Unit V

Chapter 5

Cryptographic Algorithm

Syllabus

Mathematical preliminaries : Groups, Rings, Fields, Prime numbers, **Symmetric key algorithms :** Data encryption standards, Advanced encryption standard, **Public key encryption and Hash function :** RSA Digital signatures, **Digital certificates and Public key infrastructure :** Private key management, Diffie Hellman key exchange, The PKIX model.

Chapter Contents

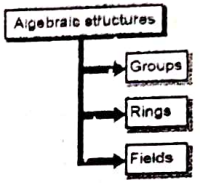
5.1 Algebraic Structures :

Definition :

- The combination of a set of integers and the operations that are defined on the elements of the set is called as an algebraic structure.

Common Algebraic Structures :

- Fig. 5.1.1 shows three commonly used algebraic structures.



(G-2968)**Fig. 5.1.1 : Common Algebraic Structures**

5.1.1 Groups :

Definition :

- A Group (G) is a set of elements along with a binary operation " • " performed on each ordered pair (x, y) of elements of G such that x • y satisfies four properties Closure, Associativity, Existence of identity and Existence of inverse.
- A group (G) is denoted as G = <(...), • >.

Properties of Groups :

- The following are the four properties of Groups :
  1. Closure
  2. Associativity
  3. Existence of identity
  4. Existence of inverse

1. Closure :

- If both a and b belong to the same group G, then a • b is also an element of G.
- That is, if a and b are the elements of the same group, then the result of applying the operation on these elements is another element of that group.
- For a, b ∈ G, a.b ∈ G

2. Associativity :

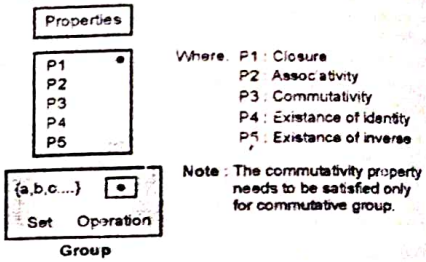
- If a, b and c belong to the same group G, then (a.b) • c = a • (b • c). That means, the order of operation does not affect the result.

3. Existence of identity :

- For all a in G , there always exists an identity element 'e' within the same group such that a • e = e • a = a.

4. Existence of inverse :

- For each a in G , there always exists an inverse element a' within the same group such that a • a' = a' • a = e.
- Fig. 5.1.2 shows the concept of a group.



(G-2969)**Fig. 5.1.2 : Concept of a group**

Definitions related to Groups :

Abelian group :

- A group that satisfies all the four properties of a group (Closure, Associativity, Existence of identity, Existence of inverse) and an additional property called commutativity is said to be an Abelian group, is also known as commutative group.
- The commutative property states that for all a and b belonging to G , a • b = b • a.

Finite group :

- A group that contains a finite number of elements is called as a finite group, otherwise it is an infinite group.
- For example, a group G_1 = < {1, 3, 5, 7, 9),+ > is a finite group while a group G_2 = < Z_n, + > where Z_n is a set of integers, is an infinite group.

Order of a Group :

- The number of elements in a group indicates the order of the group.
- For example, the order of group G_1 is finite while the order of group G_2 is infinite.

Subgroup :

- A subset of G is nonempty subset H of group G, which itself is a group with respect to the operation on G.

Cyclic Group :

- A group G is cyclic if every element of G is power a^k (k is an integer) of a fixed element aHG.

Cyclic Subgroups :

- The subgroup is the cyclic subgroup if a subgroup of a group can be generated using the power of an element.
- Eg. a^n = a·a·.....·a(n times)

Order of an Element :

- The order of an element 'a' in a group ord(a) is the smallest integer n such that $a^n = e$.

Application of Group :

- A group involves a single operation. The properties imposed on the operation allow the use of a pair of operations as long as they are inverses of each other.
- If the defined operation is addition, the group supports both addition and subtraction as subtraction is addition using the additive inverse.
- Similarly, this is also true for multiplication and division.
- A group can support only addition/subtraction or multiplication/division operations, but not the both at the same time.

5.1.2 Rings :

Definition :

- A ring is a set of elements with two binary operations, " · " and " □ ".
- A ring R is denoted by R = < {. . . }, ·, □ >.
- Note that □ can be an operator such that . (product) or + (addition) etc.
- The first operation should satisfy all properties required for an Abelian group.
- In other words, " · " should satisfy the properties closure, associativity, commutativity, existence of identity, and existence of inverse.
- The second operation "□" should satisfy two properties closure and associativity.
- In addition, the second operation (□) must be distributed over the first operation (·).

Distributivity :

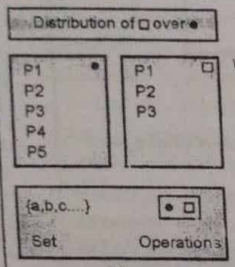
- If □ and · are two binary operators working on a ring R, then □ is said to be distributive over · if the following condition is satisfied.

$$a \square (b \cdot c) = (a \square b) \cdot (a \square c)$$

and $(a \cdot b) \square b = (a \square c) \cdot (b \square c)$.

Commutative ring :

- A ring is said to be a commutative ring if it satisfies the commutative property for the second operation (□).
- That is for all a and b belonging to the ring R, a □ b = b □ a.
- Fig. 5.1.3 shows a ring and a commutative ring.



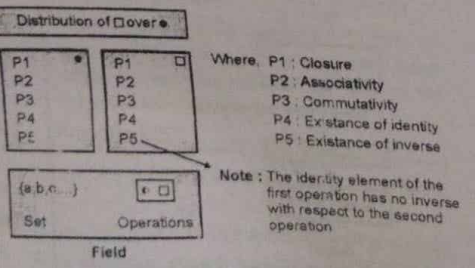
(G-2970) Fig. 5.1.3 : Concept of Ring

Application :

- A ring involves two operations. But, the second operation □ can fail to satisfy the third and fourth properties.
- In a ring, the first operation is a pair of operation such as addition and subtraction but the second operation is a single operation such as multiplication, but not division.

5.1.3 Fields :

- A field F is denoted by F = < {. . . }, ·, □ >.
- A field F is a set of elements with two binary operations, " · " and "□ ", such that F is a commutative ring where the second operation (□) satisfies all the five properties defined for the first operation (·) except that there is no inverse for the identity element of the first operation with respect to the second operation.
- Fig. 5.1.4 shows the field.



(G-2971) Fig. 5.1.4 : Concept of Field

Application of Field :

- A field supports two pairs of operations : addition/subtraction and multiplication /division.
- Division by zero is not allowed in a Field.

Finite fields :

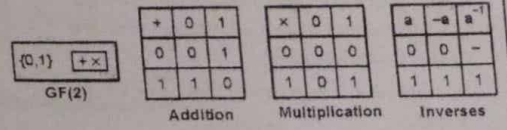
- A field with a finite number of elements is known as a finite field .
- Finite fields are used in cryptography for performing modular arithmetic operations.
- The concept and theory of finite fields was given by Galois.

Definition of GF(q) :

- A field with finite number of elements in it is called as a Galois field. It is pronounced as Galva Field and denoted by GF(q).
- According to Galois, if a field is finite, it contains p^n number of elements, where p is a prime number and n is a positive integer.
- Thus, the finite fields are usually known as **Galois field** and is denoted by $GF(p^n)$.
- A finite field with n = 1 is known as the GF(p) field.
- This field can be the set Z_p ={0, 1,.....,p−1} with two arithmetic operations addition and multiplication.
- Each element of this set has an additive and multiplicative inverse except zero, which has no multiplicative inverse.
- A very common field GF(2) will have only two elements 0 and 1 in it, A field GF(3) will have three elements {0, 1, 2} in it and GF(4) will have four elements {0, 1, 2, 3} in it and so on.

Addition and multiplication for GF (2) field :

- The addition and multiplication for the fields GF(2) is as shown in Fig. 5.1.5.



(G-2972)Fig. 5.1.5 : Addition and multiplication for a field GF(2)

- In this case, addition/subtraction is equivalent to the exclusive-OR (XOR) operation, and multiplication / division is equivalent to the logical AND operation.

$GF(2^n)$ Fields :

- We know that, the positive integers in the computers are stored in the form of n-bit words, where the value of n can be 8, 16, 32, and so on.
- This means , the range of integers that can be stored is 0 to 2^{n-1} and the modulus is 2^n .
- Using the GF(p) finite field with the set Z_p (where p is the largest prime number less than 2^{n-1}) would be inefficient as the range of integers from p to 2^{n-1} will not be used.
- To overcome this inefficiency of the GF(p) field, the $GF(2^n)$ field is used. This field uses a set of 2 n elements, and each element is an n -bit word.

5.1.4 Summary of Algebraic Structures :

- Table 5.1.1 gives the summary of algebraic structures.

Table 5.1.1 : Summary of Algebraic Structures

Sr. No.	Parameter	Group	Ring	Field
1.	Supported Operations	(+ −) or (× ÷)	(+ -) and (×)	(+ −) and (× ÷)
2.	Set of Integers	Z_n or Z_n^*	Z	Z_p

5.2 Prime Numbers :

Definition :

- A prime number is a positive integer greater than 1 which is divisible by only two integers, 1 and itself.
- For example, the numbers 2, 3, 5, 7, 11, 13, 17 and 19 are all prime numbers, whereas the numbers 4, 6, 8 and 10 are composite (means not prime), because they have more than two divisors.
- The smallest prime number is 2, which is divisible by 2 itself and 1.
- The integer 1 is not a prime because a prime must be divisible by two different integers, no more, no less and the integer 1 is divisible only by itself.

Relatively prime / Coprimes :

- Two positive integers a and b are said to be relatively prime or co-prime, if gcd(a, b)= 1.
- The gcd stands for Greatest common divisor.

- GCD of two positive integers is the largest integer that can fully divide both the integers.
- E.g. GCD (5, 10) is 5. GCD of (11, 13) is 1.
- Two numbers are said to be relatively prime if they have no common factors except the integer 1.
- For example, the integers 14 and 15 are relatively prime; however, the integers 14 and 16 are not relatively prime as they have a common factor other than the integer 1.
- The integer 1 is relatively prime with any integer.
- Also, if p is a prime number, all integers ranging from 1 to p−1 are relatively prime to p.

