential function, written $\exp(x)$ or e^x, is the function whose derivative is equal to its equation.

In other words:

If $y = e^x$, $\frac{dy}{dx} = e^x$.

If $y = e^{kx}$, $\frac{dy}{dx} = ke^{kx}$, where k is a constant.

Logarithms:

Discrete logarithms are fundamental to a number of public key algorithms, including Diffie-Hellman key exchange and the DSA.

The powers of an Integer, modulo n:

* Every a and n that are relatively prime:

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Where $\phi(n)$ Euler's quotient function, is the number of positive integers less than 'n' and relatively prime to 'n'.

* For more general expression:

$$a^m \equiv 1 \pmod{n}$$

If a and n are relatively prime, then there is at least one integer m that satisfies above equation, namely $m = \phi(n)$.

* The logarithm of a number is defined to be the power to which some positive base must be raised in order to equal the number. For base x and for a value y:

$$y = x^{\log_x(y)}$$

* The properties of logarithms include the following:

a) $\log_x(1) = 0$.

b) $\log_x(x) = 1$.

c) $\log_x(yz) = \log_x(y) + \log_x(z)$.

d) $\log_x(y^r) = r \times \log_x(y)$.

Euclidean Algorithm:

* One of the consequences of the Euclidean algorithm is as follows:

Given integers a and b, there is always an integral solution to the equation $ax + by = \gcd(a, b)$.

* Furthermore, the Extended Euclidean Algorithm can be used to find values of x and y to satisfy the equation above. The algorithm will look similar to the proof in some manner.

* Consider writing down the steps of Euclid's algorithm:

$$a = q_1 b + r_1, \text{ where } 0 < r < b$$

$$b = q_2 r_1 + r_2, \text{ where } 0 < r_2 < r_1$$

$$r_1 = q_3 r_2 + r_3, \text{ where } 0 < r_3 < r_2$$

$$\vdots$$

$$r_i = q_{i+2} r_{i+1} + r_{i+2}, \text{ where } 0 < r_{i+2} < r_{i+1}$$

$$\vdots$$

$$r_{k-2} = q_k r_{k-1} + r_k, \text{ where } 0 < r_k < r_{k-1}$$

$$r_{k-1} = q_{k+1} r_k$$

* Consider solving the second to last equation for r_k, you get $r_k = r_{k-2} - q_k r_{k-1}$ or

$$\gcd(a, b) = r_{k-2} - q_k r_{k-1}$$

Now, solve the previous equation for r_{k-1}:

$$r_{k-1} = r_{k-3} - q_{k-1} r_{k-2} \text{ and}$$

Substitute this value into the previous derived equation:

$$\gcd(a,b) = r_{k-2} - q_k (r_{k-3} - q_{k-1} r_{k-2})$$

$$\gcd(a,b) = (1 + q_k q_{k-1}) r_{k-2} - q_k r_{k-3}$$

Now we have expressed gcd(a,b) as a linear combination of r_{k-2} and r_{k-3}. Next we can substitute for r_{k-2} in terms of r_{k-3} and r_{k-4}, so that the gcd(a,b) can be expressed as the linear combination of r_{k-3} and r_{k-4}. Eventually, by continuing this process, gcd(a,b) will be expressed as a linear combination of a and b as desired.

Example: Using Euclidean algorithm, calculate GCD

GCD(48, 30)

$48 = 1 \times 30 + 18$, gcd(30, 18)

$30 = 1 \times 18 + 12$, gcd(18, 12)

$18 = 1 \times 12 + 6$, gcd(12, 6)

$12 = 2 \times 6 + 0$, gcd(6, 0)

Therefore,

$$\gcd(48, 30) = 6.$$

Asymmetric Key Ciphers:

❀ Diffie and Hellmen proposed a new type of cryptography that distinguished between encryption and decryption keys. One of the keys would be publicly known; the other would be kept private by its owner.

❀ These algorithms have the following important characteristic.

1) It must be computationally easy to encipher or decipher a message given the appropriate key.

2) It must be computationally infeasible to derive the private key from the public key.

3) It must be computationelly infeasible to determine the private key from a chosen plaintext attack.

❀ A public key encryption scheme has six ingredients.

1) Plaintext: It is input to algorithm and in a readable message or data.

2) Encryption algorithm:

It performs various transformations on the plaintext.

3) Public and private keys:

One key is used for encryption and other is used for decryption.

4) Ciphertext:

This is the scrambled message produced as output. It depends on the plaintext and the key.

5) Decryption algorithm:

This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

* The public key is accessed to all participants and private key is generated locally by each participant.

* System controls its private key. At any time, a system can change its private key.

Requirements of public key cryptography:

1) It is computationally easy for a party B to generate a pair.

2) It is computationally easy for a sender A, to generate the corresponding ciphertext: $C = E(PU_b, M)$

3) It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

$$M = D(PR_b, C) = D[PR_b, E(PU_b, M)].$$

4) It is computationally infeasible for an adversary, knowing the public key (PU_b) to determine the private key PR_b.

5) It is computationally infeasible for an adversary, knowing the public key (PU_b) and a ciphertext (C) to recover the original message (M).

Advantages of public key algorithm:

- Only the private key must be kept secret.
- A private / public key pair remains unchanged for considerable long periods of time.

Disadvantages of public key algorithm:

- Large key size.
- Lack of extensive history.
- No - asymmetric key scheme has been proven to be secure.

RSA Cryptosystem:

- RSA is a block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.
- A typical size for n is 1024 bits.
- The RSA algorithm developed in 1977 by Rivest, Shamir, Adleman (RSA) at MIT.
- RSA algorithm is public key encryption type algorithm.
- The details of the RSA algorithm are described as follows:

Key generation :

1) Pick two large prime numbers p and q, $p \neq q$.
2) Calculate $n = p \times q$;
3) Calculate $\phi(n) = (p-1)(q-1)$;
4) Pick e, so that $\gcd(e, \phi(n)) = 1$, $1 < e < \phi(n)$;
5) Calculate d, so that $d \cdot e \bmod \phi(n) = 1$, i.e) d is the multiplicative inverse of e in mod $\phi(n)$;
6) Get public key as $K_U = \{e, n\}$;
7) Get private key as $K_R = \{d, n\}$;

Encryption :

For plaintext block $P < n$, its ciphertext $C = P^e \bmod n$.

Decryption :

For ciphertext block C, its plaintext is $P = C^d \bmod n$.

Advantages :

1) RSA can be used both for encryption as well as for digital signatures.
2) Trapdoor in RSA is in knowing value of n but not knowing the primes that are factors of n.

Disadvantages :

1) If any one of p, q, m, d is known, then the other values can be calculated. So secrecy is important.
2) To protect the encryption, the minimum number of bits in n should be 2048.

Attacks on RSA:

Attacks on RSA algorithm are as follows:

1) Brute force.
2) Mathematical attacks.
3) Timing attacks.
4) Chosen ciphertext attacks.

Example: For the given values $p=19$, $q=23$, and $e=3$ find n, $\phi(n)$ and d using RSA algorithm.

Solution:

$$n = p \times q$$
$$n = 19 \times 23$$
$$n = 437$$
$$\phi(n) = (p-1) \times (q-1)$$
$$\phi(n) = 18 \times 22$$
$$\phi(n) = 396$$
$$ed = 1 \bmod \phi(n)$$
$$3(d) = 1 \bmod 396$$
$$d = \frac{1}{3}$$

Example: Perform encryption and decryption using RSA algorithm for $p=7$, $q=11$, $e=7$ and $M=9$.

Solution:

$$N = p \times q = 7 \times 11$$
$$N = 77$$
$$\phi(N) = (p-1) \times (q-1)$$

$$\phi(N) = (7-1) \times (11-1)$$

$$\phi(N) = 60$$

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

$$7 \times d = 1 \bmod 60$$

$$d = 43$$

Encryption, $C = M^e \pmod N$

$$= 9^7 \bmod 77$$

$$C = 37$$

Decryption, $M = C^d \pmod N$

$$= 37^{43} \pmod{77}$$

$$M = 9$$

Key Management and Distribution :

* The purpose of public key cryptography is

1) The distribution of public keys.

2) The use of public key encryption to distribute secret keys.

Distribution of Public keys:

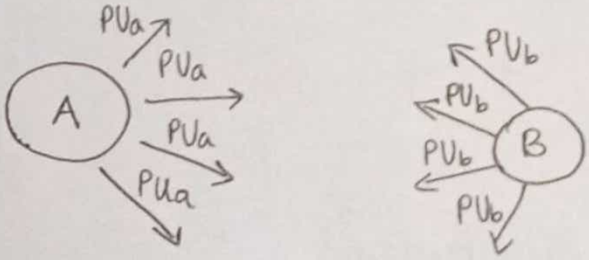
Different methods have been proposed for the

distribution of public keys. These are :

1) Public announcement.
2) Publicly available directory.
3) Public key authority.
4) Public key certificates.

1) Public announcement :

* In public key algorithm, any participant can send his or her public key to any other participant or broadcast the key to the community at large.

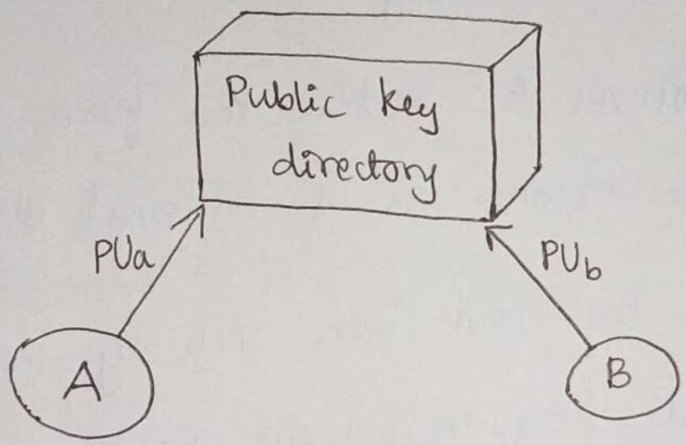


Public key distribution

* The disadvantage is that, anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key.

2) Public available directory:

Greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.



Public key publication

3) Public key authority:

Following steps occur in public key distribution.

1) A sends a timestamped message to the public key authority containing a request for the current public key of B.

2) The authority responds with a message that is encrypted using the authority's private key, PR_{auth}. The message also contains B's public key (PU_b), original request and timestamp.

