Q 1) What do you understand by Network Security and why is it required? Explain principles of Network Security. Explain various types of attacks.

NETWORK SECURITY:
Network Security is the identification and mitigation of undesirable information flow. Individuals as well as organizations are becoming increasingly dependent on networks to carry on their activities. Networks have become assets like computers, data. Without these assets most organizations, individuals find it impossible to conduct day to day work. The network asset must be protected like other assets and surrounded with proper controls and appropriate security. Network security is needed to lay down appropriate measures to prevent the network from attacks.

PRINCIPLES OF NETWORK SECURITY:

CONFIDENTIALITY:
The principle of confidentiality specifies that only the sender and the intended recipient should be able to access the contents of a message. Confidentiality gets compromised if an unauthorized person is able to access a message. Here, the user of computer A sends a message to user of computer B. Another user C gets access to this message, which is not desired and therefore defeats the purpose of confidentiality. For example a confidential email message, sent by A to B, which is accessed by C without the permission of A or B.

AUTHENTICATION:
Authentication mechanism's help establish proof of identities. The authentication process ensures the origin of an electronic message or document is correctly identified. For instance, user C posing as user A sends a funds transfer request (form A's account to C's account) to bank B. The bank might transfer the funds from A's account to C's account thinking user A has requested for the funds transfer.

INTEGRITY:
When the contents of a message are changed after the sender sends it, but before it reaches the intended recipient, we say that the integrity of a message is lost. For example, a check is issued for Rs 100. In the next account statement the same check resulted in a payment of Rs 10000. This is a case of loss of message integrity.

REPUDIATION:
There are situations where a user sends a message and later on refuses that she had sent that message. For example user A could send a funds transfer request to bank B over the internet. After the bank performs the funds transfer as per A's instruction. A could claim that she never sent the funds transfer instruction to the bank. Thus A repudiates or denies her fund's transfer

instruction. The principle of non-repudiation defeats such possibilities of denying something, having done it.

ACCESS CONTROL:

The principle of access control determines who should be able to access what. For example, we should be able to specify that user A can view the records in the database, but cannot update them. However user B might be allowed to make updates as well. An access control mechanism can be setup to ensure this. Access control is broadly related to role management and rule management. Role management concentrates on the user side, whereas rule management focuses on the resources side.

AVAILABILITY:

The principle of availability states that resources should be available to authorized parties at all times. For example, due to intentional actions of an unauthorized user C, an authorized user A may not be able to contact a server computer B. This would defeat the principle of availability. These attacks are further grouped into two types passive attacks and active attacks.

PASSIVE ATTACKS:

Passive attacks are those, wherein the attacker indulges in monitoring of data transmission. The attacker aims to obtain information that is in transit. The term passive indicates that the attacker does not attempt to perform any modifications to the data. Passive attacks are harder to detect, therefore passive attacks should be prevented rather than detected. Passive attacks are further classified into two categories release of message contents and traffic analysis.

RELEASE OF MESSAGE CONTENTS:

When we send a confidential message to our friend we desire that only he/she be able to access it. Otherwise the contents of the message are released against our wishes to someone else.

TRAFFIC ANALYSIS:

If we send multiple messages to our friend, a passive attacker could try to figure out similarities between them to come up with some sort of pattern that provides some clues regarding the communication that is taking place. Such attempts of analyzing encoded messages to come up with likely patterns are the work of the traffic analysis attack.

ACTIVE ATTACKS:

Active attacks are based on modification of the original message in some manner or the creation of a false message. These attacks cannot be prevented easily. However, they can be detected with some effort and attempts can be made to recover from them. Active attacks can be subdivided into four categories masquerade, replay, modification, and denial of service.

MASQUERADE

Masquerade is caused when an unauthorized entity pretends to be another entity. Consider three users A, B, C. User C might pose as user A and send a message to user B. User B might be led to believe that the message indeed came from user A. A masquerade attack usually includes one of the other forms of active attacks. For example the attack may involve capturing user's authentication sequence(eg user id and password). Later, those details can be replayed to gain illegal access to the computer system.

REPLAY ATTACK:

In replay attack, a user captures a sequence of events or some data units and resends them. For instance, suppose user A wants to transfer some amount to user C's bank account. Both users A and C have accounts with bank B. User A might send an electronic message to bank B, requesting for the funds transfer. User C could capture this message and send a copy of the same to the bank B. Bank B would have no idea that this is an unauthorized message and would treat this as a second and different, funds transfer request from user A. User C gets benefit of funds transfer twice, once authorized, once through a replay attack.

ALTERATION OF MESSAGES:

It means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect. For example user A sends an electronic message Transfer 1000 Rs to D's account to bank B. User C might capture this and change it to Transfer 10000 Rs to C's account.

DENIAL OF SERVICE:

In these attacks, attempt is made to prevent legitimate user's from accessing some services, which they are eligible for. For example an unauthorized user might send too many login requests to a server using random user id's one after the other in quick session, so as to flood the network and deny other legitimate users from using the network facilities.

Q 2) Explain different types of Security Policy?

Ans. Security Policy**:** A security policy is the set of decisions that collectively determines an organization attitude towards security.

A security policy defines the boundaries of acceptable behavior and what the response to violations should be.

Security policy is the set of decisions/rules & regulation written or verbally understood that collectively determines organizations posture towards security. It delimits the boundaries, specifies acceptable & non acceptable behavior dictates what is ethical and what is non-ethical. States the degree of seriousness of the offence and also mention the consequences of the actions if it is violated. It focuses on different stack holders and satisfying their requirements in an efficient way. Organisation differs in their culture,structure and strategy. Thus security policy will also differ from organization to organization.

Security policy will decide on the following issues:

- What legal course of action will you follow if attacked?

- What will be considered as a cognizable crime?

- Can anyone be sued?

- Infringing on someone else's rights?

A security policy holds guidelines or rules within itself that need to be enforced as to ensure a secured network. A good security policy generally takes care of four key aspects:

- Affordability

- Cultural issues

- Legality

- Functionality

There are mainly four types of security policies namely:

- Confidentiality policy

- Integrity policy

- Military security policy

- Commercial policy

1. Confidentiality Policy: This type of policy is concerned with the privacy of an organization. As the name suggest,it deals with the issues related to confidentiality of a system. At times, there is certain data that is required to be kept secret.This is achieved using the norms of the confidentiality policy failing to which the data is considered to be unsecured and unprotected.

2. Integrity Policy: This type of policy is concerened with the integrity issues of an organization. The data that is to be used by organization should not be tampered in any possible way. This is taken care with the help of integrity policy which ensures that right data moves in and out of the organization.

3. Military Security Policy: It is also known as government security policy. This security policy is developed primarily to provide confidentiality. The name comes from the military's need to keep information such as the data that a troop will sail,secret. This policy also considers integrity and availiability as an important part of the policy along with confidentiality being its key issued because the compromise of this feature would be catastrophic for the organization. Unauthorize disclosure can result in penalties that include jail or fines etc.

4. Commercial Security Policy: This is a security policy developed primarily to provide integrity.The name comes from the need for commercial firm to prevent tampering with the data, because they could not survive such compromises. For example, if the confidentiality of a bank's computer is compromised, a customers account balance may be revealed. This would certainly embarrass the bank, but the loss to the bank's data integrity would lead to financially ruinous effects. This policy gives impetus on integrity along with confidentiality and availiability as its other issue.

Some integrity policies use the notion of a transaction like database specifications which require the database to be in a consistent state. Such policies are called transaction-oriented integrity security policies.