How to check Primeness of number :

- To check the primeness of number, apply the divisibility test, where we check whether the number n is divisible by any of the prime numbers less than $\sqrt{n}$.

Ex. 5.2.1 : Check whether 89 is a prime.

Soln. :

- The floor of $\sqrt{89} = 9$.
- The integral value of 89 is 9.
- The prime numbers less than 9 are {2, 3, 5, 7}. As 89 is not divisible by any of these numbers, it is a prime number.

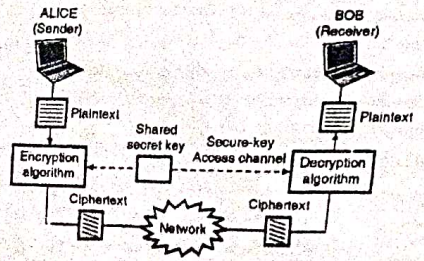
Ex. 5.2.2 : Check whether 308 is a prime.

Soln. :

- The floor of $\sqrt{308} = 17$
- The integral value of 308 is 17.
- The prime numbers less than 17 are {2, 3, 5, 7, 11,13,17}. As 308 is divisible by 7, it is not a prime number.

5.3 Symmetric Key Cryptography :

- It is also called as the secret key cryptography.
- Fig. 5.3.1 shows the block schematic of symmetric key cryptography.



(G-2339) Fig. 5.3.1 : Symmetric key cryptography

- In the symmetrical key cryptography, the same key (shared secret key) is used by the sender and receiver. The same secret key is used at the sending end as well as at the receiving end as shown in Fig. 5.3.1.
- The sender uses this key along with the encryption algorithm to encrypt the data, and the receiver uses it along with the decryption algorithm to decrypt the data.
- The encryption algorithm makes use of a combination of addition and multiplication to encrypt the data whereas the decryption algorithm uses a combination of subtraction and division in order to carry out the decryption.
- The encryption and decryption algorithms are thus inverse of each other.
- Thus **Encryption** is equivalent to locking the message in a box while **Decryption** is equivalent to unlocking the box.
- In the symmetric key ciphers, the same key **(Shared key)** is used for locking as well as unlocking.

5.3.1 Advantages of Symmetric Key Ciphers :

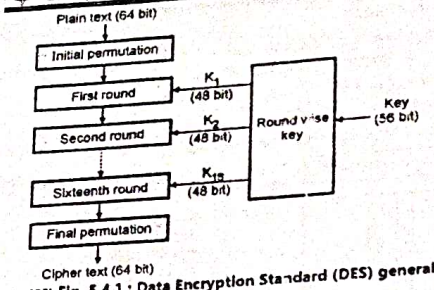
1. The major advantages of symmetric key algorithm is that it is more efficient than the public key algorithms. It takes less time to encrypt a message using the symmetric key algorithm. This is because this key is of smaller size (length).
2. The symmetric key algorithms can be used for encryption and decryption of long messages.

5.3.2 Disadvantages of Symmetric Key Ciphers :

1. The first disadvantages is that the sender and receiver both should have a unique symmetric key. So a large number of keys are required when the numbers of users increases.
2. The distribution of keys between two users can be difficult.

5.4 Data Encryption Standard (DES) :

- We will now discuss the **Data Encryption Standard (DES)** as an example of modern block cipher.
- The elements of DES cipher at the encryption have been shown in Fig. 5.4.1.



(G-680) Fig. 5.4.1 : Data Encryption Standard (DES) general structure

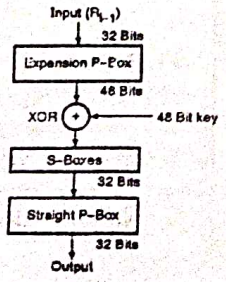
- The Data Encryption Standard (DES) is a block cipher that uses shared secret encryption. DES is based on a symmetric-key algorithm that uses a 56-bit key. The DES at the sending site takes in the 64-bit plaintext, works on it using a 56 bit symmetric key and produces a 64 bit ciphertext.
- At the receiving end, the DES takes the 64-bit ciphertext, works on it using the same 56 bit key and produces the 64 bit plaintext again.

Rounds :

- As shown in Fig. 5.4.1, there are 16 rounds in a DES, each one of which is an invertible transformation. Each round consists of upto two cipher elements namely the swapper and the mixer. Both of them are invertible.
- Each rounds take L_{i-1} and R_{i-1} (Left and Right parts of 32 bit each) from the previous round and produces its own L_i and R_i parts of 32 bits each. The output of the sixteenth round is applied to the **Final Permutation** block which produces the 64 bit ciphe text.

DES function :

- Fig. 5.4.2 shows the block diagram of DES function, which is the heart of DES.

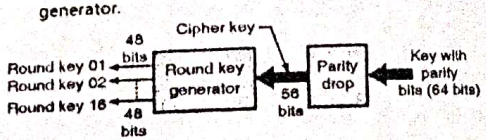


(G-2341) Fig. 5.4.2 : Block diagram of DES function

- As shown a DES function consists of four sections :
  1. An expansion P-box.
  2. A whitener to add key.
  3. S-boxes.
  4. Straight P-box.
- A DES function applies a 48-bit key to the rightmost 32 bits (R_{i-1}) and produces a 32 bit output.
- The input (R_{i-1}) is 32 bit long and the key has 48 bits. Therefore the input is expanded to 48 bits using the expansion P-box. After expansion, the XOR operation is used on the expanded right section and the 48 bit key.
- Real mixing takes place at the S-boxes. The straight permutation is carried out in the straight P-box to produce the final 32-bit output.

5.4.1 Key Generation :

- Fig. 5.4.3 shows block diagram of the round key generator.

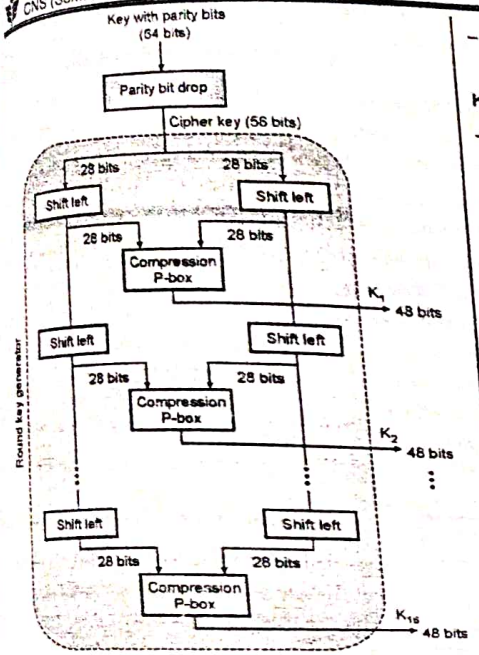


(G-2342) Fig. 5.4.3 : Key generation

- It generates sixteen 48 bit keys from one 56 bit cipher key.
- The cipher key is generally mentioned to be of 64 bit length.
- But out of these 64 bits, 8 bits are **parity bits** which do not contain any information.
- Therefore these bits are first dropped by means of the **parity drop** block.
- The remaining 56 bits are then used by the round key generator to produce sixteen 48 bit keys.
- The round key generator is actually a very complex combination of different units like shifting, splitting and combining units.

Detail steps in DES :

- Fig. 5.4.4 shows the key generation during each round.



(G-2963)**Fig. 5.4.4 : Key generation during each round**

- The round key generator uses the 56-bit cipher key and performs the following steps to generate the key for that round.
  1. The plaintext is divided into two halves of 28 bits each.
  2. The circular left shift operation is performed on each 28-bit half. Shift either 1 or 2 bits, depending on the round number. For the rounds 1, 2, 9 and 16, shifting is done by 1 bit, whereas in the case of the other rounds, shifting is done by 2 bits.
  3. After performing the shifting, both halves are combined again to form a 56-bit part. These 56 bits are applied to the compression P-box.
  4. The compression P-box, compresses the 56-bit input to produce 48-bit output. This 48-bit output generated from the P-box is used as a key for the round.

5.4.2 Strengths of DES :

- The strength of any cryptographic system is measured by the fact that how resistive it is to an attack.
- In case of DES, important aspects are : key size and the use of S-boxes.

Key size :

- DES uses 56-bit keys in each round, which means 2^{56} number of keys.
- Hence, a brute-force attack on DES is practically impossible.
- Assume that to get the correct key, only half of the total keys are needed to be examined, a single computer performing one DES encryption per microsecond will take more than 1000 years to break the DES.

Use of S-boxes :

- DES makes use of eight S-boxes (substitution tables) in each round.
- The internal design of these substitution tables has been kept secret by IBM.
- A suspicion think that there can be some weaknesses in the internal design of S -boxes that can be broken by cryptanalysts to break the DES security.
- Over the years, according to a number of studies there is a scope of attacking DES through S-boxes; however, no one has succeeded till date.

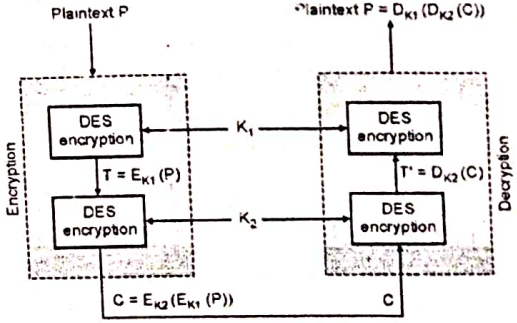
5.4.3 Weaknessess of DES :

- Following are the weaknesses of DES :
  1. Two specifically selected inputs to an S-box can create same output.
  2. The purpose of initial and final permutation is not clear.
  3. Weal keys create same 16 round keys.
  4. Semi weak keys create two different round keys.
  5. Possible weal keys create 4 distinct round keys.
  6. DES, each pair of semi-weak keys is a key cluster where two or more different keys result in the same ciphertext from the same plaintext.