3) A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1) which is used to identify this transaction uniquely.

4) B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

5) Public keys have been securely delivered to A and B and they may begin their protected exchange.

6) B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2).

7) A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

Drawbacks:

Public key authority could be somewhat of a

bottleneck in the system. The directory of name and public keys maintained by the authority is vulnerable to tampering.

4) Public key Certificates:

Certificates can be used by participants to exchange keys without contacting a public key authority. Certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.

* The third party is a certificate authority, such as government agency or a financial institution, that is trusted by the user community.

* A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate.

* Each participant applies to the certificate authority, suppling a public key and requesting a certificate.

For participant A, the authority provides a certificate of the form $C_A = E(PR_{auth}, [T \| ID_A \| PU_a])$

Where PR_{auth} is the private key used by the authority and T is a timestamp.

Distribution of Secret Keys using Public Key Cryptography:

Public key encryption provides for the distribution of secret key to be used for conventional encryption.

Simple Secret Key distribution:

If user A wishes to communicate with user B, the following procedure is employed:

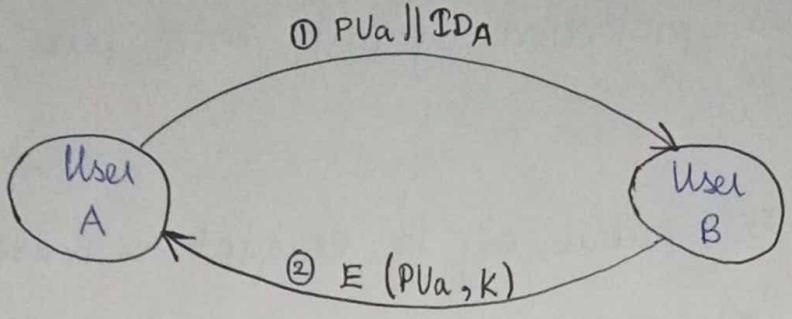
1) User A generates a public/private key pair $[PU_a, PR_a]$ and transmits a message to user B consisting of PU_a and an identifier of A, ID_A.

2) User B generates a secret key (K_s) and transmits it to user A, encrypted with A's public key.

3) User A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key. Because only A can decrypt the message, only user A and user B know the identity of K_s.

4) User A discards PUa and PRa and user B discards PUb.

5)

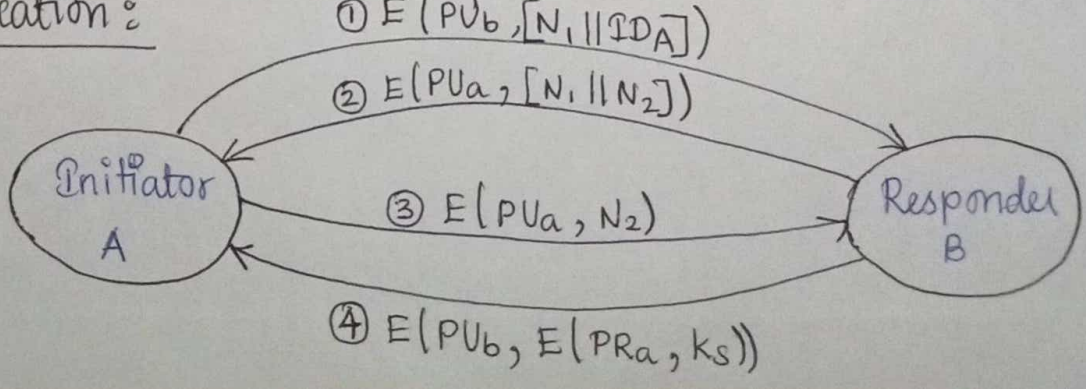


Use of Public key encryption

* User A and B can now securely communicate using conventional encryption and the session key Ks. At the completion of the exchange, both user A and B discard Ks.

* The protocol discussed above is insecure against an adversary who can intercept messages and then either relay the intercepted message or substitute another message. Such an attack is known as a man in middle attack.

~~Scret~~ Secret key distribution with confidentiality and authentication:



* This diagram shows the public key distribution of secret keys.

* It provides protection against both passive and active attacks.

1) A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.

2) B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B (N_2).

3) A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

4) A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B.

5) B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.

Key Distribution:

✱ Key distribution refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key.

✱ For two parties A and B, key distribution can be achieved in a number of ways, as follows.

1) User A can select a key and physically deliver it to user B.

2) A third party can select the key and physically deliver it to user A and user B.

3) If user A and user B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.

4) If user A and user B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to user A and user B.

✱ The use of a key distribution center is based on the use of a hierarchy of keys. Minimum two levels of keys are used.

- Communication between end systems is encrypted using a temporary key, often referred to as a session key.

- The session key is used for the duration of a logical connection, such as a frame relay connection, or transport connection and then discarded.

- session keys are transmitted in encrypted form, using a master key that is shared by the key distribution center and an end system or user. For each end user, there is a unique master key that is shares with the key distribution center.

A key distribution Scenario :

- User A wishes to establish a logical connection with user B and requires a one time session key to protect the data transmitted over the connection. User A has a master key (k_a), known only to itself and the KDC. User B shares the master key k_b with the KDC.

The following steps occur:

1) A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier (N_1) for this transaction.

2) KDC responds with a message encrypted using k_a.

3) A stores the session key for use in the upcoming session and forward to B the information that originated at the KDC for B:

4) User B sends a nonce N_2 to A.

Session key lifetime:

1) For connection-oriented protocol

* Use the same session key for the length of time that the connection is open. Use new session key for each new session.

* For long lifetime, change the session key periodically

2) For connectionless protocol:

The most secure approach is to use a new session key for each exchange. For connectionless protocol, such as a transaction-oriented protocol, there is no explicit

connection initiation or termination.

Transparent key control scheme:

- Assume that communication make use of a connection-oriented end-to-end protocol, such as TCP.

- Following steps occurs:

1) Host sends packet requesting connection.

2) Session Security Module (SSM) saves that packet and applies to the KDC for permission to establish the connection.

3) KDC distributes session key to both hosts.

4) The requesting SSM can now release the connection request packet, and a connection is set up between the two end systems.

Decentralized key control:

- Decentralized approach requires that each end system be able to communicate in a secure manner with all potential partner end systems for purposes of session key distribution.

* A session key may be established with the following sequence of steps.

1) A issues a request to B for a session key and includes a nonce, N_1.

2) B responds with a message that is encrypted using the shared master key.

3) Using the new session key, A returns $f(N_2)$ to B.

Diffie - Hellmen key Exchange:

* The Diffie - Hellmen key ~~generati~~ agreement protocol was developed by Diffie and Hellmen in 1976.

* This protocol allows two users to exchange a secret key over an insecure medium without any prior secrets.

Algorithm:

* Select two numbers (1) prime number q (2) α an integer that is a primitive root of q.

* Suppose the user A and B wish to exchange a key.

1) User A select a random integer $x_A < q$ and computes $Y_A = \alpha^{x_A} \bmod q$.

2) User B selects a random integer $X_B < q$ and compute $Y_B = \alpha^{X_B} \bmod q$.

3) Both side keeps the X value private and makes the Y value available publicly to the other side.

4) User A computes the key as $k = (Y_B)^{X_A} \bmod q$.

5) User B computes the key as $k = (Y_A)^{X_B} \bmod q$.

* Both side gets same results:

$$k = (Y_B)^{X_A} \bmod q = (\alpha^{X_B} \bmod q)^{X_A} \bmod q.$$

$$= (\alpha^{X_B})^{X_A} \bmod q = \alpha^{X_B X_A} \bmod q.$$

$$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q = (Y_A)^{X_B} \bmod q.$$

Advantages:

* Ideal for use with emails.
* Phone book must be maintained by a TTP.

Disadvantage:

Does not protect against man-in-the-middle attacks.

ElGamal :

* The ElGamal algorithm provides an alternative to the RSA for public key encryption.

1) Security of the RSA depends on the difficult of factoring large integers.

2) Security of the ElGamal algorithm depends on the difficulty of computing discrete logs in a large prime modulus.

* The ElGamal system is a public key algorithm so it has one set of key numbers that are published and another secret number that is used for deciphering.

1) The keys are generated by selecting a large prime number p. It is recommended that p-1 be divisible by another large prime.

2) Compute a generator number g and select a random integer "a" less than p-1.

3) With these numbers, compute $b = g^a \pmod p$.

4) The public key consists of the three number (p, g, b) and the secret key is the number a.

5) To find "a" given the public key, an attacker must be able to solve the discrete logarithm problem.

Encryption:

* If Bob wants to send a message to Alice he begins by looking up her public key (p, g, b) and representing the message as an integer m in the range 0 to p-1.

* He then selects a random key, k that is less than p-1.

* Using these numbers, Bob computes two numbers:

$$c_1 = g^k \text{ and } c_2 = mb^k$$

* He sends (c_1, c_2) to Alice.

Decryption:

* When Alice receives the cipher-text, she will recover the plaintext using her secret key, "a" to compute:

$$m = c_2 c_1^{-a} \bmod p$$

* This works because:

$$c_2 c_1^{-a} = mb^k (g^k)^{-a} \; mb^k (g^a)^k (g^k)^{-a}$$

$$= mg^{ak} g^{-ak} = m \pmod p.$$

* Bob should choose a different random integer k for each message he sends to Alice. If M is a longer message, so it is divided into blocks, he should choose a different k for each block.

Say he encrypts two messages M_1 and M_2, using the same k, producing ciphertents.

Eve intercepts both cipher-text messages and discovers one plaintent message M_1, she can compute the other plaintent message M_2.

Elliptic Curve Arithmetic:

Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithm that have applications in cryptography, such as Lenstra elliptic curve factorization.

Abelian Groups:

Abelian group G is denoted by $\{G, \cdot\}$. It is a set of elements with a binary operation.

Each ordered pair (a, b) of elements in G an element $(a \cdot b)$ in G, such that the following axioms are obeyed:

1) Closure: If a and b belong to G, then $a \cdot b$ is also in G.

2) Associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c in G.

3) Identity element: There is an element e in G such that $a \cdot e = e \cdot a = a$ for all a in G.

4) Inverse element: For each a in G there is an element a' in G such that $a \cdot a' = a' \cdot a = e$.

5) Commutative: $a \cdot b = b \cdot a$ for all a, b, in G.