Q 3) What are different modes of algorithm? Explain

Ans- An algorithm mode is a combination of a series of the basic algorithm steps on block chipper and some kind of feedback from the previous step.

There are four important algorithm modes as follows,

I. Electronic Code Book (ECB)
II. Cipher Block chaining (CBC)
III. Cipher Feedback (CFB)
IV. Output Feedback (OFB)

The first two modes operate on block cipher, whereas the latter two modes are block cipher modes, which can be used as if they are working on stream cipher.

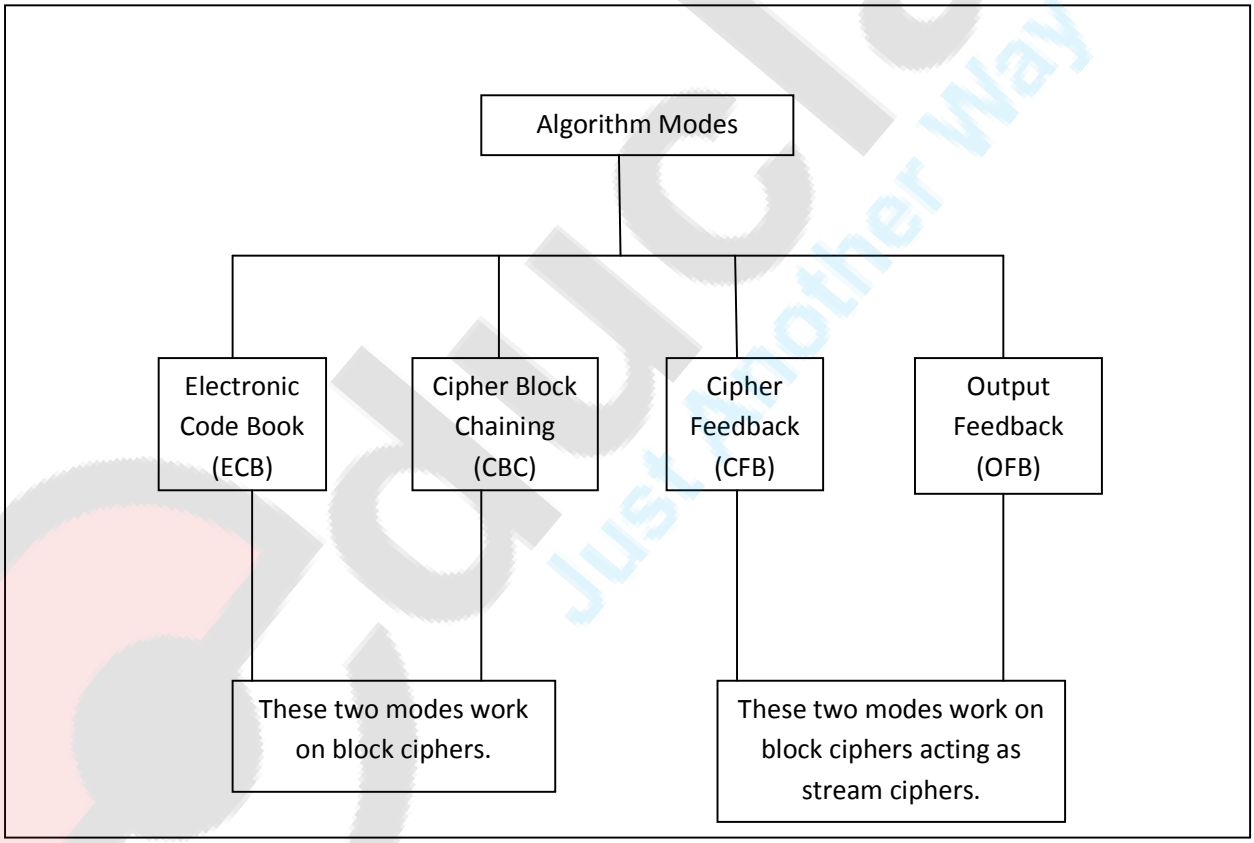


Fig. Algorithm modes

I. Electronic Code Book (ECB) mode:

It is the simplest mode of operation. Here, the incoming plain text message is divided into blocks of 64 bits each. Each such block is then encrypted independently of the other blocks. For all blocks in a message, the same key is used for encryption. The process is shown as follows:

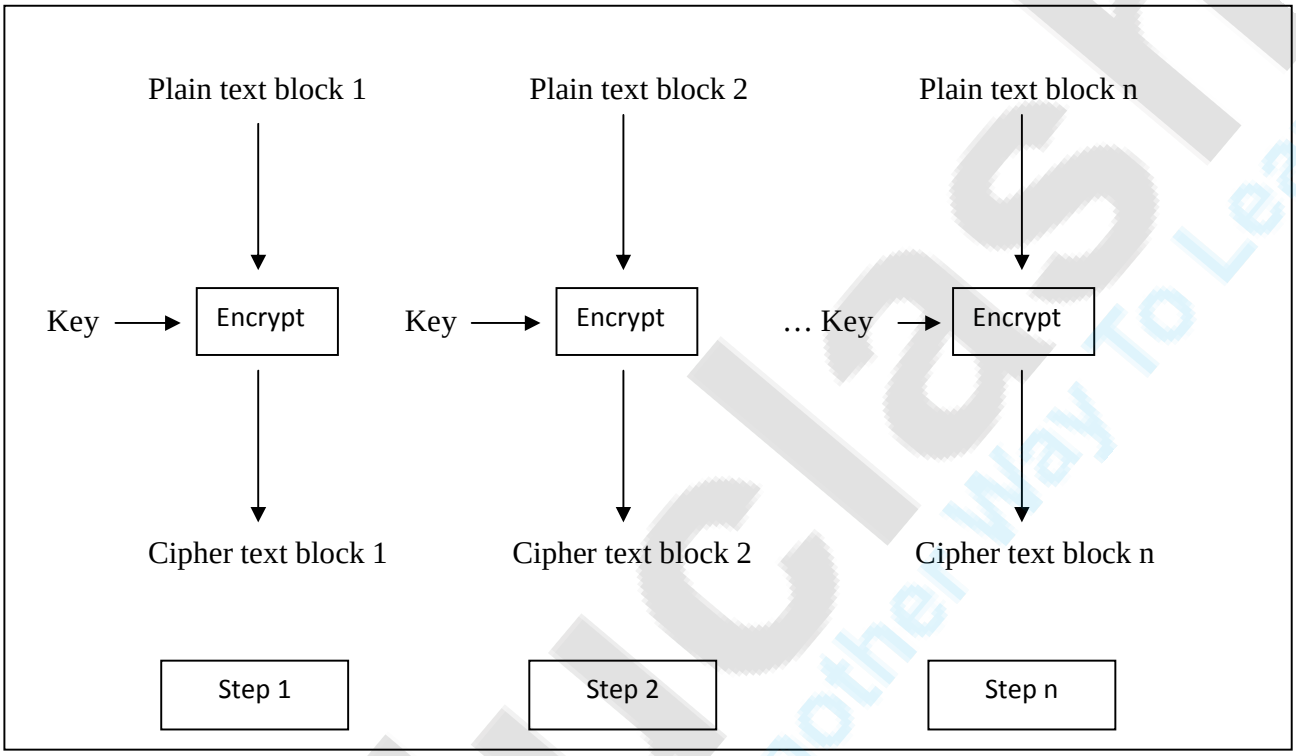


Fig. ECB mode-The encryption process

At the receiver's end, the incoming data is divided into 64 bit blocks and by using the same key as was used in encryption, each block is decrypted to produce the corresponding plain text block. The process is shown in fig.In ECB, since a single key is used for encrypting all the blocks of a message, if a plain text block repeats in the original message, the corresponding cipher text block will also repeat in the encrypted message. Therefore, ECB is suitable only for encrypting small messages, where the scope for repeating the same plain text block is quite less.