5.5 Double DES :

- Double DES (2-DES) is the simple version of multiple-DES.
- Double DES performs DES encryption and decryption twice using two different keys (K1 and K2) of 56 bits each.

- The size of key is double to 112 bits, thus, increasing the cryptographic strength to double as compared to normal DES.
- Fig. 5.5.1 shows the encryption and decryption processes in double DES.



(G-2962)**Fig. 5.5.1 : Encryption and decryption in double DES**

Encryption :

- As shown in Fig. 5.5.1, the plaintext P is initially encrypted using DES with key K1 in order to obtain the temporary ciphertext T = E_{K1}(P).
- The temporary ciphertext T is again encrypted using DES with key K2 to obtain the final ciphertext C = E_{K2}(T), that is, C = E_{K2}(E_{K1}(P)).

Decryption :

- In the decryption process, the reverse process is followed to decrypt the ciphertext.
- In decryption, the keys are used in the reverse order of that of encryption.
- As shown in Fig. 5.5.1, the ciphertext C is first decrypted using DES with key K2 to obtain the temporary plaintext T' = D_{K2}(C).
- Then the temporary plaintext T' is again decrypted using DES with key K1 and the original plaintext is obtained, P = D_{K1}(T'), i.e. P = D_{K1}(D_{K2}(C)).

Meet-in-the-middle Attack :

- An attacker would need 2^{112} attempts to break the cipher key as the key size is 112 bits in the double DES.
- However, this is not true because of the meet-in-the-middle attack introduced by Merkle and Hellman.
- In this type of attack, encryption is performed from one end and decryption is performed from the other end and matching the result in the middle, hence the name Meet-in-the-middle Attack.
- The meet-in-the-middle attack is based on the observation.
- That means if we have C = E_{K2}(E_{K1}(P)) , then we can have E_{K1}(P) = D_{K2}(C), that is, T = T' .

How this attack happens ?

- Let us consider that the attacker knows a plaintext P and a ciphertext C of some message.
- The attacker may perform the following steps in order to determine the keys K1 and K2 :

1. For each of the 2^{56} possible values of K1, assign a large table in the memory and perform the following :
   1. Calculate the temporary ciphertext T = E_{K1}(P).
   2. Store the value of temporary ciphertext T in the next available row of the table in memory.
- After performing the above two steps, a table containing the values of the temporary ciphertext T is obtained.

2. For each of the 2^{56} possible values of K2, perform the following :
   1. Calculate the temporary plaintext T'= D_{K2}C).
   2. Compare the value of T' with all the values in the table of temporary ciphertext T.
   3. If the value of 'T' matches with any value of T in the table, use the corresponding pair of K1 and K2 to encrypt and decrypt another known pair of plaintext (P') and ciphertext (C') respectively.
   4. If E_{K1}(P')= D_{K2}(C'), then K1 and K2 are the exact keys and can be used for remaining blocks of the message.

- Although the meet-in-the-middle attack is possible on double-DES, it requires a lot of memory space to store the values of temporary ciphertext T.
- E.g. if a 64-bit plaintext block and a 56-bit key are used, then 2^{56} 64-bit blocks of memory would be required, which is very high.
- Hence, the meet-in-the-middle attack is practically impossible.

$\therefore$ T = (A – 1) (B – 1)

3. Choose the public key (E) which is a randomly chosen number such that it has no common factors with T.
4. Obtain the private key (D) as follows

$$D = E^{-1} \bmod T \quad ...(5.9.2)$$

5. The rule (algorithm) for encryption of a block of plaintext M into ciphertext (C) is as follows :

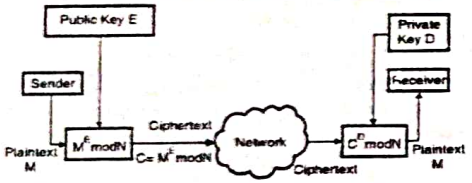
$$C = M^E \bmod N \quad ...(5.9.3)$$

- That means the plaintext M is raised to the power of E (public key) and then divided by N.
- The mod term in Equation (5.9.3) tells us that the remainder of this division is sent as the ciphertext C as shown in Fig. 5.9.3.

6. The received message C at the receiver is decrypted to obtain the plaintext back by using the following rule (algorithm).

$$M = C^D \bmod N \quad ...(5.9.4)$$

- The encryption and decryption process using the RSA algorithm is illustrated diagrammatically in Fig. 5.9.1.



(G-682) **Fig. 5.9.1 : Encryption and decryption in RSA**

Security of RSA :

- The security of RSA is decided by the ability of the hacker computer to factorize numbers. RSA provides a very good security because it uses very large prime numbers A and B their product is so large that an attempt to break the code using even the fastest computer will need a few years.
- But as the computers improve all the time, the time required to break the code will also reduce and one has to use larger keys. But then the time required for encryption and decryption also will increase.
- A key size of 768 bits is recommended for the personal use, 1024 bits for the corporate use and 2048 bits for extremely valuable keys.
- The user's key should be changed regularly in order to enhance security.

Ex. 5.9.1 : What is RSA algorithm ? If N = 119, public key E = 5, and private key D = 77 then demonstrate how to send the character F using RSA.

Soln. :

- Refer section 5.9.3 for RSA algorithm. The character F is the sixth character in alphabets. So we can represent it by 6.
- So as per RSA, the encryption is given by,

$$C = M^E \bmod N = 6^5 \bmod 119$$

- The process of getting C is as follows :

$$6^5 = 7776$$

$$\therefore C = 7776 \div 119$$

- Since the quotient of this division is 65 and remainder is 41 we take C = 41. This number is sent to the receiver as ciphertext.
- The receiver uses the decryption algorithm to get the plaintext M back as follows :

$$M = C^D \bmod N = 41^{77} \bmod 119$$

$$= 6 \quad \text{... which is the original number}$$

Ex. 5.9.2 : Select suitable values of A, B, N, D and E and demonstrate the encryption and decryption procedures used in RSA algorithm.

Soln. :

Step 1 : Select A and B :

- Let the two prime numbers A and B be as follows :

$$A = 11, \quad B = 23$$

Step 2 : Calculate N and T :

$$N = A \times B = 11 \times 23 = 253$$

$$T = (A - 1)(B - 1) = 10 \times 22 = 220$$

Step 3 : Calculate D and E :

- E (public key) should not have any factor other than 1 in common with T i.e. 220.

$$220 = 2 \times 2 \times 55 = 2 \times 2 \times 5 \times 11$$

So we can select E = 3

- Now calculate D (private key) with the help of following expression

$$D = E^{-1} \bmod T$$

$$\therefore D = 3^{-1} \bmod 220$$

- D is calculated as follows :
- Find the (multiple of 220 + 1) which is divisible by 3. Then divide that number by 3 and select the quotient of this division as D.

$\therefore$ (220 × 1) + 1 = 221 not divisible by 3

(220 × 2) + 1 = 441 it is divisible by 3

$\therefore \frac{441}{3} = 147$

$\therefore$ D = 147

Step 4 : Encryption :

- Let the letter I is to be sent.

$\therefore$ plaintext M = 9, as F is the ninth alphabet.

$\therefore$ Ciphertext C = $M^E \bmod N$

$\therefore C = 9^3 \bmod 221$

$\therefore C = 729 \bmod 221$

$$C = \frac{729}{221}$$

Q = 3 remainder = 66

$\therefore$ Ciphertext C = 66

- This number is sent to the receiver.

Step 5 : Decryption :

Plaintext M = $C^D \bmod N = 66^{147} \bmod 221 = 9$

- Thus the original number is obtained.

Ex. 5.9.3 : Generate the public key and secret key for the following prime numbers using RSA algorithm P = 3, Q = 11 take E = 5.

Soln. :

Given : The two prime numbers, P = 3, Q = 11, Public key E = 5

Step 1 : Calculate N and T :

$$N = P \times Q = 3 \times 11 = 33$$

$$T = (P - 1)(Q - 1) = 2 \times 10 = 20$$

Step 2 : Calculate D and E :

- E (public key) should not have any factor other than 1 in common with T i.e. 20.

$$T = 20 = 2 \times 2 \times 5$$

$\therefore$ Select E = 3 (public key)

- Now calculate D (secret key) with the help of following expression.

$$D = E^{-1} \bmod T$$

$$\therefore D = 3^{-1} \bmod 20$$

Find D as follows :

- Find the (Multiple of 20 + 1) which is divisible by 3. Then divide that number by 3 and select the quotient of this division as D.

$\therefore$ (20 × 1) + 1 = 21 is divisible by 3. So select it

$\therefore \frac{21}{3} = 7 \leftarrow$ Quotient

$\therefore$ D = 7 ...Ans.

$\therefore$ Public key E = 3

Private key D = 7

5.9.1 Comparison of DES, AES and RSA :

Table 5.9.1 : Comparison of DES, AES and RSA

Sr. No.	Parameter	DES	AES	RSA
1.	Algorithm type	Symmetric	Symmetric	Asymmetric
2.	Key length	56 bits	123,192,256 bits	More than 1024 bits
3.	Cipher block size	64	128	Min 512
4.	Scalable	Yes	No	No
5.	Power utilization	Less	Less	Very high
6.	Security	Moderate	High	Less
7.	Key for Encryption and Decryption	Same key	Same key	Different key
8.	Number of rounds	16	10 / 12/ 14	single
9.	Hardware and software implementation	Better hardware than software	Better in both	Not efficient

5.10 Hash Function :

Definition :

- The process of transforming input message M into a fixed size string (called as hash value h) is known as hash function.
- Here h is the output of hashing function applied on input message M.

$$h = H(M)$$

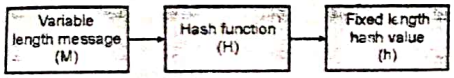
Where,

M = message (string) of any length

H = hash function

H(M) = a fixed-length string (hash code)

- Fig. 5.10.1 shows the generation of hash value.