For elliptic curve cryptography, an operation over elliptic curves, called addition, is used. Multiplication is defined by repeated addition. For example,

$$a \times k = \underbrace{(a + a + \cdots + a)}_{k \text{ times}}$$

Where the addition is performed over an elliptic curve. Cryptanalysis involves determining k given a and $(a * k)$.

Elliptical Curve Cryptography:

An elliptic curve is a set of points on the coordinate plane satisfying an equation of the form $y^2 + axy + by = x^3 + cx^2 + dx + e$. In order to use elliptic curves for say, Diffie-Hellmen, there needs to be some mathematical operation on two points in the set that will always produce a point also in the set.

* ECC can be done with atleast two types of arithmetic, each of which gives different definitions of multiplication. The two types of arithmetic are

1) Z_p arithmetic.

2) $GF(2^n)$ arithmetic, which can be done with shifts and $\oplus$s.

* To form a cryptographic system using elliptic curves, we need to find a hard problem corresponding to factoring the product of two ~~point~~ primes or taking the discrete logarithm.

* Consider the equation $Q = kP$ where $Q, P \in E_p(a,b)$ and $k < p$. It is relatively easy to calculate Q given k and P, but it is relatively hard to determine k given Q and P. This is called the discrete logarithm problem for elliptic curves.

Analog of Diffie-Hellman key exchange:

A key exchange between users A and B can be accomplished as follows:

1) A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the

public key is a point in $E_q(a,b)$.

2) B similarly selects a private key n_B and computes a public key P_B.

3) A generates the secret key $k = n_A \times P_B$.

B generates the secret key $k = n_B \times P_A$.

The two calculations in step3 produce the same result because

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G)$$

$$= n_B \times P_A$$

Elliptic curve encryption and decryption:

* For an encryption/decryption, system requires a point G and an elliptic group $E_q(a,b)$ as parameters. Each user A selects a private key n_A and generates a public key $P_A = n_A \times G$.

* To encrypt and send message P_m to user B, A chooses a random positive integer k and produces the ciphertext Cm consisting of the pair of points

$$C_m = \{kG, P_m + kP_B\}.$$

* To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from 2nd point,

$$= P_m + kP_B - n_B(kG) = P_m + k(n_B G) - n_B(kG) = P_m.$$

Unit - IV

Message Authentication and Integrity

Authentication and Authorization:

Authentication:

* Authentication techniques are used to verify identity. The authentication of authorized users prevents unauthorized users from gaining access to corporate information systems.

* Authentication method is of validating the identity of user, service or application. The use of authentication mechanisms can also prevent authorized users from accessing information that they are not authorized to view.

* Data authentication means providing data integrity as well as that the data have been received from the individual who claimed to supply this information.

Authorization:

* Authorization is a procedure of controlling the access of authenticated users to the system resources. An authorization system provides each user with exactly those rights granted to them by the administrator.

* Besides providing users with access rights to files, directories, printers etc, an authorization system might control user privileges, such as local access to the server, setting the system time, creating backup copies of the data and server shutdown.

Authentication Requirements:

* Attacks can be identified as follows:

1) Disclosure.
2) Traffic analysis.
3) Masquerade.
4) Sequence modification.
5) Content modification.
6) Timing modification.
7) Source repudiation.
8) Destination repudiation.

* Message authentication is a procedure to verify that received messages come form the alleged source and have not been altered.

* Digital signature is an authentication technique that also includes measures to counter repudiation by the source.

Authentication Function :

Functions are at two levels in message authentication. At the lower level, function that produces an authenticator. These value is used to authenticate a message. The lower level function is used in the higher level authentication protocol. The higher level authentication protocol enables a receiver to verify the authenticity of message.

Following are the some types of functions that may be used to produce an authenticator. They may be grouped into three classes.

1) Message encryption.
2) Message authentication code (MAC).
3) Hash function.

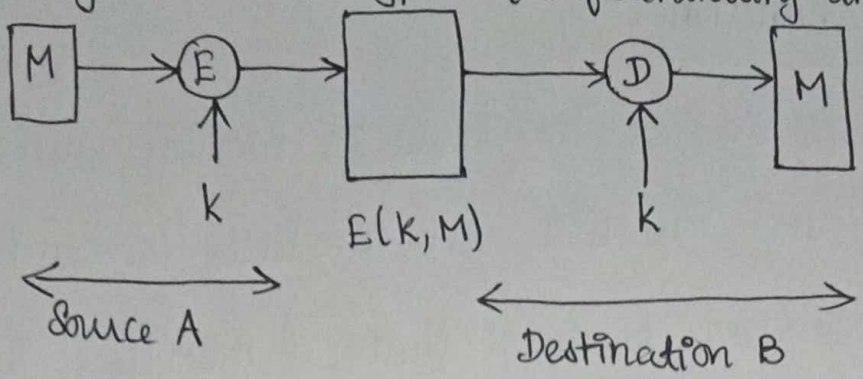
1) Message encryption :

Ciphertext of the entire message serves as its authenticator. Message encryption by itself can provide a measure of authentication.

Symmetric encryption :

* A message M transmitted from source A to destination B is encrypted using a secret key k shared by A and B. If no other party knows the key, then confidentiality is provided.

Symmetric encryption (confidentiality and authentication)



* Destination B is assured that the message was generated by A. Because of secret key used by both party, it provides authentication as well as confidentiality.

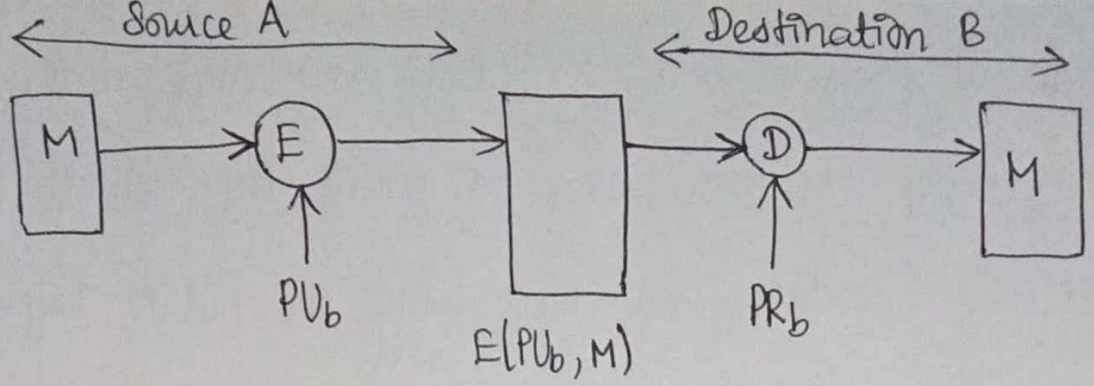
* Given a decryption function D and a secret key k, the destination will accept any input x and produce output $Y = D(k, X)$.

* If X is the ciphertext of a legitimate message M produced by the corresponding encryption function, then Y is some plaintext message M. Otherwise, Y will likely be a meaningless sequence of bits.

Public key encryption:

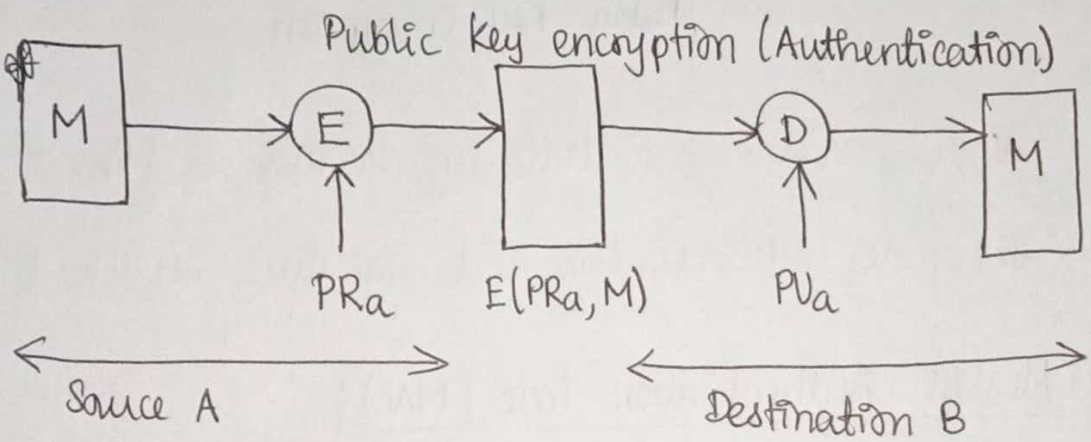
* Public key encryption provides confidentiality but not authentication.

* Source A uses the public key PU_b of the destination B to encrypt message M. Because only B has the corresponding private key PR_b, only B can decrypt the message.



Public key encryption (confidentiality)

* This method provides no authentication because any opponent could also use B's public key to encrypt a message, claiming to be A.

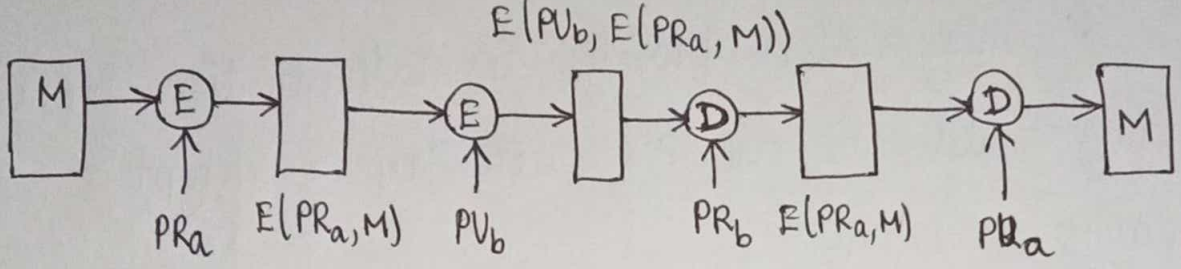


* A uses its private key to encrypt the message, and B uses A's public key to decrypt.

* It provides authentication. The message must have come from A because A is the only party that possesses PRa.

* It also provide digital signature. Only A could have constructed the ciphertext because only A possesses PRa. Not even B, the recipient could have constructed the ciphertext.

* To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature and then using B's public key, which provides confidentiality.



Public key encryption

* It provides confidentiality because of PU_b.
* Provides authentication and signature because of PRa.