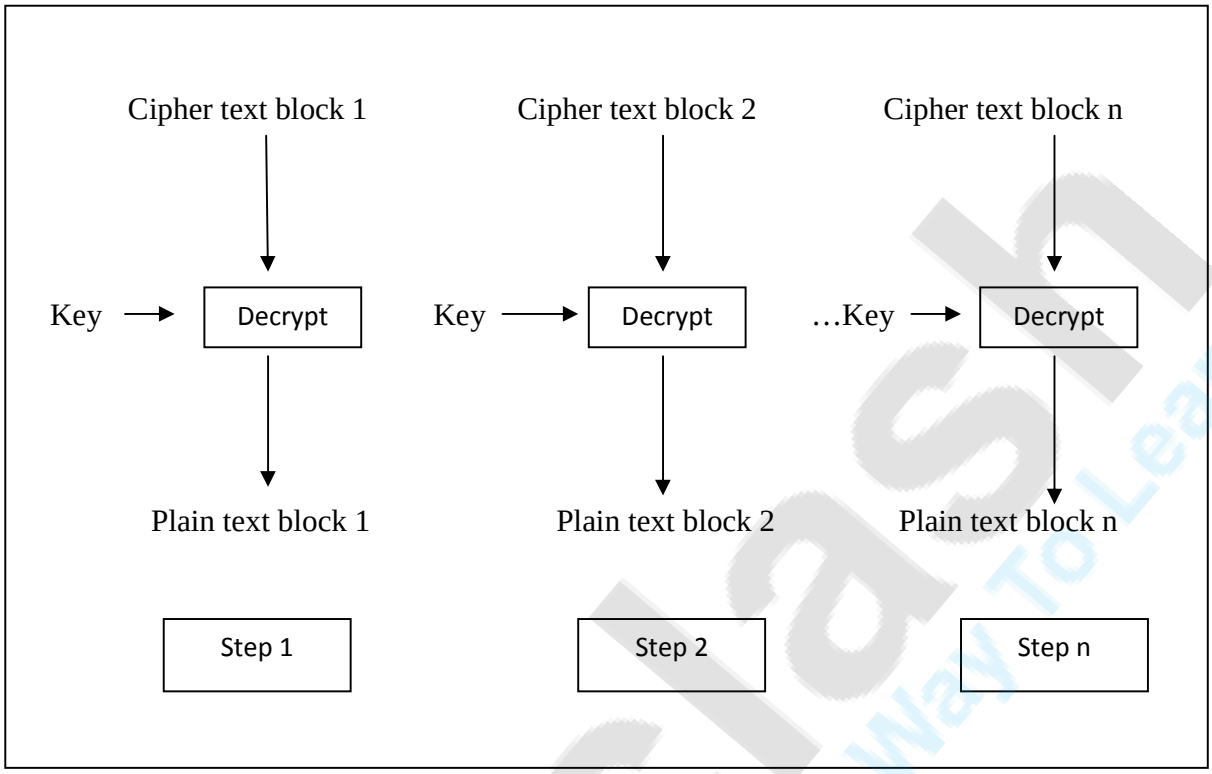


Fig. ECB mode-The decryption process

II. Cipher Block chaining (CBC) mode:

In case of ECB, within a given message, a plain text block always produces the same cipher text block. Thus, if a block of plain text occurs more than once in the input, the corresponding cipher text block will also occur more than once in the output, thus providing some clues to a cryptanalyst.

To overcome this problem, the Cipher Block chaining (CBC) mode ensures that even if a block of plain text repeats in the input, these two or more identical plain text blocks yield totally different cipher text blocks in the output. For this a feedback mechanism is used. Chaining adds a feedback mechanism to a block cipher.

In CBC the result of the encryption of the previous block are fed back into the encryption of the current block.

That is, each block is used to modify the encryption of the next block.

Thus, each block of cipher text is dependent on the corresponding current input plain text block as well as all the previous plain text blocks.

The encryption process of CBC is depicted in following fig.:

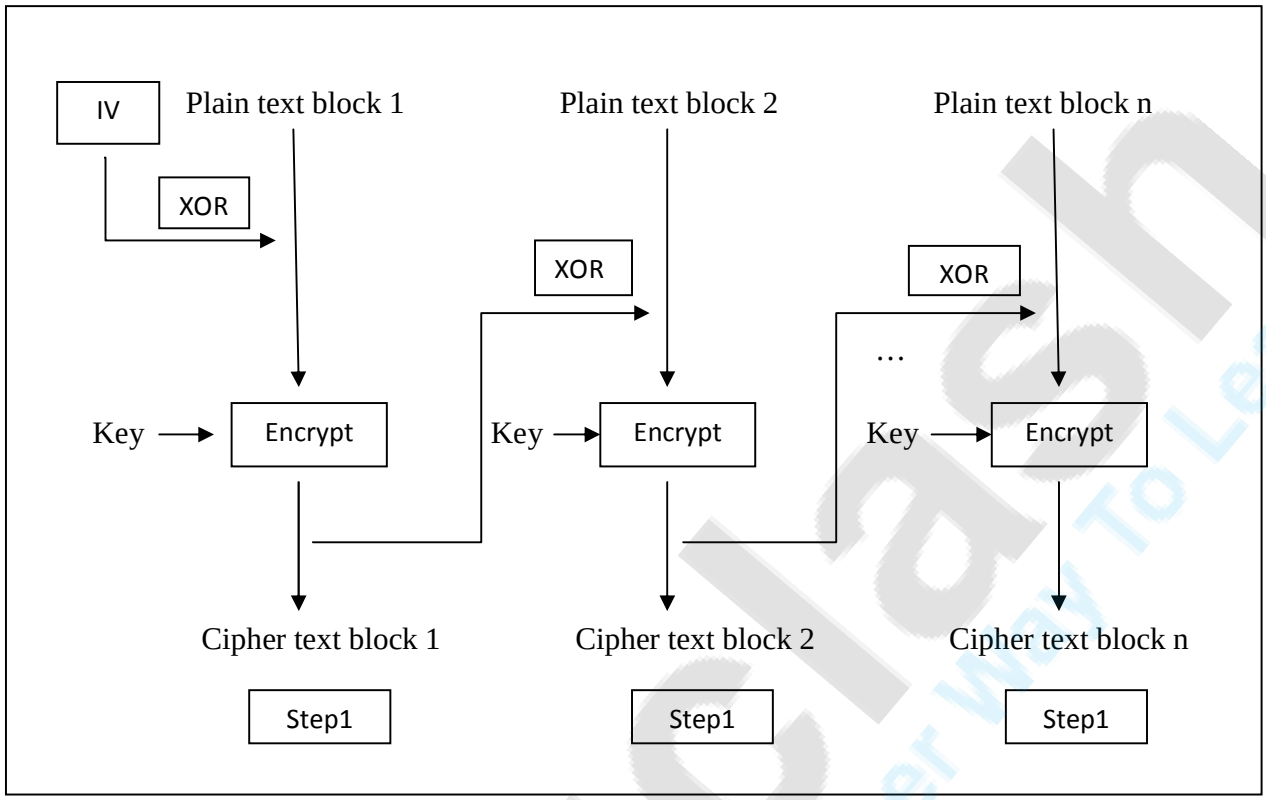


Fig. CBC mode – The encryption process

1. As shown in fig, the first step receives two inputs:
   The first block of plain text and a random block of text called as Initialization Vector (IV).
   a) The IV has no special meaning; it is simply used to make each message unique. Since the value of IV is randomly generated, the likelihood of it repeating in two different messages is quite rare. It is not mandatory to keep IV secret.
   b) The first block of cipher text and IV are combined using XOR and then encrypted using a key to produce the first cipher text block. The first cipher block then provided as a feedback to the next plain text block.
2. In the second step, the second plain text block is XORed with the output of step1, i.e. the first cipher text block. It is then encrypted with the same key, as used in step 1. This produces cipher text block 2.
3. In the third step, the third plain text block is XORed with the output of step 2, i.e. the second cipher text block. It is then encrypted with the same key, as used in step 1.