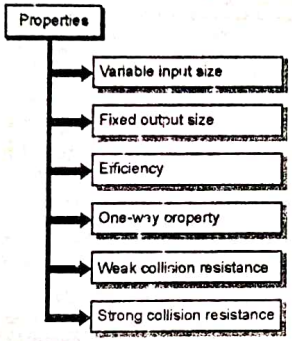


(G-2966)**Fig. 5.10.1 : Generation of hash value**

- A hash function or one-way hash function is a variation of MAC (Message Authentication Code). MAC is used for message authentication.
- Similar to MAC, hash function takes a variable-length message as input and generates a fixed-length output.
- The fixed length output is referred to as the hash code or hash value or a message digest.
- A hash function does not need a secret key and hence, is also known as a non-key message digest.
- At the source, the hash code is calculated and concatenated with the message.
- The information along with hash code is sent to the destination through the network.
- At the destination, the receiver separates the message from the hash code and again applies the hash function on it to create a new hash code.
- The message is authenticated, if the recalculated hash code is the same as the received hash code.
- Hash code plays the role of a 'signature' for the message being sent from the sender to the receiver as a secret key is not given as an input to hash function.
- The hash function considers all bits of the message, hence, hash code changes with change in any bit of the message.

5.10.1 Requirements for a Hash Function :

- The main purpose of a hash function is to create a "fingerprint" of a file, message or other block of data.
- Fig. 5.10.2 shows the properties or requirements of Hash function.



(G-2965)**Fig. 5.10.2 : Properties of Hash Function**

- The hash function must have the following properties in order to be used for message authentication.

1. **Variable input size :**
- Hash function H can be applied to a block of data of any size.
2. **Fixed Output size :**
- The output generated by the hash function should always be of fixed length.
3. **Efficiency :**
- It is easier to generate the hash code for any given message or block of data.
- That means for a given message, H(M) should be easily computable making the hardware and software implementation, feasible.
4. **One-way property :**
- It is nearly impossible to determine the corresponding message or block of data for any given hash value .
- That means, one should not be able to determine M if h is given, such that H(M)= h . This is referred to as one-way property .
- This property is mainly important when a secret value is being used in the authentication. Even though the secret value is not sent, the attacker can still easily find out the secret value if the used hash function does not show the one-way property.

5. **Weak collision resistance :**
- It should not be computationally possible to determine another message or block of data generating the same hash code as that of the given message or block of data.
- That means, for given M_1, there is no other M_2 (where $M_1 \neq M_2$) such that $H(M_1) = H(M_2)$. This property is referred to as weak collision resistance .
6. **Strong collision resistance :**
- There should not be same hash code for two messages or blocks of data, even being almost similar.
- That means, it is almost impossible to determine a pair (M_1, M_2) such that $H(M_1) = H(M_2)$. This property is referred to as strong collision resistance .
- Out of these six properties, if the first five properties are satisfied, then the hash function is called a **weak hash function**.
- If all the six properties are satisfied, then the hash function is called a **strong hash function**.
- Because the sixth property protects the hash function from the birthday attack.

5.10.2 Simple Hash Function :

- The hash functions considers the input message as a sequence of blocks. Where each block is of m bits.
- The input message is processed one block at a time iteratively and generate an m -bit hash code.
- One of the simplest hash functions takes the bitwise XOR of every block of the input message to generate the hash code.
- This can be expressed as follows :

$$h_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \ldots \oplus b_{in}$$

- Where,

h_i = i^{th} bit of the hash code with $1 \leq i \leq m$

n = number of m -bit blocks in the input message

b_{ik} = i^{th} bit of the k^{th} block with $1 \leq k \leq n$

- This operation generates a simple parity corresponding to each bit position and is known as Longitudinal Redundancy Check (LRC).
- It effectively guarantees data integrity for randomly selected input whereas it is less effective in case of predictable formatted data.
- An simple way is used to improve the effectiveness of the hash function. It rotates (circular-shifts) the hash value by one bit after processing each block.
- This hash function follows the following steps to generate an m-bit hash code from an input message consisting of m-bit blocks.

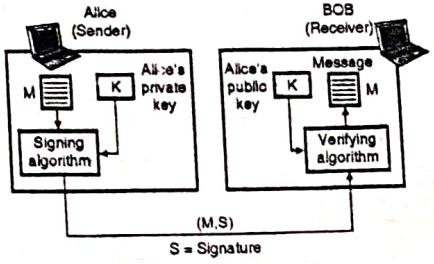
1. Set all the m bits of hash value to zeros.
2. For each successive m-bit block, perform the following:
   (a) Rotate (Shift left) the current hash value by one bit.
   (b) Take the XOR of new hash value and the block.

5.11 Digital Signature :

- We can use the **digital signature** to provide the message integrity, message authentication and some more security services.
- The digital signature makes use of a pair of public and private keys to protect the digest.
- **Digital signature** is the electronic signature done by the sender (Alice) on the document being sent to confirm its authenticity.
- The receiver (Bob) will verify the digital signature to confirm that the document received by him is authentic.
- The digital signature can provide the nonrepudiation.

5.11.1 Digital Signature Process :

- The digital signature process is as shown in Fig. 5.11.1.



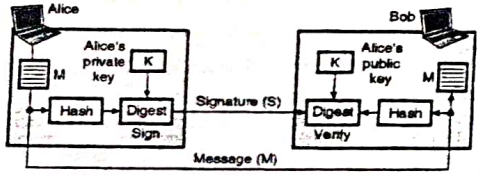
(G-2348) **Fig. 5.11.1 : Digital signature process**

- A signing algorithm is used by Alice to sign the message (M). And the message is sent along with the signature (S) on the unsecure channel. At the receiver, the message and signature (M, S) are applied to a verifying algorithm.

- If the result is true the message is accepted, otherwise it is rejected.
- Alice uses her **private key** (which is known only to her) applied to the signing algorithm to sign the document.
- Bob applies the **public key** (which is available to everyone) to the verifying algorithm to confirm the digital signature.
- Thus anybody can verify the digital signature but only Alice can sign it using her own private key.
- It is important to note that we cannot use **a secret key** to sign and verify a signature.

5.11.2 Signing the Digest :

- The asymmetric key cryptosystems are very inefficient in dealing with long messages and we need to use long messages in the digital signature systems.
- And we have to use the asymmetric key cryptosystem to realize the digital signature scheme.
- In order to solve this problem, the **digest** of the message (which is short) is signed instead of the message itself.
- The digest is selected very carefully for the same. This is illustrated in Fig. 5.11.2.



(G-2349) Fig. 5.11.2 : Digital signature with the concept of signing the digest

- At the sending end, the hash function produce the digest from the message M. Alice's private key is then used to sign the digest.
- The message (M) and the signature (S) are then sent to Bob.
- At the receiving end, the received message (M) is applied to the same hash function to create the digest.
- After applying the verifying process to this digest, to check the authenticity of the message.
- The message is kept if found authentic and discarded otherwise.

5.11.3 Services Provided by Digital Signature :

- The digital signature system can directly provide the following three services :
  1. Message Integrity
  2. Message authentication
  3. Non repudiation
- However the message confidentiality is not provided directly by digital signature. For that the encryption and decryption has to be done.

1. Message Authentication :

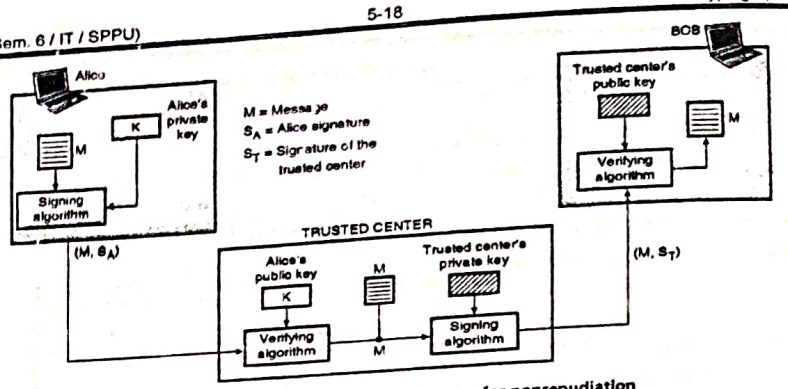
- A digital signature provides message authentication in exactly the same way as provided by the conventional signature.
- This is also called as the data origin authentication.
- That means Bob can verify the received message is indeed originated by Alice because he uses Alice's public key for such verification.

2. Message Integrity :

- By signing either the message itself or the message digest, we can preserve the message integrity.
- This is because, if an unauthorized entity changes a part of the message, then we won't get the same digest. Thus a digital signature scheme preserves the integrity of a message.

3. Nonrepudiation :

- Can a digital signature be used to ensure nonrepudiation ? That means in the event of Alice signing a message and latter denying it, can Bob prove that Alice actually signed the message ?
- The answer is yes. This can be done by creating a **trusted third party.** Fig. 5.11.3 explains this principle.
- As shown, Alice creates her signature (SA) from her message and then sends the message along with her own identity, Bob's identity and the signature (S_A) to the **trusted center.**



(G-2350) Fig. 5.11.3 : Use of a trusted center for nonrepudiation

- The center first checks the validity of Alice's public key and then using it verifies that the message has come from Alice.
- Then the centers keeps a copy of this message with Alice's identity, Bob's identity and a timestamp with itself. Using this message, the trusted center produces its own signature (S_T) with the help of its own private key.
- The center then sends this message, Alice's identity, Bob's identity and the new signature (S_T) to Bob, who uses the trusted center's public key to verify this message.
- In the event of Alice denying that sent the message, the trusted center can show the saved copy of Alice's message at the center. We can further add encryption and decryption to this scheme to make everything confidential.
- This can be achieved by encrypting the message and the signature using either the symmetric key or asymmetric key cryptosystem.

Digital signature standard :

- DSS is a very complex but more secure digital signature scheme.

5.11.4 Entity Authentication :

Definition :

- We may define the entity authentication as the technique using which one party in communication can verify the identity of the other party.

Entity :

- An entity could be anything, a person, a process, a client or a server.

Claimant :

- A claimant is defined as the **entity** whose identity is to be proven.

Varifier :

- Varifier is the party which tries to verify identity of a claimant.

5.11.5 Difference between Entity and Message Authentication :

- Following are the two differences between the entity authentication and message authentication.