2) Message Authentication Code (MAC):

* MAC is an alternative technique which uses secret key. This technique assumes that two communicating parties, share a common secret key k.

* When A has a message to sent to B, it calculates the MAC.

MAC = C(K, M) where

M = Message

C = MAC function

k = shared secret key

MAC = Message authentication code

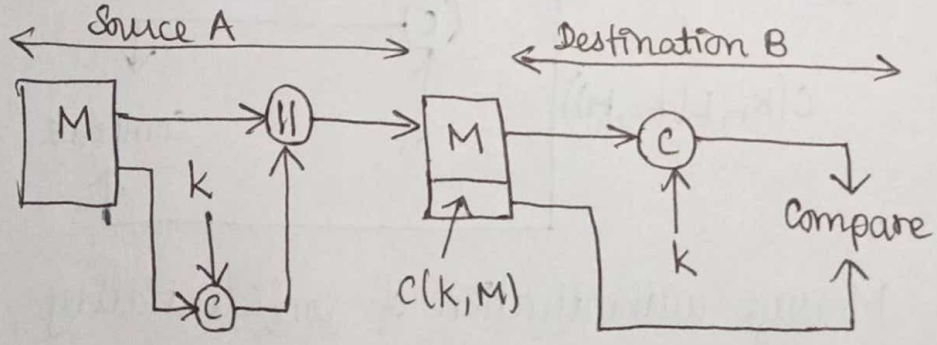
* Calculated MAC and message are transmitted to the receiver. The receiver performs the same calculation on the received message.

* Received MAC is compared with the calculated MAC. If both are matches, then

1) The receiver is assured that the message has not been altered.

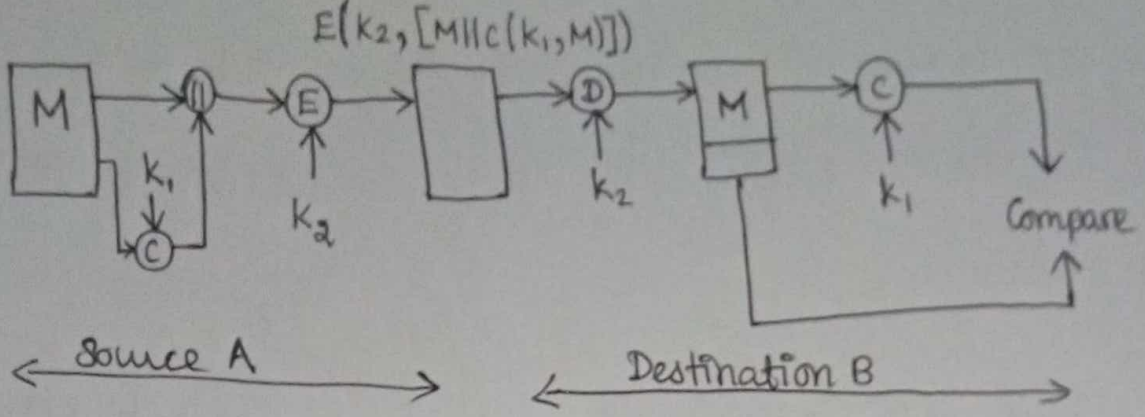
2) The receiver is assured that the message is from the alleged sender.

3) If the message includes a sequence number, then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.



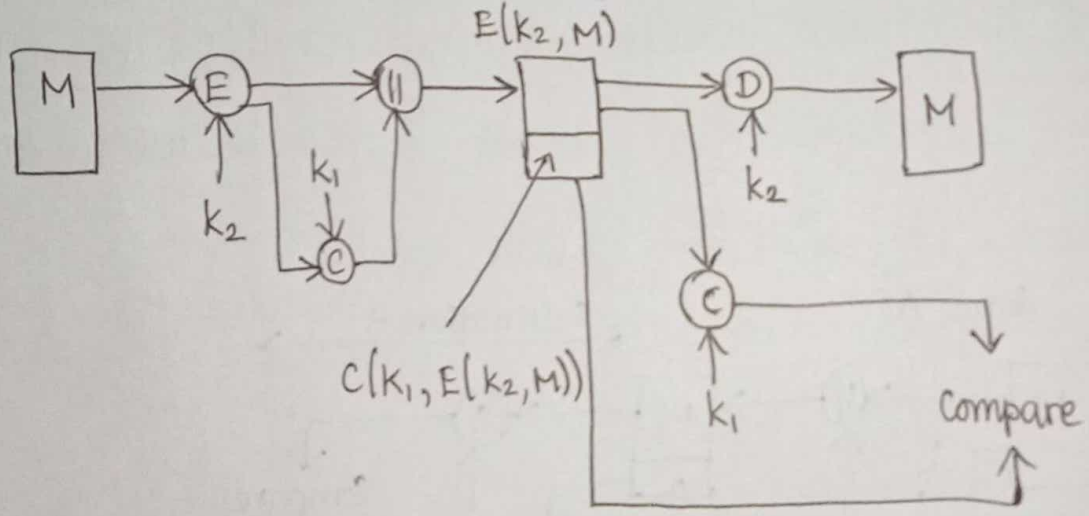
Message authentication

* It provides authentication but not confidentiality. Confidentiality can be provided by performing message encryption either after or before the MAC algorithm.



Message authentication and confidentiality

- It shows encryption after the MAC.
- Two separate keys are needed, each of which is shared by the sender and the receiver. Here MAC is calculated with the message input and is then concatenated to the message. The entire block is then encrypted.



Message authentication of confidentiality

- It shows the message authentication and confidentiality with encryption.
- Here also two separate keys are needed. The message is encrypted first. Then the MAC is calculated using the resulting ciphertext and is concatenated to the ciphertext to form transmitted block.

Applications of MAC:

Following are the situations in which MAC used.

1) Application in which the same message is broadcast to a number of destinations.

2) Authentication of a computer program in plaintext is an attractive sevice.

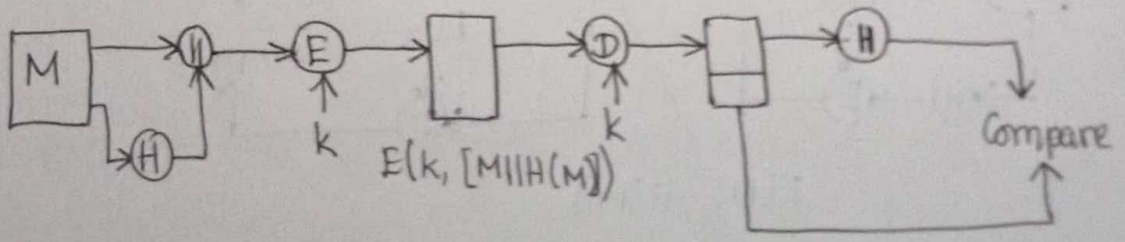
3) Another scenario is an exchange in which one side has a heavy load and cannot afford the time to decrypt all incoming messages.

3) Hash function:

* A hash function takes an input m, and computes a fixed size string known as a hash.

* Hash code is also referred to as a message digest or hash value.

* A change to any bit or bits in the message results in a change to the hash code.



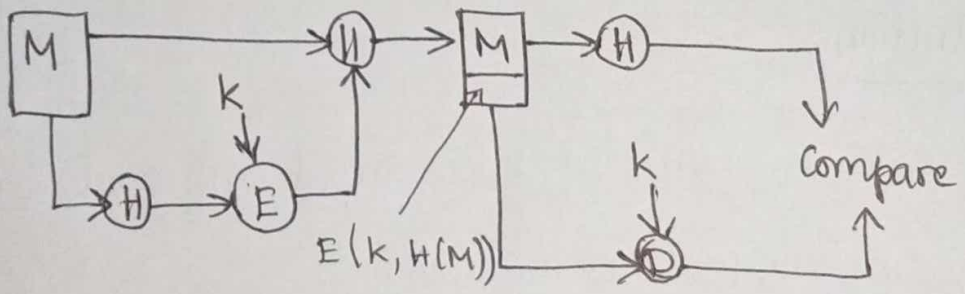
Encrypt message plus hash code

1) Encrypt message plus hash code :

- Provide confidentiality : Only A and B share K.
- Provide authentication : H(M) is cryptographically protected.

2) Encrypt hash code - shared secret key :

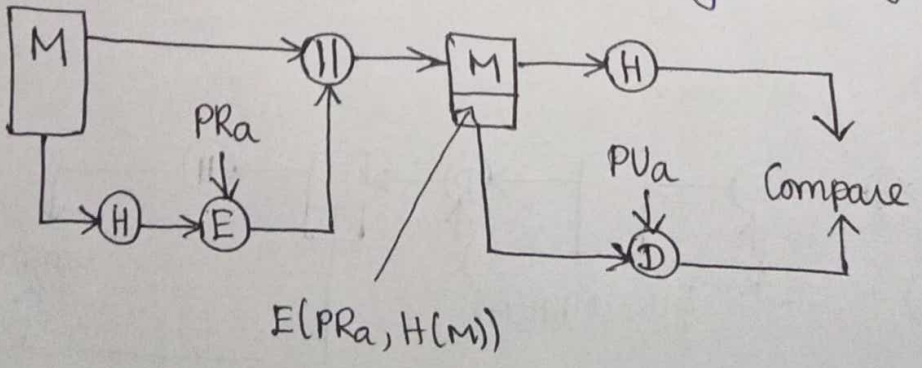
- Only the hash code is encrypted, using symmetric encryption.
- Reduces the processing burden for those applications that do not require confidentiality.



Encrypt hash code - shared secret key

3) Encrypt hash code - sender's private key :

Provides authentication and digital signature.



Encrypt hash code - sender's private key

SHA:

- The Secure Hash Algorithm (SHA) was developed by National Institute of Standards and Technology (NIST). It is based on the MD4 algorithm.
- Based on different digest lengths, SHA includes algorithms such as SHA-1, SHA-256, SHA-384 and SHA-512.