4. This process continues for all the remaining plain text blocks of the original message.

The decryption process works as follows:

1. The cipher text block 1is passed through the decryption algorithm using the same key, which was used during the encryption process for all the plain text blocks. The output of this step is then XORed with the IV. This process yields plain text block 1.
2. In step 2, the cipher text block 2 is decrypted, and its output is XORed with cipher text block 1, which yields plain text block 2.
3. This process continues for all the cipher text blocks in the encrypted message.

The decryption process is shown as follows:

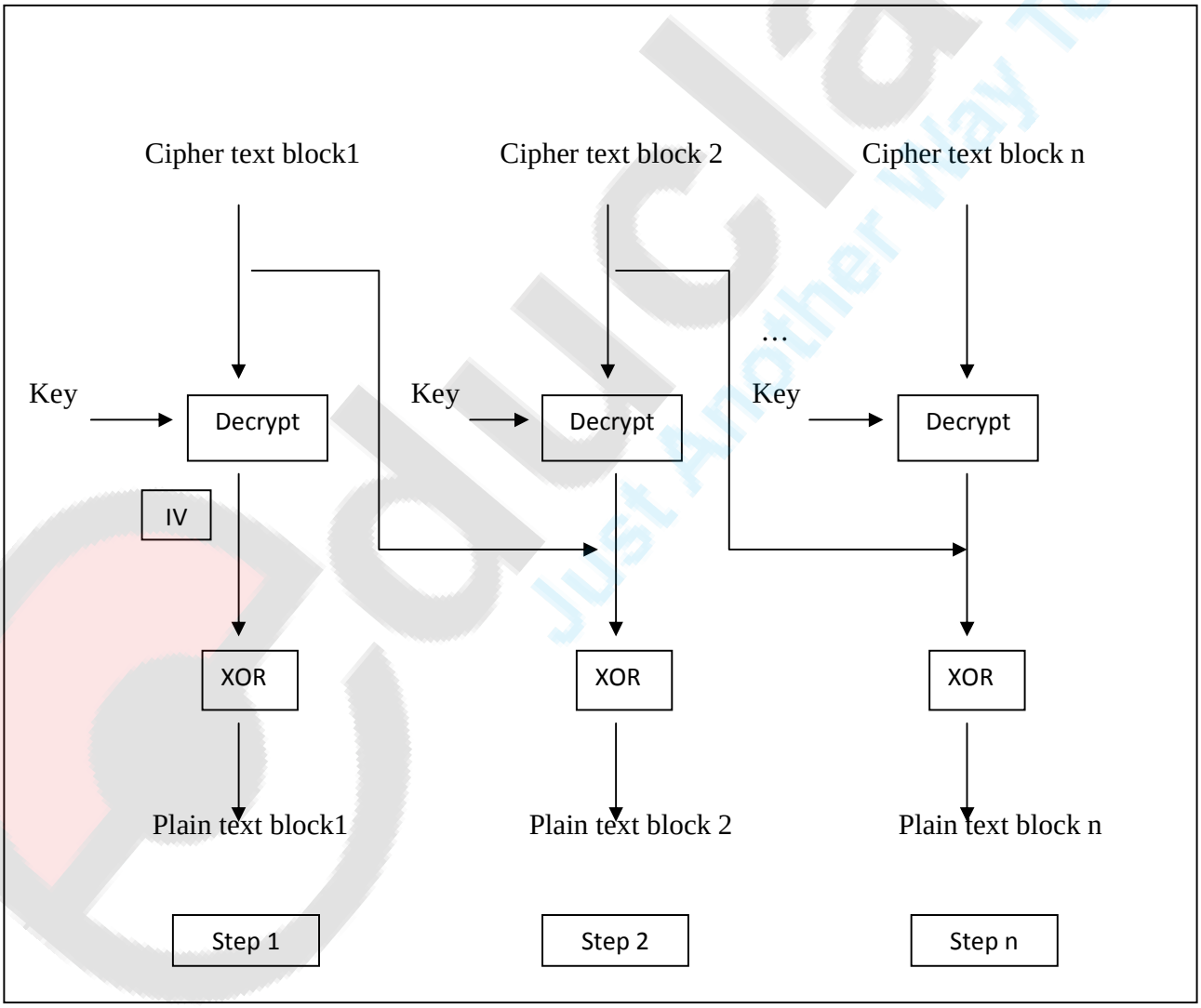


Fig. CBC mode- The decryption process

III. Cipher Feedback (CFB) mode:

Not all applications can work with blocks of data. Security is also required in applications that are character-oriented. The Cipher Feedback (CFB) mode is useful in such cases. In this mode data is encrypted in units that are smaller than a defined block size.

CFB works as follows;

Step 1:

Like CBC, a 64-bit IV is used in the case of CFB mode. The IV is kept in a shift register. It is encrypted in the first step to produce a corresponding 64-bit IV cipher text as shown in following fig.

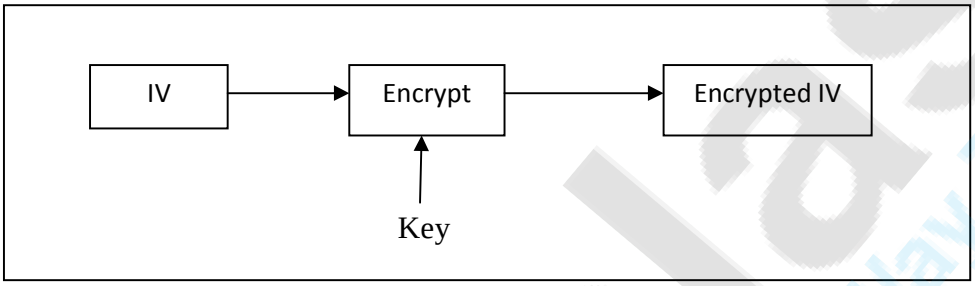


Fig.-CFB-Step 1

Step 2:

Now, the leftmost j bits of the encrypted IV are XORed with the first J bits of plain text. This produces the first portion of cipher text (say c) as shown in fig. c is then transmitted to the receiver.