1. The message authentication or data origin authentication might not take place in real time but the entity authentication does take place in real time.
2. The process of message authentication will authenticate only one message. So it is necessary to repeat it for every new message. But the entity authentication will authenticate the claimant for the entire session.

Verification categories :

- The claimant needs to identify himself to the verifier with one of the three witnesses as below :
  1. Something known.
  2. Something possessed.
  3. Something inherent.
- We will consider only the first one i.e. something known in this section. This type of witness is used for the online entity authentication.

Passwords :

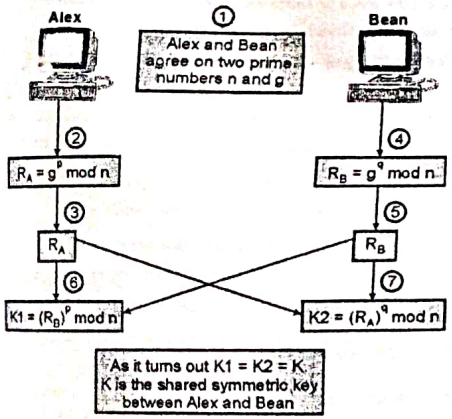
- Use of passwords is the simplest entity authentication method. But passwords are very prone to attacks.

5.12 Diffie-Hellman key Exchange :

- The Diffie-Hellman key exchange or Agreement Algorithm is the first public-key algorithm.
- It was published by Whitefield Diffie and Martin Hellman in 1976.
- This algorithm was developed to exchange the secret keys between the communicating users in a secure way.
- It allows two users to exchange the key which can be further used for encryption of messages.
- This algorithm can be used only for the exchange of keys not for encryption and decryption.
- This algorithm allows two users to set up a symmetric session (secret) key without the use of KDC. This is known as the symmetric-key agreement.
- Once both the parties agreed (exchanged) on the secret key to be used, then the other symmetric-key encryption algorithm can be used for encryption and decryption of messages.

5.12.1 Diffie-Hellman Algorithm :

- *Consider* two users Alex and Bean want to communicate with each other securely over an insecure network.
- Initially, both the users Alex and Bean need to agree upon a key that is to be used for encryption and decryption of the messages that would be exchanged between them.
- For this, they can follow the Diffie-Hellman key exchange algorithm as shown in Fig. 5.12.1.



(G-2973) **Fig. 5.12.1 : Diffie-Hellman key exchange algorithm**

- The Diffie-Hellman key exchange algorithm works as follows :

1. Select two large prime numbers n and g by the mutual agreement of Alex and Bean. There is no need to keep these two numbers secret. The values of n and g are public.
2. Alex selects a random number p and calculates the value of R_A such that $R_A = g^p \bmod n$.
3. Alex sends the value of R_A to Bean.
4. Bean selects a random number q and calculates the value of R_B such that $R_B = g^q \bmod n$.
5. Bean sends the value of R_B to Alex.
6. Now, Alex calculates the secret key K1 as follows :

$$K1 = (R_B)^p \bmod n$$

7. Bean calculates the secret key K2 as follows :

$$K2 = (R_A)^q \bmod n$$

- This means that K1 = K2 = K is the symmetric key, which Alex and Bean must keep secret and can use for encrypting/decrypting their messages.

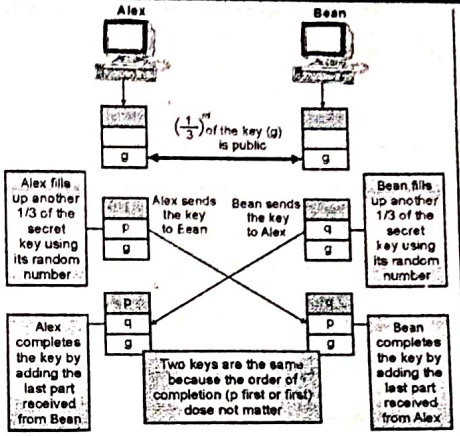
Proof of algorithm :

- To show that both Alex and Bean have calculated the same secret key, we need to prove that the calculation of K by Alex and Bean which creates the identical results.

$$K1 = (R_B)^p \bmod n$$
$$= (g^q \bmod n)^p \bmod n$$
$$= (g^q)^p \bmod n$$
...[using the rule of modular arithmetic]
$$= g^{pq} \bmod n$$
$$= (g^p \bmod n)^q \bmod n$$
... [using the rule of modular arithmetic]
$$= (R_A)^q \bmod n$$ **Hence, proved.**

5.12.2 Concept of Diffie-Hellman :

- The idea behind Diffie-Hellman is quite simple. Fig. 5.12.2 shows the idea behind Diffie-Hellman.



(G-2974) **Fig. 5.12.2 : Diffie-Hellman Idea**

- Think about the final shared secret key between Alex and Bean made up of three parts : g, p and q.
- The first part of the key (g) is a public and known to everyone.
- The other two parts of the key p and q should be made available by Alex and Bean.
- Alex adds the second part (p), while Bean adds the third (q).
- When Alex receives the two-third completed key from Bean, he/she adds remaining one-third part (i.e. p) in it. This completes Alex's key.
- Similarly, when Bean receives the two-third completed key from Alex, he/she adds remaining one-third part (i.e. q) in it. This completes Bean's key.
- Even though sequence of Alex's key is g-q-p and Bean's key is is g-p-q, the two keys are the same because gpq = gqp.
- Even though the final two keys are the same, Alex cannot find Bean's part (i.e. q) and Bean cannot find Alex's part (i.e. p) because the calculation is done using modulus n.

Example of the Algorithm :

1. Alex and Bean agree on two large prime numbers say n = 17 and g = 7.
2. Alex selects another secret large random number 5 i.e. p = 5 and calculate R_A such that

$$R_A = g^p \bmod n.$$
$$R_A = 7^5 \bmod 17 = 11$$

3. Alex sends the number R_A to Bean.
4. Bean selects another secret large random number 3. i.e. q = 3 and calculate s such that

$$R_B = g^q \bmod n.$$
$$R_B = 7^3 \bmod 17 = 3$$

5. Bean sends number R_B to Alex.
6. Now, Alex calculates the secret key K1 as follows :

$$K1 = (R_B)^p \bmod n$$
$$K1 = 3^5 \bmod 17 = 5$$

7. Bean calculates the secret key K2 as follows :

$$K2 = (R_A)^q \bmod n$$
$$K2 = 11^3 \bmod 17 = 5$$

8. If K1 = K2 then Alex and Bean can agree for future communication.
9. Here, K1 = K2 = K = 5. Hence proved.

- The shared key, that can be used between Alex and Bean, is K = 5.

5.12.3 Security of Diffie-Hellman Algorithm :

- In the Diffie-Hellman algorithm, the private keys p and q are secret, the numbers n and g and the public keys, R_A and R_B are known to everyone. Thus, an rival has n, g, R_A and R_B to work with.
- To find out the key using the available information, the rival has to use the discrete logarithm.
- For example, If the rival wants to find the private key of user Alex, then he or she has to perform the following calculation:

$$p = d \log_{g,n}(R_A)$$

- After computing p, the rival can calculate the common secret key (K) in the same way that Alex calculated it.
- Since it is difficult to compute the discrete logarithm in comparison to computing exponentials modulo a prime number, the security of the Diffie-Hellman algorithm depends on this fact.
- In case of large prime numbers, it is not possible calculate the discrete logarithm which breaks the security of the Diffie-Hellman algorithm.

5.12.4 Advantages of Diffie-Hellman Algorithm :

- The advantages of the Diffie-Hellman key exchange algorithm are as follows :
  1. The secret keys are generated when required. Therefore, they need not be stored for a long time. This makes them less vulnerable to attacks.
  2. There is no need of pre-existing infrastructure for key exchange. The communicating parties just have to agree upon the values of global variables n and g .

5.12.5 Disadvantages of Diffie-Hellman Algorithm :

- The disadvantages of the Diffie-Hellman key exchange algorithm are as follows :
  1. It does not authenticate the communicating users as it does not provide any information regarding the identities of the users exchanging the key.
  2. It is vulnerable to man-in-the-middle-attack.
  3. It involves a lot of calculations which results into clogging attacks. In the clogging attack, an rival requests for a large number of keys which makes the victim busy in doing unnecessary calculations rather than doing the real work.

Ex. 5.12.1 : Alice chooses her private key x = 3 and Bob chooses y = 6. If both of them use the primitive root g = 7 for prime p = 23, what is the key exchanged between Alice and Bob using Diffie Hellman key exchange ?

Soln. :

Given : g =7, p =23, Alice's private key (x) = 3,

Bob's private key (y) = 6.

To Find : Key exchanged between Alice and Bob

1. Public key of Alice and Bob :

$$\therefore A = g^x \bmod p = 7^3 \bmod 23 = 21$$

$$\therefore B = g^y \bmod p = 7^6 \bmod 23 = 4$$

2. Common secret Key :

- Both the users compute the shared key, K, at their respective ends.
- Alice calculates it as

$$K = B^x \bmod p = 4^3 \bmod 23$$

$$\therefore K = 18$$

- Bob calculates it as

$$K = A^y \bmod p = 21^6 \bmod 23$$

$$\therefore K = 18$$

- Hence, the shared key between Alice and Bob is 18.

Ex. 5.12.2 : Consider a Diffie-Hellman scheme with a common prime q = 11 and a primitive root α = 2.

1. Show that 2 is primitive root of 11.
2. If user A has public key Y_A = 9, what is A's private key X_A ?
3. If user B has public key Y_B = 3, what is the shared secret key K?

Soln. :

Given : g = 2, q = 11, Y_A = 9, Y_B = 3, Primitive root α = 2

To Find : 1. A's private key X_A 2. Shared secret key K

Show that 2 is primitive root of 11 :

- The powers of 2 mod 11 should generate all the integers from 1 to (11–1) which means it should produce (1, 2, 3, 4, 5, 6, 7, 8, 9, 10).