Features:

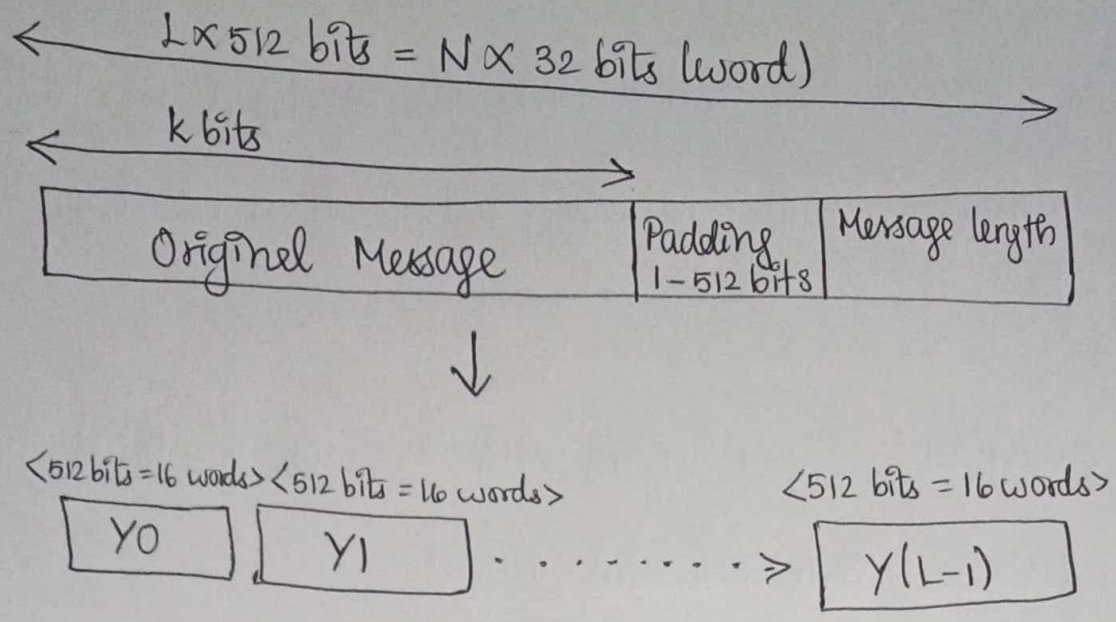
- The SHA-1 is used to compute a message digist for a message or data files that is provided as input.
- The message or data file should be considered to be a bit string.

SHA-1

- It works for any input message that is less than 2^{64} bits.
- The output of SHA is a message digest of 160 bits in length.
- This is designed to be computationally infeasible to:
  - Obtain the original message, given its message digest.
  - Find two messages producing the same message digest.

How SHA-1 works?

Step 1: Padding of bits.



Step 2 : Append Length.

Step 3 : Divide the input into 512-bit blocks.

Step 4 : Initialize chaining variables

Chaining Variables	Hex Values
A	01 23 45 67
B	89 AB CD EF
C	FE DC BA 98
D	76 54 32 10
E	C3 D2 E1 F0

Step 5 : Process blocks – Now the actual algorithm begins....

Step 5.1 : Copy chaining variables A-E into variables a-e.

Step 5.2 : Divide current 512-bit block into 16 sub-blocks of 32-bits.

Step 5.3: SHA has 4 rounds, each consisting of 20 steps. Each round takes 3 inputs -

512 bit block.

The register abcde.

A constant k[t] (where t=0 to 79).

Step 5.4: SHA has a total of 80 iterations (4 rounds × 20 iterations). Each iteration consists of following operations: -

$$abcde = (e + \text{Process P} + S^5(a) + W[t] + k[t]), a, S^{30}(b), c, d$$

Where

abcde = The register made up of 5 variables a, b, c, d, e.

Process P = The logic operation.

S^t = Circular left shift of 32 bit sub block by t bits.

W[t] = A 32 bit derived from the current 32-bit sub block.

k[t] = One of the five additive constants.

Process P in each SHA round

Round	Process P
1	(b AND c) OR ((NOT b) AND d)
2	b XOR c XOR d
3	(b AND c) OR (b AND d) OR (c AND d)
4	b XOR c XOR d

* The values of W[t] are calculated as follows:

- For the first 16 words of W (ie) t = 0 to 15), the contents of the input message sub-block M[t] become the contents of W[t].

- For the remaining 64 values of W are derived using the equation.

$$W[t] = S^1(W[t-16] \text{ XOR } W[t-14] \text{ XOR } W[t-8] \text{ XOR } W[t-3]).$$

Application:

* Secure password hashing.
* Secure Socket Layer (SSL) security protocol.
* Digital Signature.

Digital Signature:

A digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature. The signature is formed by taking the hash of the message and encrypting the message with the creator's private key.

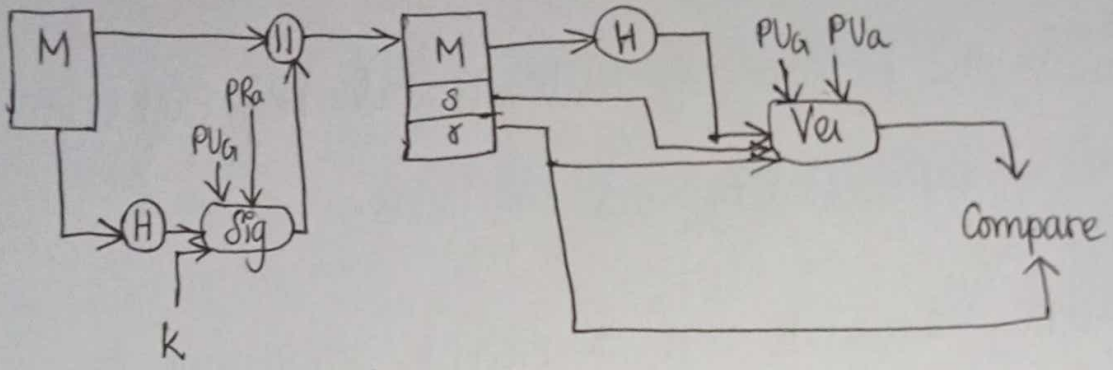
Two general schemes for digital signatures:

1) Direct.

2) Arbitrated.

Digital signature standard (DSS):

DSS makes use of the secure Hash Algorithm (SHA) and presents a new digital signature technique, the Digital Signature Algorithm (DSA). DSS cannot be used for encryption or key exchange.



DSS approach

* It uses a hash function. The hash code is provided as input to a signature function along with a random number k generated for this particular signature.

* The signature function also depends on the sender's private key (PRa) and a set of parameters known to a group of communicating principles.

* The result is a signature consisting of two components, labeled s and r.

* At the receiving end, the hash code of the incoming message is generated. This plus the signature is input to a verification function.

Authentication Protocol:

- Authentication protocols are used to convince parties of each others identity and to exchange session keys. They may be one way or mutual.
- Central to the problem of authenticated key exchange are two issues: confidentiality and timeliness.
- To prevent masquerade and to prevent compromise of session keys, essential identification and session key information must be communicated, in encrypted form.
- This requires the prior existence of secret or public keys that can be used for this purpose. The second issue, timeliness, is important because of the threat of message replays. Timeline prevent the repley attacks.

Entity Authentication:

- Entity authentication is the process by which one entity is assured of the identity of a second entity that is participating in a protocol.

❋ Properties of entity protocol:

- Reciprocity of Identification.
- Computational Efficiency.
- Communication Efficiency.
- Third-paut involvement.
- Security Guarantees.
- Storage of secrets.

Biometrics Authentication:

❋ Biometric authentication is simply the process of verifying you identity using you measurements or other unique characteristics of you body, then logging you in a sevice, an app, a device and so on.

❋ Biometric identification verifies you are you based on you body measurements.

❋ Biometric identification systems can be grouped based on the main physical characteristics that leads itself to biometric identification.

* Some of the biometrics identifications are:
  - Fingerprint identification.
  - Hand geometry.
  - Retina Scan.
  - Iris scan.
  - Face recognition.
  - Signature.
  - Voice analysis.

Password authentication:

* Password authentication is also called week authentication.

* This is amongst the most conventional schemes where in a user has an "user id" and a "password". User id acts like a claim and password as evidence supporting the claim.

* The system checks to see if it matches or not. Here demonstration of knowledge of the secret which is password in this case, corroborates that the person is verified.

Advantages:

* It has better entropy then a short password.
* It is easier to remember than the usual passwords.

Disadvantages:

- This is really weak against attacks as intruders can hear one communication channel and impersonate it later.
- It is also very easy to replay the same message and use it later.

Challenge - Response Identification:

* It is also called strong authentication.
- The central idea of challenge response is that claimant proves its identity to verifier by demonstrating knowledge of a secret known to be associated with entity without revealing the secret itself to the verifier during the protocol.
- The challenge is usually time variant and is random number.
- As every time the challenge is different, even if the adversary is monitoring the network it won't help as challenge changes every time.
- Challenge - response authentication uses a cryptographic protocol that allows to prove that the user knows the password without revealing the password itself.

* Using this method, the application first obtains a random challenge from the server.

* It then computes the response by applying a cryptographic hash function to the server challenge combined with the user's password.

* Finally, the application sends the response along with the original challenge back to the server. Because of the "one-way" properties of the hash function, it is impossible to recover the password from the response sent by the application.

* Upon receiving the response, the server applies the same hash function to the challenge combined with its own copy of the user's password. If the resulting value matches the response sent by the application, this indicates with a very high degree of probability that the user has submitted the correct password.

* Challenge-response by symmetric-key techniques. Here, A is the claimant and B is the verifier. The communication takes place as

$$A \leftarrow B : r_B$$

$$A \rightarrow B : E_K(r_B, B^*)$$

* B sends A a random number. To prove its claim, A then encrypts the random number send by B using the symmetric encryption key K.

* It also sends the optional field of the verifier as B. This prevents reflection attack as the key used is bi-directional key K.

* B then decrypts the message sent by A to see the random number is the same as it had sent. It also sees if the identifier matches. If either of them is not true it stops any further communication.

Authentication Applications:

Kerberos:

* Kerberos is an authentication protocol. It provides a way to authenticate clients to sevices to each other through a trusted third party.

* Kerberos makes the assumption that the connection between a client and sevice is insecure. Passwords are encrypted to prevent others from reading them. Clients only have to authenticate once during a pre-defined lifetime.

* Kerberos was designed and developed at MIT by Project Athena. Currently, Kerberos is upto Version 5. Version 4 being the first version to be released outside of MIT.

* Kerberos has been adopted by several private companies as well as added to several operating systems.

* Its creation was inspired by client-server model replacing time-sharing model. Kerberos is a network authentication protocol designed to allow users, clients and servers, authenticate themselves to each other.