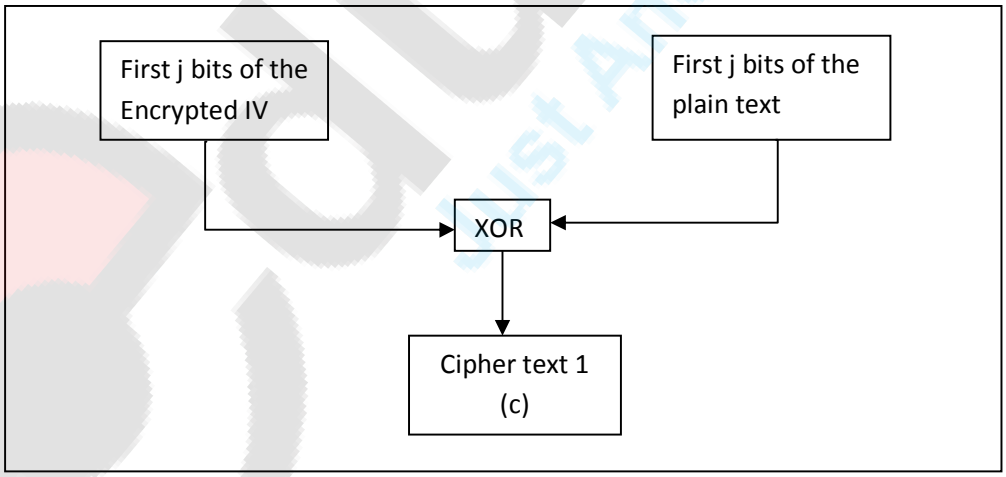


Fig.-Step 2

Step 3:

Now the bits of IV (i.e. the contents of shift register containing IV) are shifted left by j positions. Thus the rightmost j positions of the shift register now contain unpredictable data. These rightmost j positions are now filled with c. This is shown in following fig,

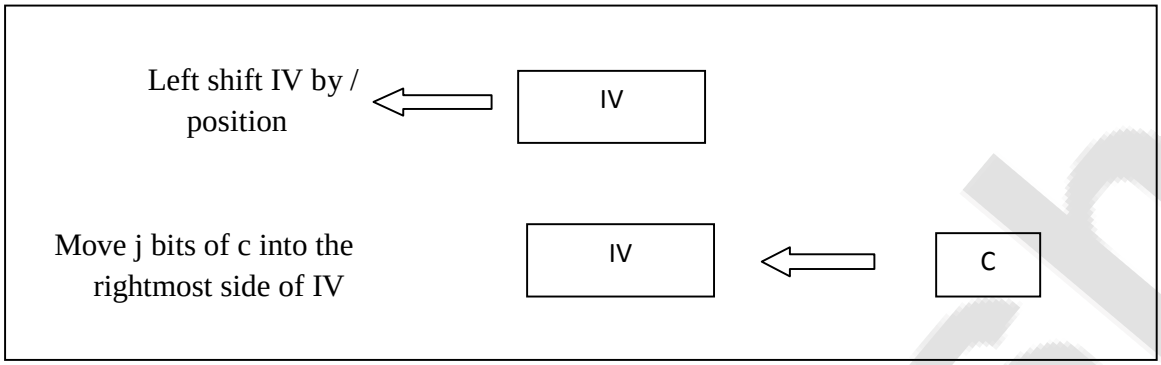


Fig. CFB-step 3

Step 4:

Now, Steps 1 through 3 continue until all the plain text units are encrypted. That is the following steps repeat:

- IV is encrypted.
- The leftmost j bits resulting from this encryption process are XORed with the next j bits of the plain text.
- The resulting cipher text portion is sent to the receiver.
- The shift register containing the IV is left-shifted by j bits.
- The j bits of the cipher text are inserted from right into the shift register containing the IV.

Following figure shows the overall conceptual view of the CFB mode.

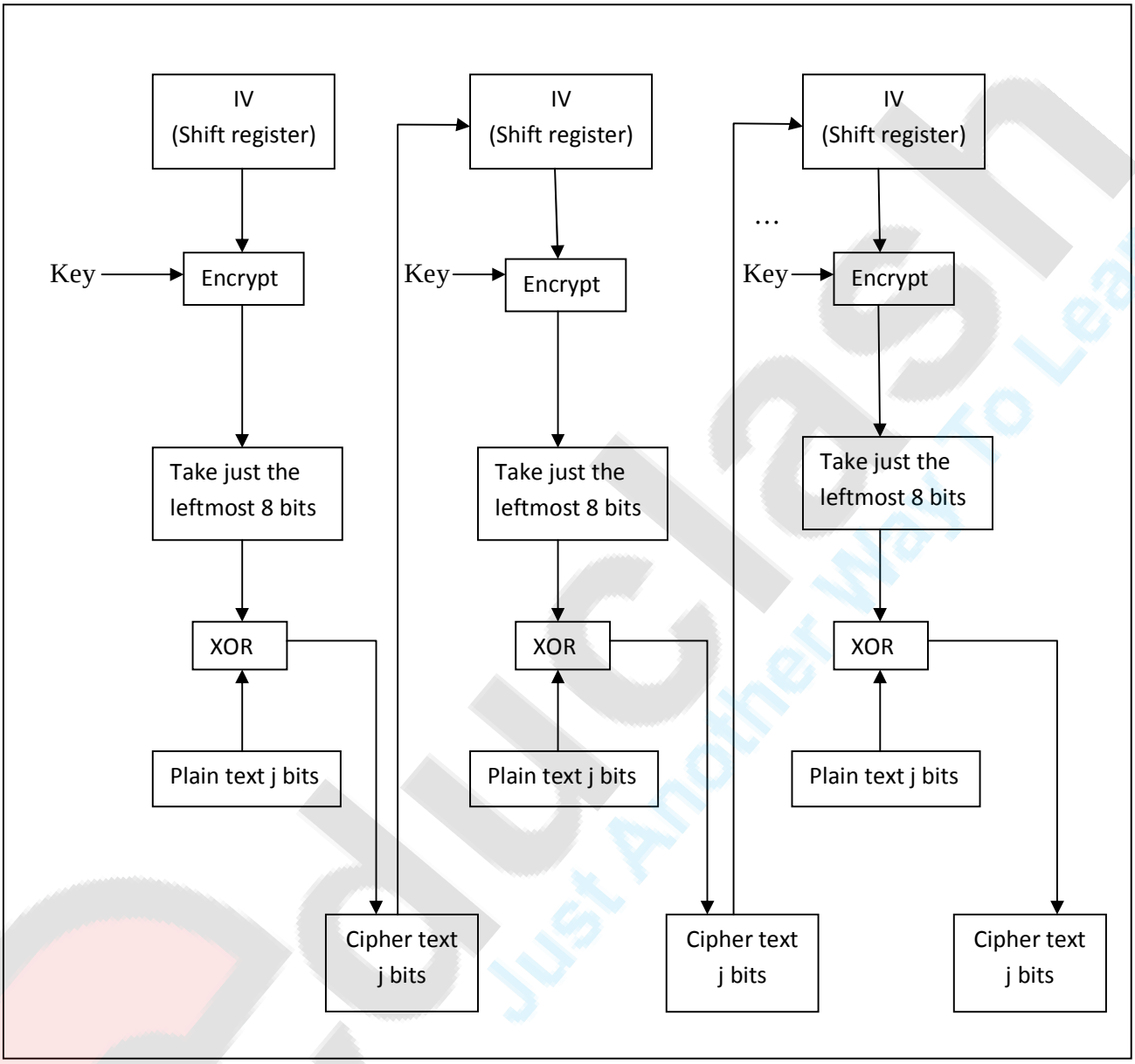


Fig. CFB-The overall encryption process

IV. Output Feedback (OFB) mode:

The OFB mode is extremely similar to the CFB. The only difference is that in case of CFB, the cipher text is fed into the next stage of encryption process but in case of OFB, the output of the IV encryption process is fed into the next stage of encryption process. Following figure shows the overall conceptual view of the CFB mode.