Table P. 5.12.2 : Primitive root table

Integer	Mod operation	Resultant Integer
2^0	1 mod 11	1
2^1	2 mod 11	2
2^2	4 mod 11	4
2^3	8 mod 11	8
2^4	16 mod 11	5
2^5	32 mod 11	10
2^6	64 mod 11	9
2^7	128 mod 11	7
2^8	256 mod 11	3
2^9	512 mod 11	6

- Above table shows that 2 is a primitive root of 11 as it generates all the numbers from 1 to 10 as expected.

A's private key X_A :

- We know that, $Y_A = g^{X_A} \bmod q$

$$9 = 2^{X_A} \bmod 11$$

- From the primitive root Table P. 5.12.2 , $2^6 \bmod 11$ gives 9.

$$\therefore X_A = 6 \quad \text{...Ans.}$$

$\therefore$ A's private key X_A = 6

B's private key X_B :

We know that, $Y_B = g^{X_B} \bmod q$

$$3 = 2^{X_B} \bmod 11$$

- From the primitive root Table P. 5.12.2, $2^8 \bmod 11$ gives 3.

$$\therefore X_B = 8$$

$\therefore$ B's private key X_B = 8

Share secret key K :

- Both the users compute the shared key, K, at their respective ends.
- A generates the secret key as,

$$K = (Y_B)^{X_A} \bmod q = 3^6 \bmod 11$$

$$\therefore K = 3$$

- B generates the secret key as,

$$K = (Y_A)^{X_B} \bmod q = 9^8 \bmod 11$$

$$K = 3$$

$\therefore$ Shared secret key K = 3 ...Ans.

5.13 Key Management :

- In this section we will discuss how the secret key in symmetric key cryptosystem and the public key in asymmetric key cryptosystem are distributed and maintained.

5.13.1 Symmetric Key Distribution :

Need of key management :

- In the symmetric key cryptography it is necessary to have a shared secret key between the two communicating parties
- If Alice wants to communicate confidentially only with Bob then only one key is required. But if she wants a confidential communication with N persons, then, N different keys are required to be used.
- Similarly if these N people want to communicate with each other, then the number of required keys would be N (N – 1). This is when two people communicate with each other using two separate keys one for each direction in a bidirectional communication.
- The number of keys will be reduced by half to N (N – 1)/2 if two communicating persons use the same key for communication in either direction.
- Still, this number N (N – 1)/2 is a huge number. If 1000 people are involved in communication then the number of keys required will be.

$$\text{Number of keys} = \frac{N(N-1)}{2} = \frac{1000 \times 999}{2}$$

$$= 499500$$

- But number of keys is not the only problem. The distribution of such a large number of key is the another one.
- The communicating people must have a way to exchange the secret keys with each other, and use of Internet is not a secure way. Therefore we must have some other efficient way of distributing and maintaining the secret keys. One way of doing it is to use a **Key Distribution Center (KDC).**

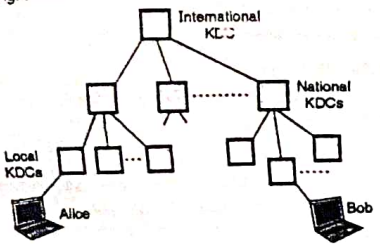
5.13.2 Key Distribution Center (KDC) :

- KDC is a trusted third party used for secret key distribution. The procedure of key distribution is as under.
- **Each person establishes a shared secret key** with the **KDC and a secret key** is established between the KDC and each of its members.
- **Now using the KDC how a confidential communication would take place between Alice and Bob ? The stepwise process is as follows :**
  1. Alice sends a request to KDC and tells that she wants a temporary secret key for the communication between herself and Bob.
  2. KDC tells Bob about Alice's request.
  3. If Bob is ready to communicate, then KDC creates a temporary (session) secret key between the two.
- This secret key will authenticate both Alice and Bob to KDC and prevents any attacker from impersonating Alice or Bob.

5.13.3 Multiple KDCs :

- Multiple KDCs are required with increase in the number of people that use KDC.

- Fig. 5.13.1 illustrates the principle of multiple KDCs in which the world is divided into multiple domains.
- For improvement in reliability, each domain will have multiple KDCs (for redundancy).
- If Alice and Bob are in different domains and want to communicate with each other then Alice will contact her KDC, it contacts Bob's KDC.
- These two KDCs create a secrete key between Alice and Bob.
- In the multiple KDC environment, there are international KDCs, national KDCs as well as local KDCs as shown in Fig. 5.13.1.



(G-2351) **Fig. 5.13.1 : Multiple KDC environment**

- In the multiple KDC environment, if Alice wants to communicate with Bob who is in some other country then the stepwise procedure is as follows :
  1. Alice sends request to local KDC.
  2. The local KDC relays the request to the national KDC.
  3. The national KDC relays it to the international KDC.
  4. In a similar way the request is relayed all the way to Bob.
- A secret key for each member is prepared by KDC which can be only used between a member and KDC but not between two members.
- The KDC create a session key, using the keys of the two members when they want to communicate with each other.
- This sessions key is useful only during that particular session of communication. Once that session is over, the corresponding session key is no longer useful.

5.13.4 Public Key Distribution :

- Public keys should be available to everybody. Therefore public keys need to distributed.
- Different ways of distributing the public keys are as follows :
  1. Public announcement.
  2. Certification authority.
  3. X.509.

5.14 Digital Certificates :

Definition :

- A digital certificate is a small computer file used to establish a relation between both the user and his or her public key.
- It is an electronic document that provides information to prove the identity of an entity.
- A digital certificate contains the user name and his or her public key. So that we can identify that the particular key belongs to the particular user.
- This concept of digital certificates is shown in Fig. 5.14.1.

Digital Certificate	
Subject name	: ABC
Public key	: key of ABC
Serial number	: 1029101
Other data	: Email, phone
Valid from	: 1 Jan 2008
Valid to	: 31 Dec 2016
Issuer name	: VeriSign

(G-2975) **Fig. 5.14.1 : Conceptual view of a digital certificate**

- A digital certificate contains the following information :
  1. Subject name
  2. Public key
  3. Serial number
  4. Other data like email, phone, etc
  5. Valid from
  6. Valid to
  7. Issuer name
- Here subject name is the name of the user. The public key is the photograph and signature.
- The serial number is the number of a digital certificate.
- Other data contain some personal information of user.
- Val dity field shows the validity of a particular digital certificate.

- The issuer's name is the name of the person who issues a digital certificate for a user.

Similarities between a passport and a digital certificate :

- As the Fig. 5.14.2 shows, the digital certificate is somewhat similar to a passport.

Passport entry	Digital certificate entry
Full name	Subject name
Passport number	Serial number
Valid from	Same
Valid to	Same
Issued by	Issuer name
Photograph & signature	Public key

(G-2976) **Fig. 5.14.2 : Similarities between a passport and a digital certificate**

- Every passport has a unique passport number, every digital certificate has a unique serial number.
- We know that, no two passports issued by the same issuer (i.e. government) can have the same passport number.
- Similarly, no two digital certificates issued by the same issuer can have the same serial number.

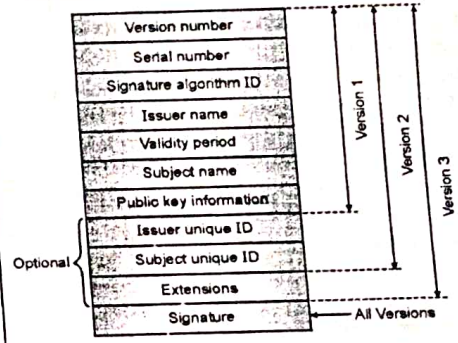
5.14.1 Certification Authority (CA) :

Definition :

- A certification authority is a trusted agency that issues digital certificates.
- A certificate authority (CA) should be the trusted one. The governments in the various countries decide who can and who cannot be a CA.
- Generally, a CA is a reputed organization like a post office, financial institution, software company, etc.
- A certification authority issues digital certificates to individuals and organizations.
- They can use those certificates in asymmetric key cryptographic applications.
- The VeriSign and Entrust are world's most famous CAs.

5.14.2 Structure of Digital Certificate :

- A standard X.509 defines the structure of a digital certificate.
- The International Telecommunication Union (ITU) designed a standard X.509 in 1988 to illustrate the public-key certificates in a structured way.
- Since then, the standard X.509 was revised twice in 1993 and again in 1995.
- The standard X.509V3 is the current version of the standard is Version 3, called as.
- Fig. 5.14.3 shows the structure of a X.509V3 digital certificate.



(G-2977) **Fig. 5.14.3 : Structure of a X.509 Digital Certificate**

- The Fig. 5.14.3 shows the various fields of a digital certificate according to the X.509 standard. The description of these fields is as follows:

Version number :

- This field identifies a particular version of the X.509 protocol. Currently this field can contain 1, 2 or 3.

Serial number :

- This field contains a unique integer number generated by the CA.

Signature algorithm ID :

- This field identifies the algorithm used by the CA to sign this certificate.

Issuer name :

- This field identifies the name of the CA that has created and signed the certificate.
- The issuer name is usually specified as a hierarchy of strings which include the name of the country, state, organization, department and so on.

Validity period :

- This field identifies the lifetime of the certificate. It includes the earliest time and latest time;.
- The certificate is invalid before the earliest time and after the latest time.
- These values generally specify the date and time up to seconds or milliseconds.

Subject name :

- This field specifies the entity to which this certificate refers.
- This certificate certifies the public key of the user who is holding the corresponding private key.
- It also specifies a hierarchy of strings; one of the strings is known as the common name, which is the actual name of the user.

Subject public key :

- This field contains the main content of the certificate, which is the public key.
- It also specifies the corresponding public-key algorithm and its associated parameters. This field can never be blank.

Issuer unique identifier :

- This is an optional field that identifies a CA uniquely if two or more CAs have used the same *Issuer Name* over time.

Subject unique identifier :

- This field helps to identify a subject uniquely if two or more subjects have used the same Subject Name over time.

Extensions :

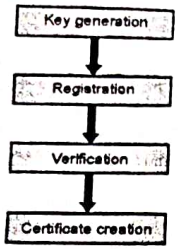
- The issuers uses this field to add more private information to the certificate.
- The extensions were added in the third version of X.509. This is also an optional field.