* This mutual authentication is done using secret-key cryptography with parties proving to each other their identity across an insecure network connection.

* Communication between the client and the server can be secure after the client and server have used Kerberos to prove their identity.

* From this point on, subsequent communication between the two can be encrypted to assure privacy and data integrity.

Requirement of kerberos:

* kerberos client / server authentication requirements are:
  - Security.
  - Reliability.
  - Transparency.
  - Scalability.

* To meet these requirements, kerberos designers proposed a third-party trusted authentication service to arbitrate between the client and server in their mutual authentication.

Strengths of kerberos:

* Shared secret keys between clients and services are more efficient than public-keys.

* Authenticators, created by clients, can only be used once. This features prevents the use of stolen authenticators.

Weakness of kerberos:

* kerberos only provides authentication for clients and services.

* Kerberos 4 uses DES, which has been shown to be vulnerable to brute-force-attacks with little computing power.

X.509 Authentication Sewices:

* X.509 is part of X.500 recommandations for directory sewice ie) set of sewers which maintains a database of information about uses and other attributes.

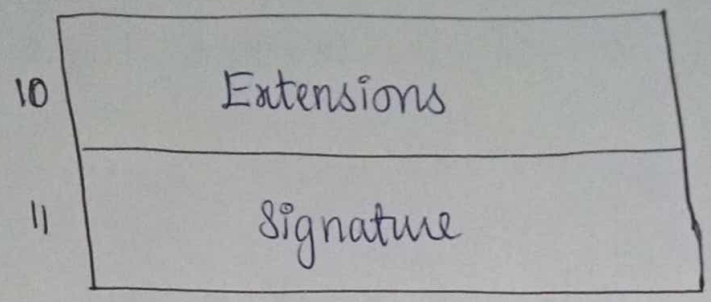
* X.509 defines authentication sewices eg) certificate structure and authentication protocols. Also X.509 also defines alternative authentication protocols base on use of public-key certificates. The X.509 certificate format is emplied in S/MIME, IP security, SET and SSL/TLS.

* X.509 standard uses RSA algorithm and hash function for digital signature.

X.509 Format of Certificate:

* The current version of the standard is version 3, called as X.509v3.

1	Version
2	Certificate Serial Number
3	Signature Algorithm Identifier
4	Issuer Name
5	Period of Validity
6	Subject Name
7	Subject's Public key Info
8	Issuer Unique Identifier
9	Subject Unique Identifier



X.509 Digital certificate format version 3

Standard notations for defining a certificate :

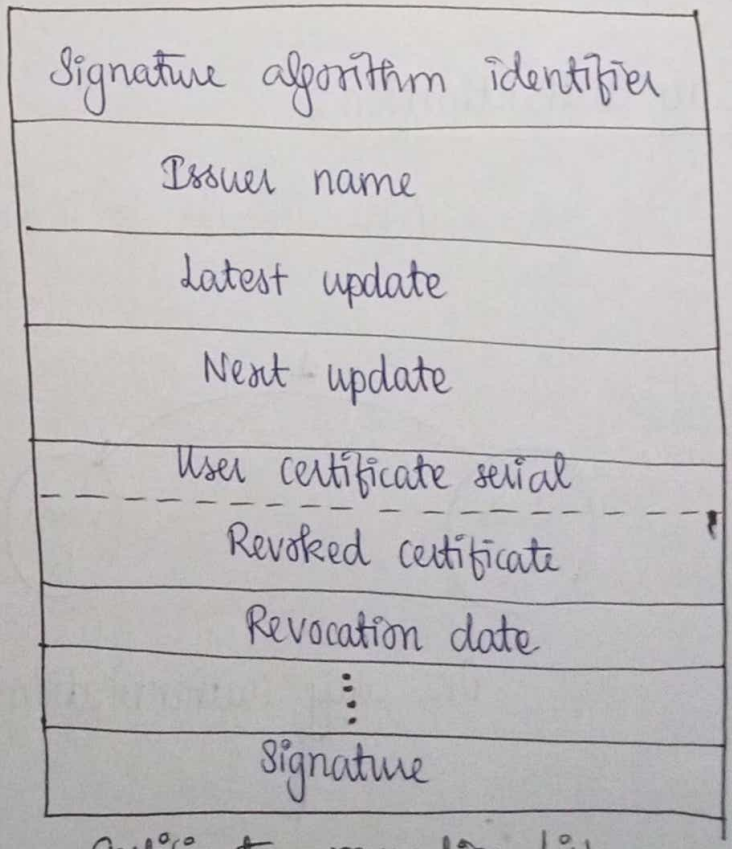
$CA \ll A \gg = CA \{ V, SN, AI, CA, T_A A, A_P \}$

Where

$CA \ll A \gg$ indicates the certificate of user A issued by certification authority CA.

$CA \{V \ldots A_P\}$ indicates signing of $V \ldots A_P$ by CA.

Revocation of Certificates :



Certificate revocation list

* The certificate should be revoked before expiry because of following reasons:

1) User's private key is compromised.
2) User is not certified by CA.
3) CA's certificate is compromised.

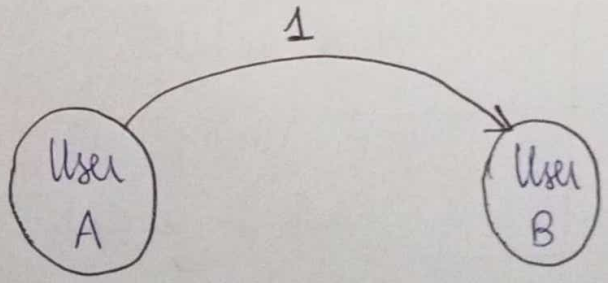
Authentication Procedures:

X.509 supports three types of authenticating using public key signatures. The types of authentication are:

1) One-way authentication.
2) Two-way authentication.
3) Three-way authentication.

1) One-way authentication:

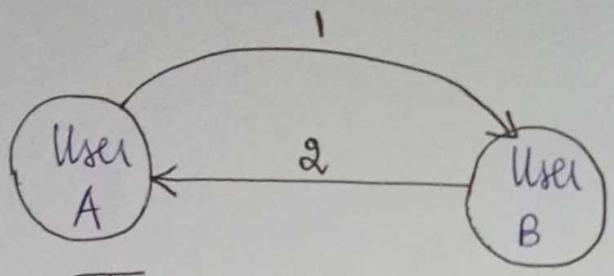
It involves single transfer of information from one user to other.



One way authentication

2) Two-way authentication:

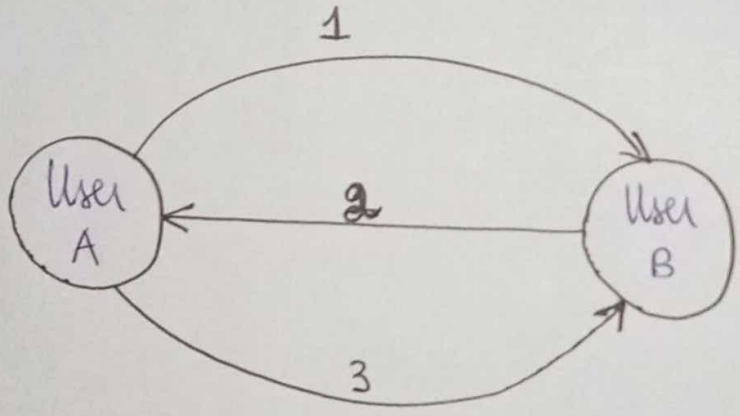
Two-way authentication allows both parties to communicate and verify the identity of the user.



Two-way authentication

3) Three-way authentication:

Three-way authentication is used where synchronized clockes are not available.



Three-way authentication

Unit-5

Security practice and System Security

Electronic mail security

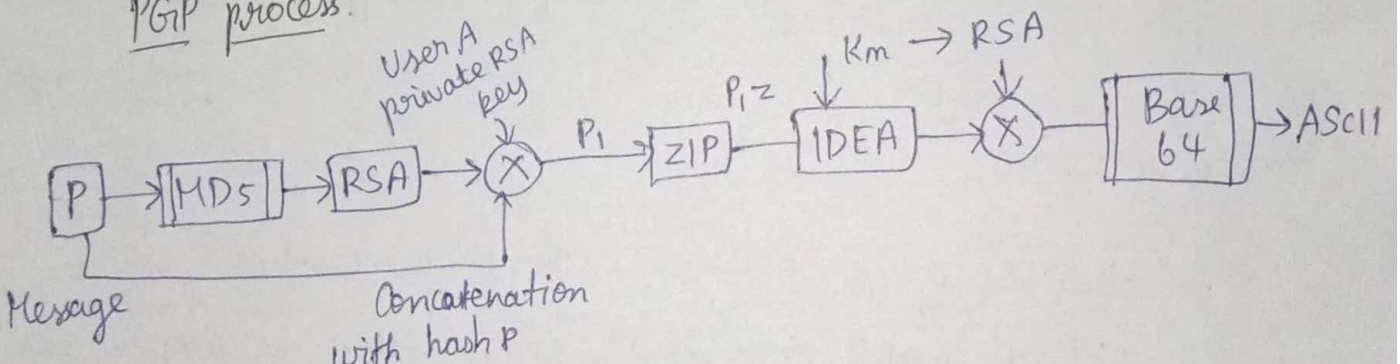
Pretty good privacy (PGP)

→ open source
→ It is a complete e-mail security package
→ encrypts data by using a block cipher called IDEA, which uses 128-bit keys.