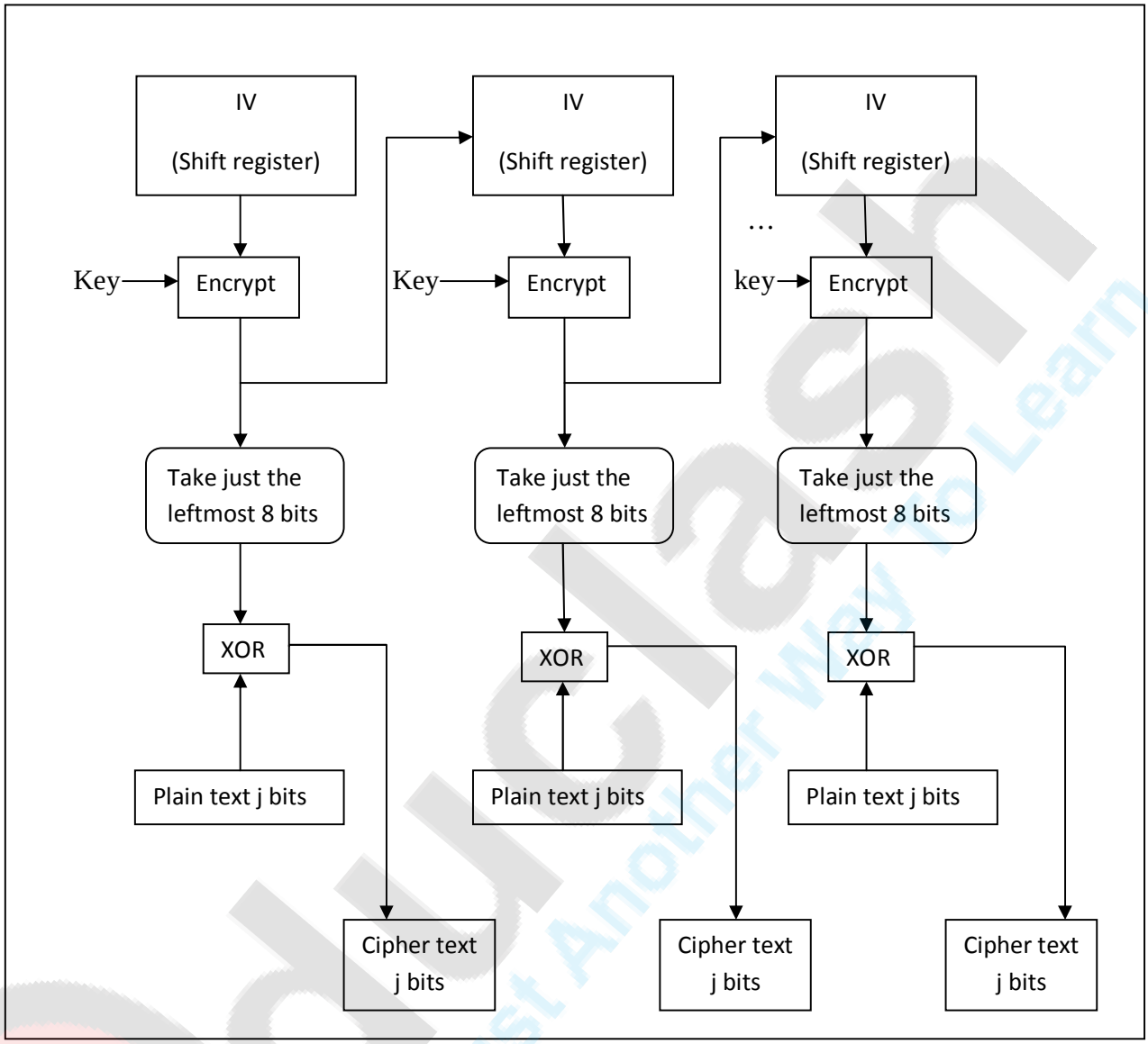


Fig. OFB-The overall encryption process

Q 4) What do you understand by Cryptography. Explain types Distinguish between symmetric and asymmetric cryptography .

Ans- :- Cryptography is an art of achieving security by encoding messages to make them non readable.

Within the context of any application-to-application communication, there are some specific security requirements, including:

- *Authentication:* The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)
- *Privacy/confidentiality:* Ensuring that no one can read the message except the intended receiver.
- *Integrity:* Assuring the receiver that the received message has not been altered in any way from the original.
- *Non-repudiation:* A mechanism to prove that the sender really sent this message.

Cryptography, then, not only protects data from theft or alteration, but can also be used for user authentication. In all cases, the initial unencrypted data is referred to as *plaintext.* It is encrypted into *ciphertext*, which will be decrypted into usable plaintext.

There are several ways of classifying cryptographic algorithms. They will be categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms are :

- Secret Key Cryptography (SKC): Uses a single key for both encryption and decryption
- Public Key Cryptography (PKC): Uses one key for encryption and another for decryption
- Hash Functions: Uses a mathematical transformation to irreversibly "encrypt" information

1. Secret key cryptography:

With *secret key cryptography*, a single key is used for both encryption and decryption., the sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver.

The receiver applies the same key (or ruleset) to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called *symmetric encryption*.

- With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, is the distribution of the key.
- Secret key cryptography schemes are generally categorized as being either *stream ciphers* or *block ciphers*.

- Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing.
- A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

2. Public-Key Cryptography

Public key cryptography depends upon the existence of *one-way functions*, or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute.

Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key.

One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext. The important point here is that it **does not matter which key is applied first**, but that both keys are required for the process to work Because a pair of keys are required, this approach is also called *asymmetric cryptography*.

In PKC, one of the keys is designated the *public key* and may be advertised as widely as the owner wants. The other key is designated the *private key* and is never revealed to another party. It is straight forward to send messages under this scheme. Suppose Alice wants to send Bob a message. Alice encrypts some information using Bob's public key; Bob decrypts the ciphertext using his private key. This method could be also used to prove who sent a message; Alice, for example, could encrypt some plaintext with her private key; when Bob decrypts using Alice's public key, he knows that Alice sent the message and Alice cannot deny having sent the message (*non-repudiation*).

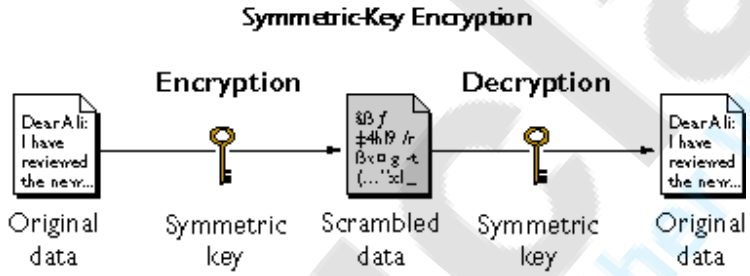
3.3. Hash Functions

Hash functions, also called *message digests* and *one-way encryption*, are algorithms that, in some sense, use no key (Figure 1C). Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms are typically used to provide a *digital fingerprint* of a file's contents, often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file.