Signature :

- This field is divided into the following three sections :
- The first section contains all the other fields in the certificate.
- The second section includes the digest of the first section, which is encrypted with the private key of CA.
- The third section contains the algorithm identifier that has been used to generate the second section.

5.14.3 Creation of Digital Certificate :

- The steps in the creation of a digital certificate are summarized in Fig. 5.14.4.



(G-2979)Fig. 5.14.4 : Creation of Digital Certificate

- Following are the steps involved in the creation of digital certificate.
  1. Key generation
  2. Registration
  3. Verification
  4. Certificate creation

Step 1 : Key Generation :

- In this step, the key is generated by either user or registration authority. A key can be generated using two approaches.

Approach 1 :

- In this approach, the user creates a private and public key. Users must keep their private key secret. A user sends the public key with additional information to the registration authority (RA). RA is an intermediate between the CA and the user.

Approach 2 :

- RA generates a key pair for a user. This approach is required when the user is unaware of the technical and software knowledge. The disadvantage of this approach is RA can find out the private key of the user.
- The disadvantages of this approach are RA can find out the private key of the user and the scope for this key to be exposed to others while in transit after it is generated and sent to the appropriate user.

Step 2 : Registration :

- If the user generates the pair of key, then only this step is required.

- If RA generates the pair of a key for the user, then key registration is done in the first step only.
- A user sends his/her public key and some other information to the RA. This software provides a wizard where users enter details and submit.
- After that the data travel through the network and reach to the RA. After that user request for the certificate will be registered, and the format for the certificate request has been standardized. This process is known as CSR (Certificate Signing Request).

Step 3 : Verification :

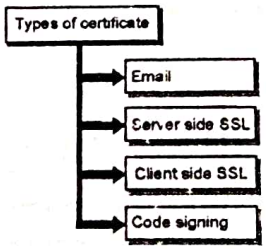
- After the completion of registration process, RA identifies the user credentials.
- It verifies the user's credentials such as the evidences provided are correct and that they are acceptable.
- It also ensures that the user who is requesting for the certificate does really possess the private key corresponding to the public key that is sent as a part of the certificate request to the RA.

Step 4 : Certificate Creation :

- After completion of all the above steps, RA passes all the details to the CA.
- CA cross-checks all the details and generates a digital certificate for the user.
- CA sends the certificate to the user and keeps one copy of certificate with itself to keep the records.
- The Copy of the certificate is stored in the certificate directory.

5.14.4 Types of Digital Certificates :

- Fig. 5.14.5 shows the classification of the certificate types :



(G-2980)Fig. 5.14.5 : Classification of the certificate types

- The Digital Certificates are classifies into :
  1. Email certificate
  2. Server-side SSL certificate
  3. Client-side SSL certificate
  4. Code Signing certificate

Email certificate :

- It includes the email id of the user. It verifies that the signer of an email message has an email id that is the same as mentioned in the user's certificate.

Server-side SSL(Secure Socket Layer) certificate :

- These types of a certificate are useful for the merchants because merchants want their users to trust on their site and buy goods or services form their web site.

Client-side SSL certificate :

- This type of certificates are used by a merchant to identify their clients.

Code Signing Certificates :

- The Code Signing certificates allows the Software developer to encrypt the code of their software or application.
- After encrypting the code attacker cannot change or modify that code. This type of certificates ensures the highest levels of security and verification.
- CA of the Code Signing Certificate verifies the reliability of software and the publisher's identity using public key infrastructure (PKI) and digital signature technology and confirms that your code has not been tampered with or corrupted.

5.14.5 Advantages of Digital Certificates :

- The advantages of digital certificates are as follows :

1. **Authentication :**
   - The identity of the entity can be verified by using digital certificates.
2. **Secure :**
   - It guarantees the public key belongs to the owner of the certificate. So a secure communication can established for confidential transactions like email, e-commerce and online transactions.
3. **Integrity :**
   - Integrity is guaranteed a long as the CA's signature on the digital certificate can be verified.

4. **Prevents attack :**
- It prevents man-in-the-middle attacks, where a malicious user pretends to be the web site that the user wants to connect with.

5. **Non-Repudiation :**
- The signature on the certificate guarantees that only the owner of web site has the private key associated with the public key in the certificate.

6. **Verification :**
- The process of verification and authentication in certificate is transparent to the end user and the process of authentication takes only a few milliseconds.

7. **Support :**
- Digital certificates are supported by many enterprise networks and applications.

5.14.6 Disadvantages of Digital Certificates :

- The disadvantages of digital certificates are as follows :
  1. The digital certificate is the complete responsibility of owner or sender.
  2. It is easy to obtain the digital certificate if the computer system becomes vulnerable.
  3. When a web site changes the certificate, a browser does not give a warning.
  4. A user has to blindly trust that the OS developer has installed authentic root certificates and not fake certificates.
  5. A fake root certificate can be installed in the browser when a malicious user gains access to the personal computer. In this case the browser will not report any security warning while browsing sites that use the fake certificate.

5.14.7 Comparison of Digital Signature and Digital Certificate :

- Table 5.14.1 gives the comparison of of Digital signature and Digital Certificate.

Table 5.14.1 : Comparison of Digital signature and Digital Certificate

Sr. No	Parameter	Digital signature	Digital Certificate
1.	Definition	Digital signature is like a fingerprint or an attachment to a digital document that ensures its authenticity and integrity	Digital certificate is a file that ensures holder's identity and provides security
2.	Process	Hashed value of original message in encrypted with sender's secret key to generate the digital signature	It is generated by CA that involves four steps: Key Generation, Registration, Verification, Creation
3.	Security Services	Authenticity of Sender, integrity of the document and non-repudiation	It provides security and authenticity of certificate holder
4.	Standard	It follows Digital Signature Standard (DSS)	It follows X.509 Standard
5.	Uses	Messaging, emails, and contracts	Used for initiating a secure connection between browser and server
6.	Algorithm used	It uses the RSA algorithm if there is a need for message encryption	It encrypts the data and only the receiver can decrypt it
7.	Visibility	Digital signatures are attachment to documents and show authenticity	Digital certificates are installed on websites and show owner identity
8.	Validation	Digital signatures validate the data sent	Digital certificates validate the sender's identity

5.15 Public Key Infrastructure (PKI) :

- Generally passwords are used for authentication in case of data transfer, but for the transfer of confidential information in distributed environment, a more secure authentication method is required.
- Such an authentication method must confirm the identity of the entities involved in the communication and must authenticate the transferred data that is being. One such method is the public key infrastructure.

Definition :

- It is the set of hardware, software, policies, processes, and procedures required to create, manage, distribute, use, store, and revoke digital certificates and public-keys.
- Public Key Infrastructure (PKI) is used to manage keys and certificates.
- PKI creates digital certificates that bind public keys to entities, stores them securely and withdraw them when required.
- Digital certificates in public key infrastructure are used to establish integrity and ownership of a public key.
- Public Key Infrastructure (PKI) is a collection of servers used to create and manage public keys and digital certificates.
- The PKI provides a secure environment for online transactions, confidential email and e-commerce by Authenticating the identity of the entities and maintaining the data integrity.
- Public key infrastructure provides an effective authentication method through digital certificates.
- It ensures a secure communication necessary in today's world as majority of the transactions are online.
- It provides secure exchange of confidential information.
- Digital certificates are used in various network applications and enterprises.
- Usage of certificates is not limited to personal computers, they are widely being used in smart phones, smart card and other devices as well.

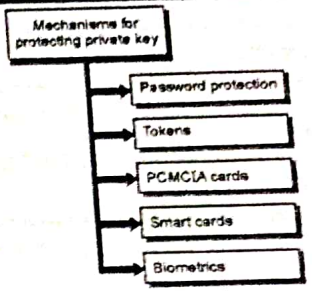
5.16 Private Key Management :

Definition :

- The private Key management is the process of administering or managing private cryptographic keys for a cryptosystem.
- Till now we have concentrated on the public key of a user.
- We know that a user must hold the private key secretly. Any user should not access someone's private key.

5.16.1 Mechanisms for Protecting Private Keys :

- A private key be protected by using the several mechanisms as shown in Fig. 5.16.1.



(G-2978)**Fig. 5.16.1 : Mechanisms for Protecting Private Keys**

- Following are the mechanisms for Protecting Private Keys :

1. **Password protection :**
- The password protection is the simplest and most common method to protect a private key.
- IN this mechanism, the private key is stored on the hard disk of the user's computer as a disk file.
- The user can access this file only with the help of a password or a Personal Identification Number (PIN).
- But, this method is considered as the less secure method because anyone who can guess the password correctly can access the privata key.

2. **Tokens :**
- The token stores the private key in an encrypted format.
- A one-time password is required to decrypt and access the key. A one-time password is valid only for that particular access, this password becomes invalid next time, and another OTP must be used.
- This is a more secure method.

3. **PCMCIA cards :**
- PCMCIA stands for the Personal Computer Memory Card International Association (PCMCIA). These cards are chip cards.
- The private key is stored on PCMCIA card, which means there is no need to store key on user's hard disk. This reduces the chances of key being stolen.
- There is scope for key being captured by an attacker as, in case of signing or encryption operation, the key must travel from the PCMCIA card to the memory of the user's computer.

4. Smart cards :

- A smart card stores the private key of the user in a tamperproof card.
- A computer chip is also included in the smart card, which can perform cryptographic functions like signing and encryption.
- The main advantage of this scheme is that the private key never leaves the smart card.
- The disadvantage of the smart card is that the user must carry the smart card with her and only compatible smart card readers can access it.

5. Biometrics :

- In this mechanism, the private key is associated with a unique characteristic of an individual.
- Unique characteristic of an individual can be a fingerprint, retina scan or voice comparison.
- This method is similar to the token, but here the user need not carry anything with him.