Characteristics of PGP:

* free world wide
* run on various platform
* secure
* Internet Standards track
* World wide acceptability

PGP process:



Notations:

K_s, PR_a, PU_a, EP, DP, EC, DC, H, ||, z, Rb4

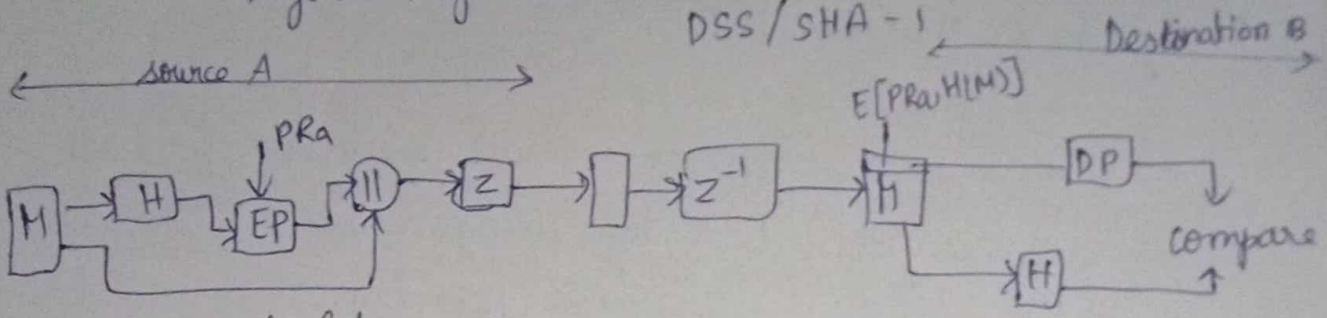
PGP operation:

1. Authentication:

Signatures are attached to the message or file are detached signatures are also supported & stored

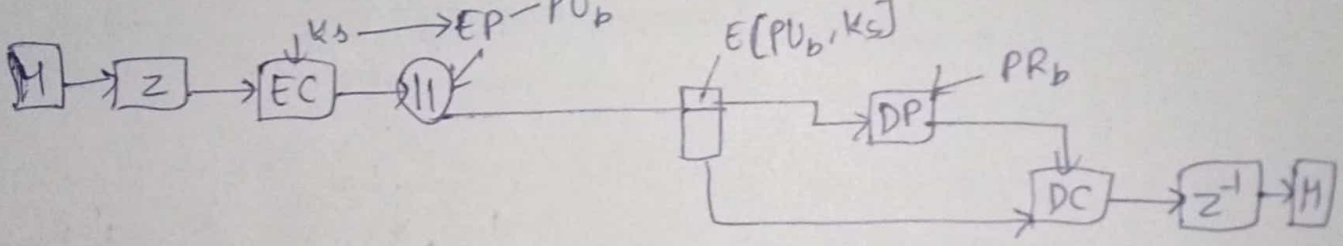
→ transmitted separately from the message it signs

→ Digital signature – SHA-1 & RSA
DSS / SHA-1



2. Confidentiality:

→ encrypting message to be transmitted.
→ Algorithms used are CAST-128, IDEA, 3DES
→ Only a portion of plaintext is encrypted
→ Service can be used for encrypting disk files.



3. Compression:

→ saves space and ease of transmission.
→ PGP makes use of a compression package called ZIP.
→ It is achieved with the ZIP algorithm.

4. E-mail compatibility:

When PGP is used, atleast part of the block to be transmitted is encrypted.

5. Segmentation & reassembly:

provides subdivision of msgs & reassembly at the receiving end.

Cryptographic keys & key rings:

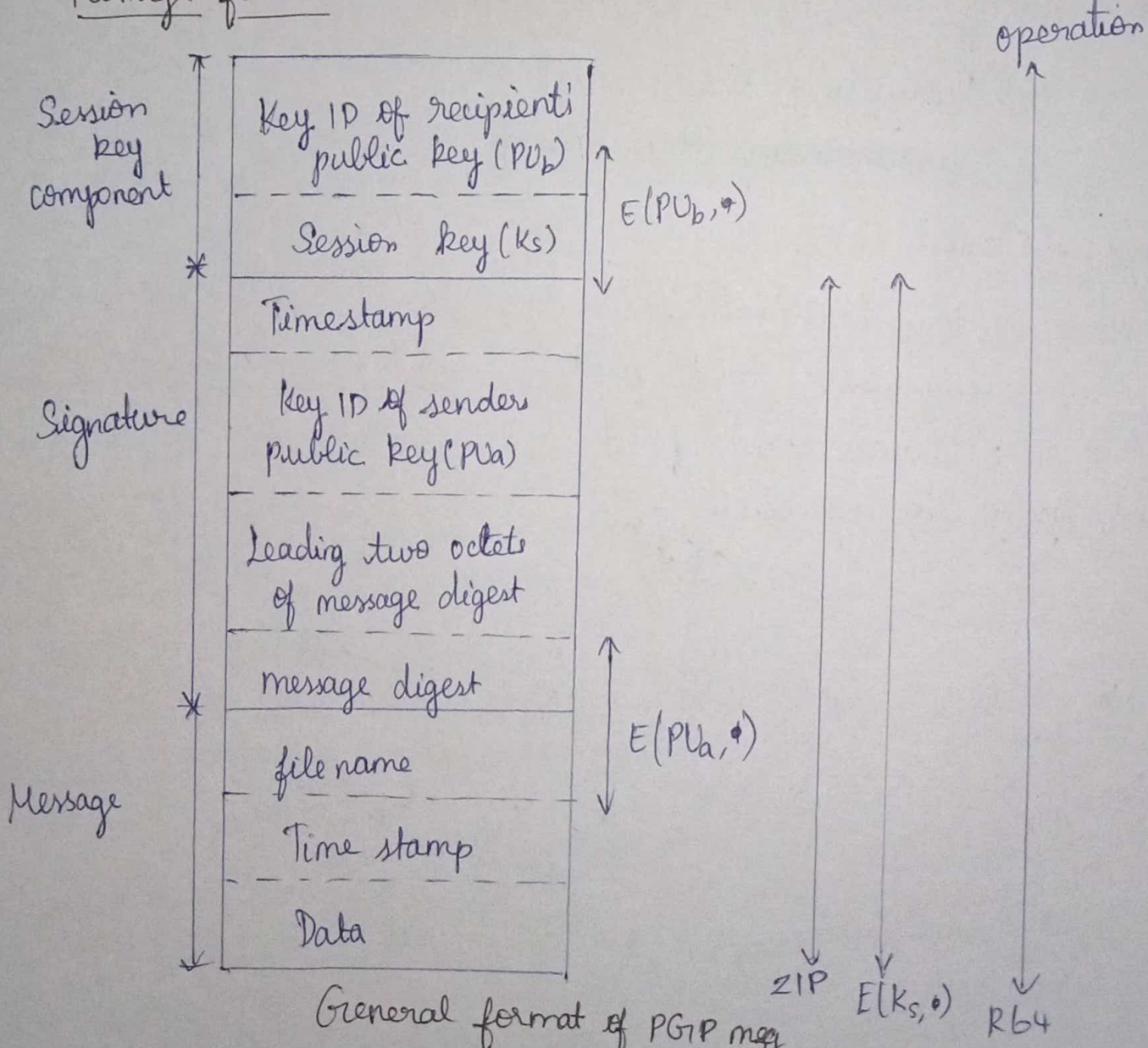
PGP makes use of four types of keys

* public keys
* private keys
* One time session conventional keys
* Passphrase based conventional keys

Key management:

* These separate requirements can be identified
* PGP protocol solves this problem by using the notion of a relatively short key identifiers.

Message format:



General format of PGP msg

Three component:

Session key component, Signature component & actual email message.

Message component:

→ Timestamp
→ Key ID of sender's public key
→ Leading two octets of message digest
→ Message digest.

PGP message generation:

→ Signs the message
→ Encrypts the message

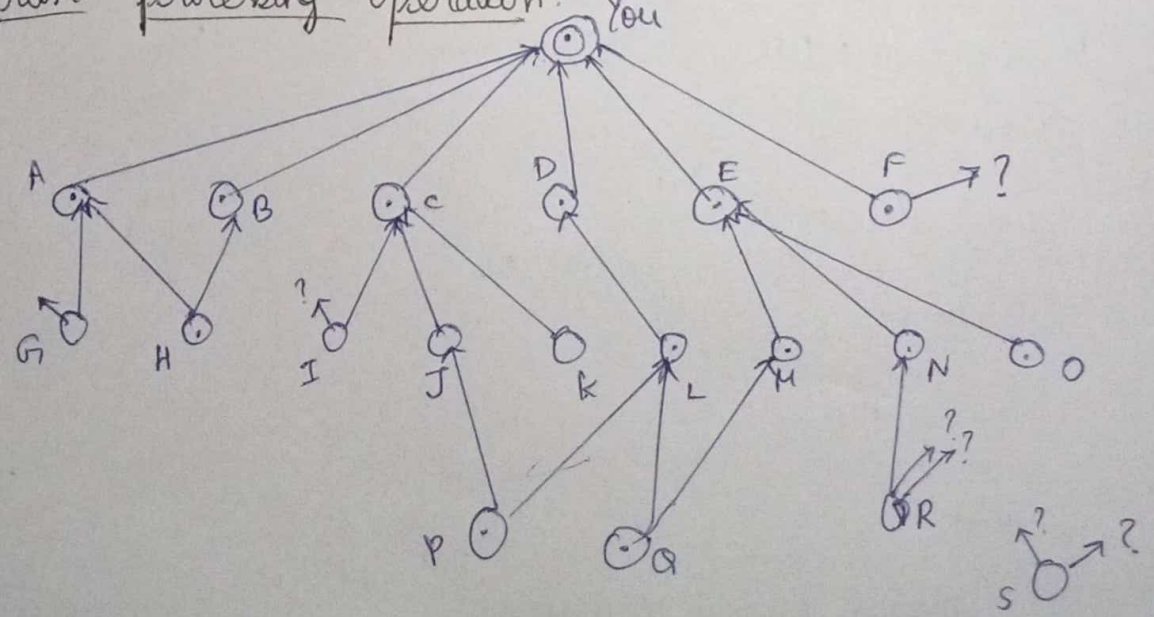
PGP message reception:

→ Decrypting the message
→ Authenticating the message

Concept of trust:

1. Key legitimate field
2. Signature trust field
3. Owner trust field.

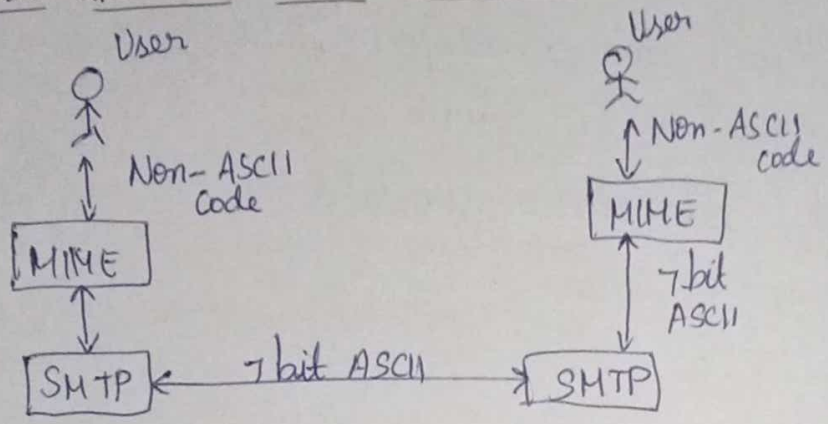
Trust processing operation:



S/MIME:

→ Secure
→ Multipurpose Internet Mail Extension
→ extension to the RFC 822 framework
→ format of text messages

Multipurpose internet mail extensions: (MIME)



MIME

Types:

Text
Multipart
Video
Audio
Image
Message
Application

Content-Transfer encoding:

Types:

7 bit
8 bit
Binary
Base 64
Quoted printable

Message headers:

Include the addresses of the receiver and the sender.