DES	IDEA
1. DES is efficient to implement in h/w but relatively slow if implemented in s/w. 2.56 bit key is used to generate 16, 48 bit round key. 3. Same key are used in reverse order deriving decryption. 4. Each DES S-box maps a 6-bits qty into a 4-bits qty. 5. DES has 16 round. 6.56 bits key is expanded to generate 16 keys.	1. IDEA was designed to compute in s/w. 2. Uses 128 bit key. 3. Decryption odd round mathematical inverses of keys is used. Even round same key used while encrypt are used No inverse of key. 4. Each Primitive maps 16-bits qty into a 16-bits qty. 5. Has 17-round odd no round are diff from even numbered round. 6.128-bit key is expanded into 52 keys.

Q 5) What are pitfalls of symmetric key cryptography?

Ans: Cryptography or **cryptology**; from Greek, "hidden, secret, *gráphin*, "writing is the practice and study of hiding information. Crytography can be Symmetric or public key and Asymmetric or private key. Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way) Symmetric-key algorithms can be divided into stream ciphers and block ciphers. Stream ciphers encrypt the bits of the message one at a time, and block ciphers take a number of bits and encrypt them as a single unit. Blocks of 64 bits have been commonly used. **Symmetric cryptography** uses a single private key to both encrypt and decrypt data. Any party that has the key can use it to encrypt and decrypt data. They are also referred to as block ciphers. Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data. The keys tend to be much smaller for the level of protection they afford.



Disadvantages

Need for secure channel for secret key exchange:

Sharing the secret key in the beginning is a problem in symmetric key encryption. It has to be exchanged in a way that ensures it remains secret. The disadvantage of symmetric cryptography is that it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. This is a significant challenge. Symmetric algorithms are usually mixed with public key algorithms to obtain a blend of security and speed. Therefore, symmetric cryptography is effective only if the symmetric key cipher is kept secret by the two parties involved. If anyone else finds the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key cipher not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

Too many keys:

- A new shared key has to be generated for communication with every different party. This creates a problem with managing and ensuring the security of all these keys. In a given

system, each pair of system that wants to communicate must share individual private key. Thus the number of keys generated is far more than the keys generated in public key cryptography.

Origin and authenticity of message cannot be guaranteed:

- Since both sender and receiver use the same key, messages cannot be verified to have come from a particular user. This may be a problem if there is a dispute.
- Symmetric ciphers have historically been susceptible to known-plaintext attacks, chosen plaintext attacks, differential cryptanalysis and linear cryptanalysis

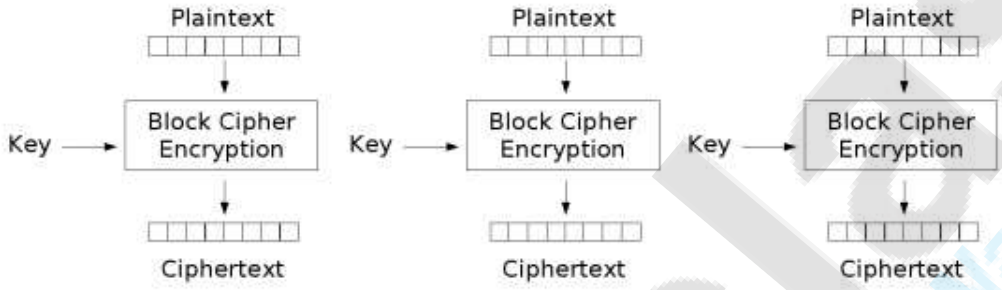
Q6) Explain different modes in which block and stream ciphers work?

Ans:

1. Electronic Code Book (ECB) Mode

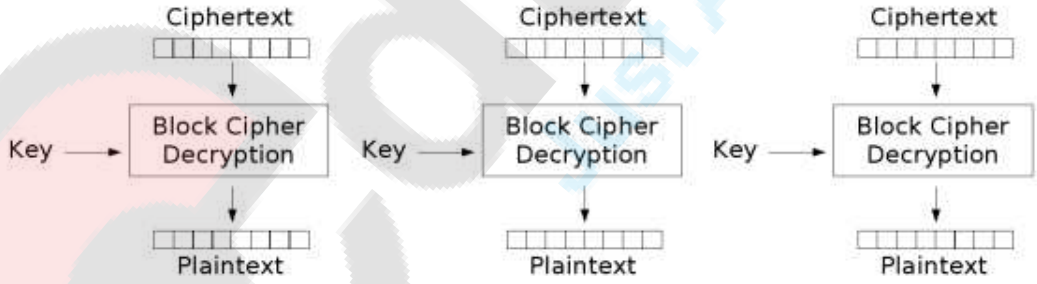
It is the simplest mode of operation. The incoming plain text message is divided into block of 64 bits each. Each such block Is then encrypted independently of the other block.

The same key is used for encryption, of all blocks in the message.



Electronic Codebook (ECB) mode encryption

At the Receiver end, the incoming data is divided into 64-bits blocks and by using the same key as was used for encryption , each block is decrypted to produce the corresponding plain text block.

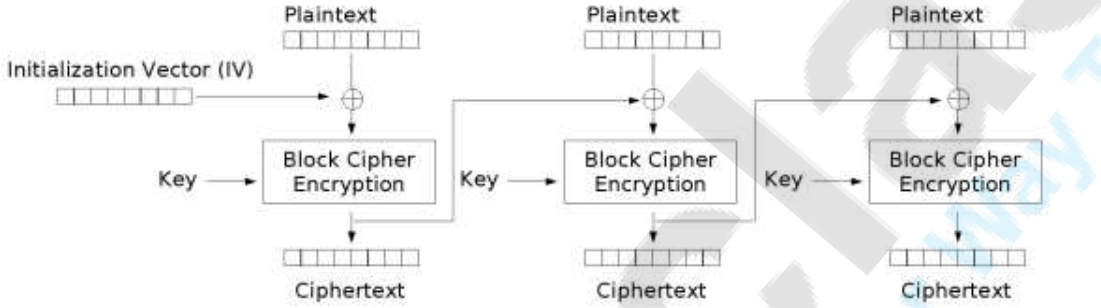


Electronic Codebook (ECB) mode decryption

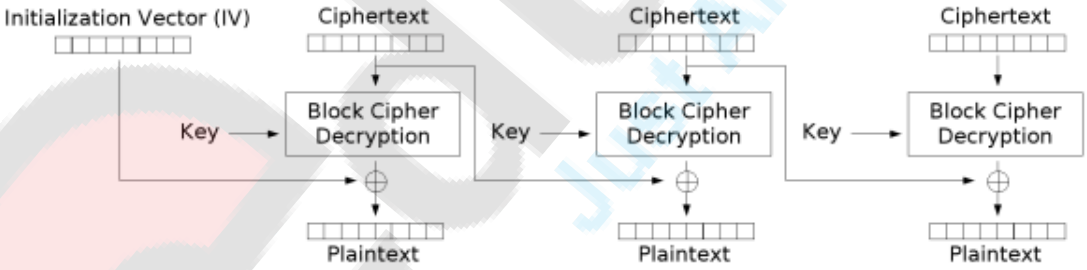
The disadvantage of this method is that identical plaintext blocks are encrypted into identical cipher text blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all.

2.Cipher Block Chaining (CBC) Mode

Cipher Block Chaining (CBC) is a mode of operation for a block cipher, one in which a sequence of bits is encrypted as a single unit or block with a cipher key applied to the entire block. Cipher block chaining uses what is known as an initialization vector (IV) of a certain length. CBC prevents the problems associated with Electronic Codebook (ECB), where every block of "plain text" maps to exactly one block of "cipher text" by having each encrypted block XORed with the previous block of cipher text. In this way, identical patterns in different messages are encrypted differently, depending upon the difference in the previous data.