Use of PKCS#12 standard :

- In many situations, the private key of the user might be required to be transported from one location to another.
- E.g. Suppose the user wants to change his/her PC.
- To handle such situations, there is a cryptographic standard PKCS#12 which allows a user to export her digital certificate and private key in the form of a computer file.
- The certificate and the private key should be protected as they are moved to another location.
- The PKCS#12 (Public Key Cryptography standard#12) guarantees that the certificate and the private key are encrypted using a symmetric key which is derived from the user's private key protection password.

5.16.2 Multiple Key Pairs :

- In case of serious business applications, users should have multiple digital certificates, which also means the pair of multiple keys.
- In the multiple key pairs, it is expected that one certificate could be strictly used for signing and another for encryption.
- Due to use of such pair of keys, the loss of one of the private keys does not affect the complete operations of the user.
- The following are guidelines for multiple key pairs :
  1. The private key used for digital signing (non-repudiation) should not be backed up or archived after it expires. It should be destroyed. This will make sure that it is not used by someone else for signing on behalf of the person in the future.
  2. There should be backup of the private key used for encryption/decryption after its expiry. Due to backup of key, the encrypted data can be recovered even in the future.

5.16.3 Key Update :

- The key pairs must be updated periodically as the keys becomes susceptible to cryptanalysis attacks over a time.
- It causes a digital certificate to expire after a certain time. To avoid this, key update is necessary.
- Following are the ways to deal with the expiry of a certificate :
  1. Based on the original key pair, the CA reissues a new certificate. This is not recommended unless there is an confidence in the strength of the original key pair.
  2. Another fresh key pair is generated and the CA issues a new certificate based on that new key pair.

Handling of Key update process :

- The key update process can be handled in the following two ways :
  1. The end user has to detect about the expiry of the certificate and request the CA to issue a new one.
  2. Whenever used, the expiry date of the certificate is automatically checked every time and its renewal request is sent to the CA as soon as it is about to expire. Special systems are required for this approach.

5.16.4 Key Archival :

- The CA should have plan for and maintain the history of the certificates and the keys of its users.
- E.g. If someone approaches the CA of Alex and requests for the digital certificate of Alex used three years back to sign a legal document for verification purposes.

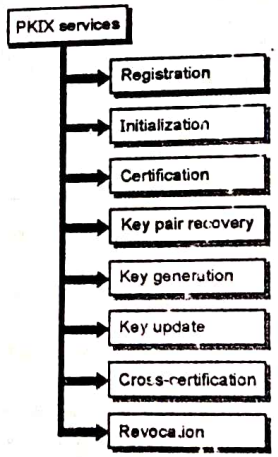
- The CA cannot provide the same, if the CA has not archived the certificates and can cause serious legal problems. So, key archival is a very important aspect of any PKI solution.

5.17 The PKIX Model :

- PKIX stands for Public Key Infrastructure X.509.
- The X.509 standard defines the digital certificate structure, format and fields.
- X.509 standard also defines the procedure for public key distribution.
- The Internet Engineering Task Force (IETF) formed the Public Key Infrastructure X.509 (PKIX) working group to extend such standards and make them universal.
- PKIX modes extends the basic philosophy of the X.509 and specifies how the digital certificates can be deployed in the Internet world.
- Other PKI models have been defined for the use by applications in various domains.
- E.g. The financial institutions makes use of the ANSI, ASC X9F standards.

5.17.1 Services Provides by PKIX :

- The services provided by PKIX are as shown in Fig. 5.17.1.



(G-2981)**Fig. 5.17.1 : PKIX Services**

Registration :

- In this process, an end-entity (subject) makes itself known to a CA through RA.

Initialization :

- This service deals with the basic problems, such as how the end-entity is sure that it is talking to the right CA?

Certification :

- The CA produces a digital certificate for the end-entity and returns it to the end-entity.
- It maintains a copy for with it for its own record and also copies it in public directories, if required.

Key pair recovery :

- To decrypt some old documents, the Keys used for encryption may be required to be recovered at a later date .
- The CA or an independent key recovery system provides the key archival and recovery services .

Key generation :

- PKIX model specifies that the end-entity should be able to generate private and public key pairs or the CA/RA should be able to generate private and public key pairs for the end-entity.
- The generated private and public key pairs are then distribute to the end-entity securely.

Key update :

- It is a process where the expired key of the digital certificate is automatically renewed and replaced with a new key pair.
- Here, the provision is made for manual digital certificate renewal requests and responses.

Cross-certification :

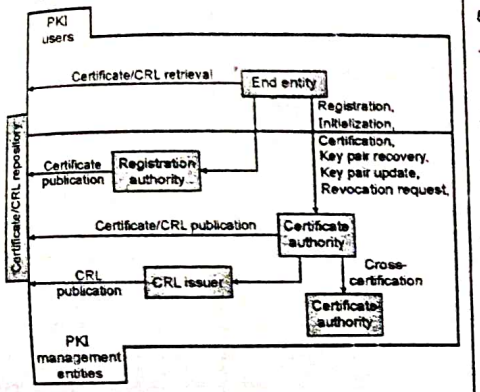
- In this process the end entities that re-certified by different CA, can cross verify each other. It helps in establishing trust models.

Revocation :

- PKIX model provides support for the checking the status of the certificate in two modes : online by using Online certificate status protocol (OCSP) or offline by using the Certificate Revocation List (CRL).

5.17.2 Components of PKIX Model :

- Fig. 5.17.2 shows the interrelationship between the key elements of the PKIX model.



(G-2982)Fig. 5.17.2 : Components of PKIX model

- The key elements of PKIX model are as follows :
  1. End entity
  2. Certification authority (CA)
  3. Registration authority (RA)
  4. CRL issuer
  5. Repository

1. End entity :

- The subject field of a public key certificate identifies a general term used to denote end users, devices (e.g., servers, routers), or any other entity.
- The function of an end entity is to consume and/or support PKI-related services.

2. Certification authority (CA) :

- The CA is the issuer of certificates and usually certificate revocation lists (CRLs).
- CA supports administrative functions, even though these are often delegated to one or more Registration Authorities.

3. Registration authority (RA) :

- Registration authority is an optional part that can assume a number of administrative functions from the CA.
- The RA is always associated with the registration process of end entity. It can assist in a number of other areas as well.

4. CRL issuer :

- CRL issuer is an optional component that a CA can hand over to publish CRLs.

5. Repository :

- The repository component stores the certificates and CRLs so that they can be retrieved by end entities.

5.17.3 PKIX Architectural Model :

- PKIX has developed a document that describes five areas of its architectural model. These areas are as follows:

X.509 V3 certificate and V2 certificate revocation list profiles :

- We know that, the X.509 standard allows the use of various options while describing the extensions of a digital certificate.
- PKIX has grouped all the options that are considered fit for Internet users. This group of options is called as the profile of the Internet users.
- This profile of the Internet user is described in RFC2459 and specifies which attributes are supported and which are not.
- It also provides ranges of appropriate value for the values used in each extension category.
- E.g. The instruction code is not specified in the basic X.509 standard when a certificate is suspended or put on hold, PKIX defines the instruction code.

Operational protocols :

- These protocols provide the transport mechanism for delivering certificates, CRLs and other management and status information to a PKI user.
- As each requirement demands a different way of service, the use of HTTP, LDAP, FTP, X.500, etc. are defined for this purpose.

Management protocols :

- These protocols allows data exchange between the various PKI entities.
- E.g. How to carry registration requests, cross-certification requests and responses or revocation status.
- The structure of the messages that float between the entities is specifies by these protocols.

- These protocols also specify details required for message processing.
- The Certificate Management Protocol (CMP) is an example of management protocol. CMP is used for requesting a certificate.

Policy outlines :

- In RFC2527, PKIX defines the outlines for Certificate Policies (CP) and Certificate Practice Statements (CPS).
- PKIX defines the policies for the document creation such as a CP. It determines what are important considerations when selecting a type of certificate for a particular application domain.

Timestamp and Data certification services (DCS) :

- A trusted third party provides a **Times tamping service** known as Time Stamp Authority.
- The main purpose of the timestamp service is to sign a message to promise that it existed prior to a specific date and time. This service is helpful to deal with non-repudiation claims.
- The **Data Certification Service (DCS)** is a trusted third party service.
- DCS verifies the correctness of the data that it receives.
- DCS service similar to the notary service in real life, where one can use it for getting one's property certified.

Review Questions

Q. 1 Write a note on : Data encryption standards.

Q. 2 What is public key cryptography ?

Q. 3 State advantages and disadvantages of public key cryptography.

Q. 4 Write a note on : RSA.

Q. 5 Compare secret key system and public key cryptosystems.

Q. 6 Write a note on : Digital signature.

Q. 7 With the help of a neat block diagram, explain the use of message and message digest to preserve the message integrity.

Q. 8 What is a digital signature ? Explain its use in providing messages integrity and message authenticity.

Q. 9 With the help of a neat block diagram, explain the digital signature process.

Q. 10 State and explain the various services provided by the digital signature.

Q. 11 Briefly define a group.

Q. 12 Briefly define a ring.

Q. 13 Briefly define a field.

Q. 14 Explain the term prime numbers.

Q. 15 Check whether 89 is a prime.

Q. 16 Explain various characteristics of hash functions.

Q. 17 Explain Diffie-Hellman Key Exchange Algorithm in steps.

Q. 18 How digital signatures work ? Explain with a neat diagram.

Q. 19 Explain how RSA signature scheme generate and verify the digital signature.

Q. 20 Explain how Digital Signature Standard (DSS) scheme generate and verify the digital signature.

Q. 21 What are the typical contents of a digital certificate ?

Q. 22 What is the role of a CA and a RA ?

Q. 23 Name the four key steps in the creation of a digital certificate.

Q. 24 Describe the mechanisms of protecting the private key of a user.

Q. 25 Explain the main concepts in DES.

Q. 26 How can the same key be reused in triple DES ?

Q. 27 How does the one-time initialization step work in AES ?

Q. 28 Explain the steps in the various rounds of AES.

Q. 29 Explain various modes of operation for DES.

Q. 30 State the major attributes of AES.

Q. 31 Write a short note on : Private key management.

Q. 32 Explain PKIX model.

Q. 33 Explain services provided by PKIX.

□□□