S/MIME functionality:

Enveloped data
signed data
clear signed data
Signed and enveloped data.

Cryptographic algorithms in S/MIME

Digital signature standard
Diffie Hellman
Triple DES

RFC 2119

MUST
SHOULD

S/MIME Messages:

1. Securing a MIME entity:

→ secures a MIME entity with a signature, encryption or both.

→ prepared according to the normal rules

→ prepared by using MIME entity plus

→ converted to canonical.

2. Enveloped data:

Steps: 1. Generate a pseudorandom session key
2. encrypt the session
3. encrypt the message
4. enveloped data is encoded.

3. Signed data:

Steps: 1. select a message
2. Compute the message
3. encrypt the message
4. prepare a block

4. Clear signing:

→ achieved using the multipart content type
→ MIME type
→ MIME content type

5. Registration request:

→ certification request info block
→ identifier of the public key encryption algorithm
→ Signature of the certification RequestInfo block.

S/MIME certificate processing:

User agent:

→ Key generation A user agent
→ Registration
→ Certificate storage and retrived.

Verisign certificates:

→ Owners public key
→ Owners name
→ Expiration data of the Digital ID
→ Serial number
→ Name of the certification
→ Digital signature

PEM:

* primary goal → add security services
* consists of extensions to existing message
* Developed by IETF
* Uses symmetric cryptography
* use of X.509 certificate

Security services:

→ Integrity
→ Authenticity
→ Non-repudiation
→ Confidentiality

PEH message processing:

Steps: 1. Uses the canonicalization
2. MIC is calculated
3. Renders an Encrypted or MIC-only

Overview of IPSec:

Two modes: 1. Transport mode
2. Tunnel mode

Two protocols: 1. Authentication Header protocol
2. Encapsulating security payload protocol.

Applications of IPSec:

1. Secure connectivity over the internet
2. Secure remote access over the internet
3. Extranet & intranet connectivity
4. Enhanced electronic-commerce security

Benefits of IPSC

* strong security.

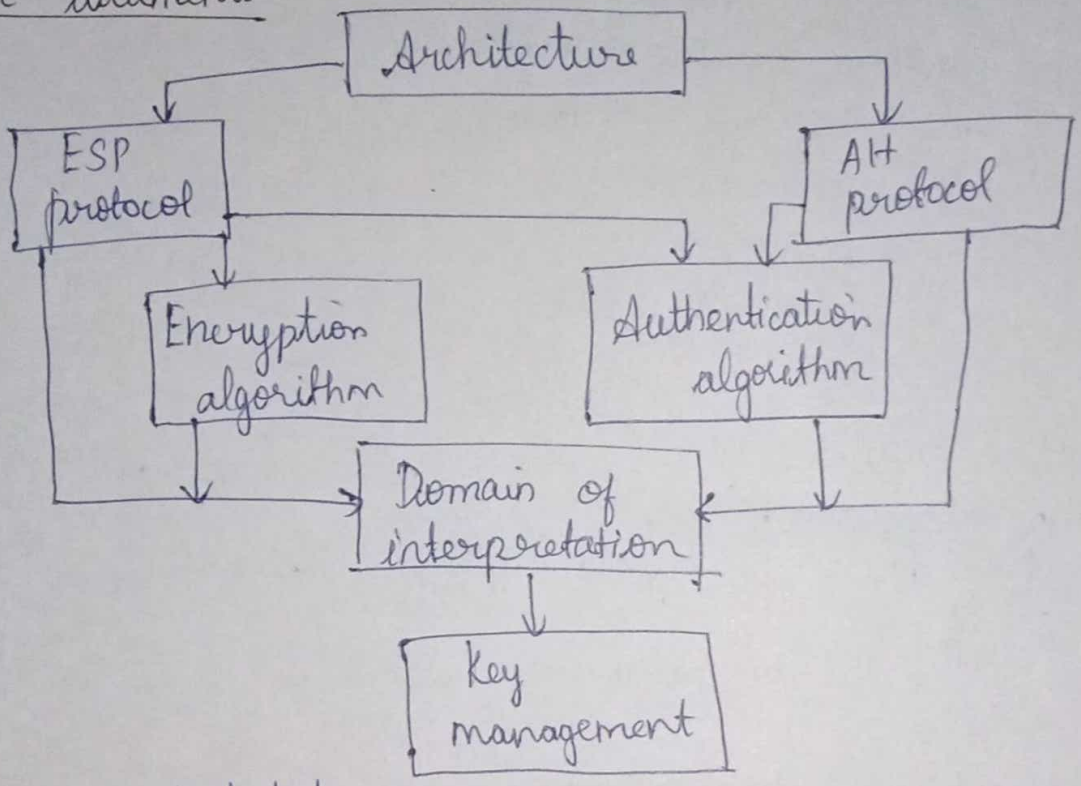
* firewall avoids bypass
* transparent
* change software

IP security architecture:

→ uses security policy database (SPD)

components: 1. IPSec documents
2. IPSec services
3. Security associations (SA)

IPSec documents:



→ Architecture
→ Encapsulating security payload
→ Authentication header
→ Authentication algorithm

IPSec services:

Access control
Connectionless integrity
Data origin authentication
Rejection of replayed packets
Confidentiality
Limited traffic flow confidentiality

IPSec protocol suit:

1. AH
2. ESP

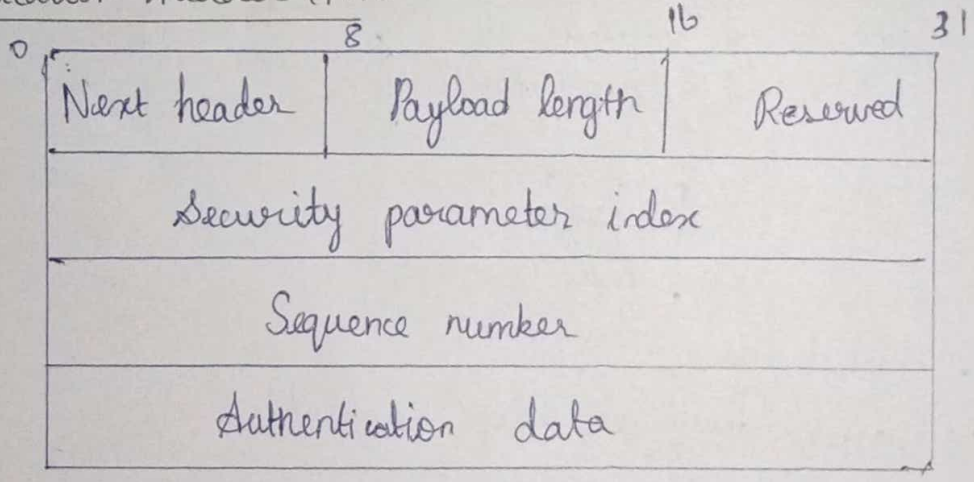
Security Associations (SA)

1. Security parameters Index
2. IP destination address
3. Security protocol identifiers

SA parameters:

→ Sequence number counter
→ Sequence counter overflow
→ Anti-replay window
→ AH information
→ ESP information
→ IPSec protocol mode
→ Path MTU

Authentication header (AH)



AH Transport mode → AH is between IP header & TCP header

AH Tunnel mode: IP packet is authenticated.

ESP

→ SPI
→ Sequence number
→ payload data
→ padding
→ padding length
→ Next header
→ Authentication data.

Combining security association:

1. Transport adjacency
2. Iterated tunneling

Key management of IPSec:

1. Manual
2. Automated

Internet key exchange protocol

Step 1 : IKE

Step 2: AH/ESP

ISAKHP header format:

→ Initiator cookie
→ Responder cookie
→ Next payload
→ Major version
→ Minor version
→ Exchange
→ Flags
→ Length.

Web security

Approaches:
→ Integrity
→ Confidentially
→ Denial of service
→ Authentication

TLS:
1. Handshake
2. Data exchange protocol

System Security: &
→ Masquerader
→ Misfeasor
→ Clandestine user

Intrusion Detection:

Functions:

- → Monitoring
- → Auditing
- → Assessing
- → Recognition
- → Statistical

Process model:

1. Information sources
2. Analysis
3. Response

Signature-based detection:

Adv:

* easy
* understand

Host based IDSs:

It logs for evidence of malicious or suspicious application activity in real time.

Techniques:

- → Threshold detection
- → Anomaly detection
- → Rule based detection

Password management:

To acquire protected information

Password protection

1. Password vulnerability
2. Encrypted passwords
3. One time passwords

Password selection strategies:

→ User education
→ computer generated passwords
→ Reactive password checking
→ Proactive password checking

Malicious software:

→ Trap Door
→ Logic bomb
→ Trojan horse
→ Virus

Types of Virus:

→ parasitic virus
→ Memory-resident virus
→ boot sector virus
→ stealth virus
→ polymorphic virus
→ Metamorphic virus

Worms:

→ Multiplatform
→ Multiexploit
→ polymophic

Virus countermeasures:

→ Detection
→ Identification of virus
→ Removal of traces of virus.

Firewall:

Types:

→ Packet filtering router
→ Application level gateways
→ Circuit level gateways

Firewall location:

1. DMZ network
2. Virtual private network
3. Distributed firewall.

Firewall configuration:

1. Screened host, single homed bastion host
2. Screened host, dual homed bastion host
3. Screened subnet.

1. Screened host, single homed bastion host:

* packet filtering router
* bastion host

2. Screened host, dual homed bastion host.

* prevents a security breach.

3. Screened subnet.

* isolated subnetwork