Cipher Block Chaining (CBC) mode encryption

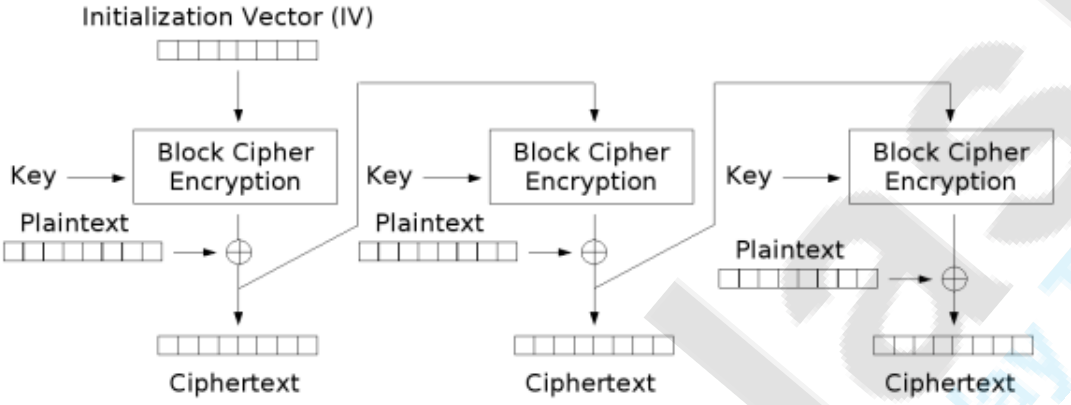


Cipher Block Chaining (CBC) mode decryption

The Initialization Vector (IV) is simply used to make each message unique. Since the value o IN is randomly generated, the likelihood of it repeating in two different message is quite rare. IV helps in making the cipher texts somewhat unique or at least quite different from all other cipher texts in a different message. It is not mandatory to keep IV secret.

3. Cipher Feedback (CFB) Mode

In this mode,data is encrypted in units that are smaller(e.g. they could be of size 8 bits i.e. the size of a character typed by an operator) than defined block size which is usually 64 bits.



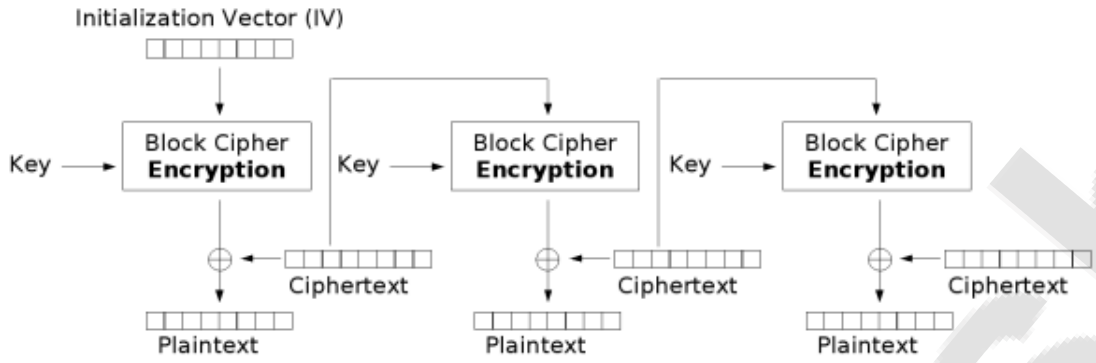
Cipher Feedback (CFB) mode encryption

i) Like CBC, a 64-bit IV is used in the case of CFB mode. The IV is kept in a shift register. It is encrypted to produce a corresponding 64-bit IV cipher text.

ii) Now , the leftmost j bits of the encrypted IV are XORed with the first j bits of the plain text. This produces the first portion of cipher text say C.

iii) Now, the bits of IV are shifted left by j positions. Thus the j positions of the shift regieter now contain unpredictable data. These rightmost j position are now filled with C.

iv) Now steps i) through iii) continue until all the plain text units are encrypted .

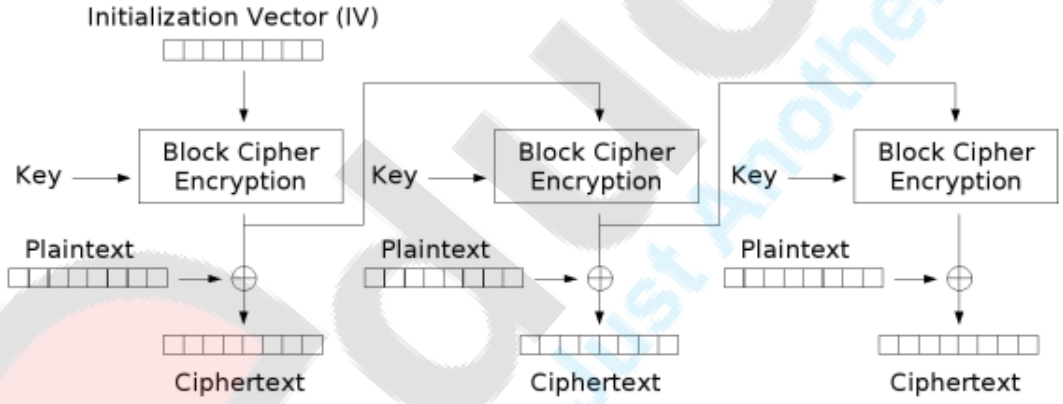


Cipher Feedback (CFB) mode decryption

4.Output Feedback (OFB) Mode

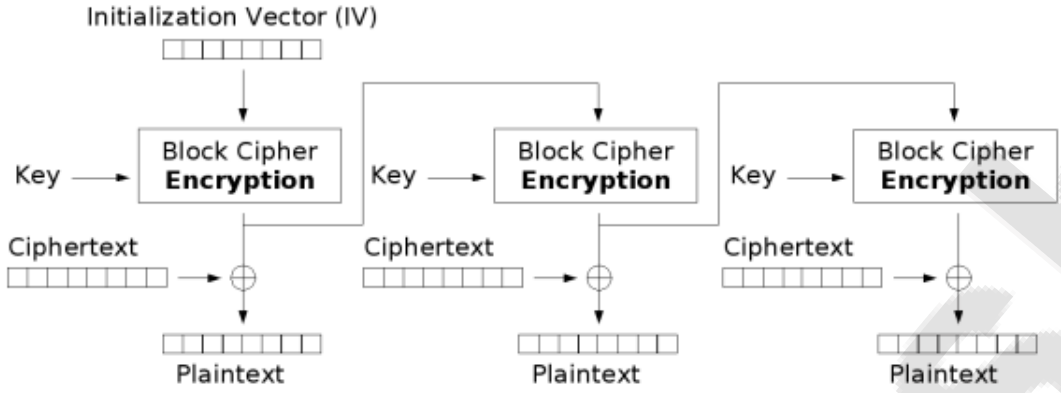
The only difference is that in case of CFB, the cipher text is fed into the next stage of encryption process. But in case of OFB , the output of the cipher text is fed into the next stage of encryption process.

We can state that in this mode if there are errors in individual bits, they remain errors on individual bits and do not corrupt the whole message. That is ,bit errors do not get propagated.



Output Feedback (OFB) mode encryption

If a cipher text bit Ci is in error, only the decrypted value corresponding to these bits, i.e. Pi is wrong. Other bits are not affected.

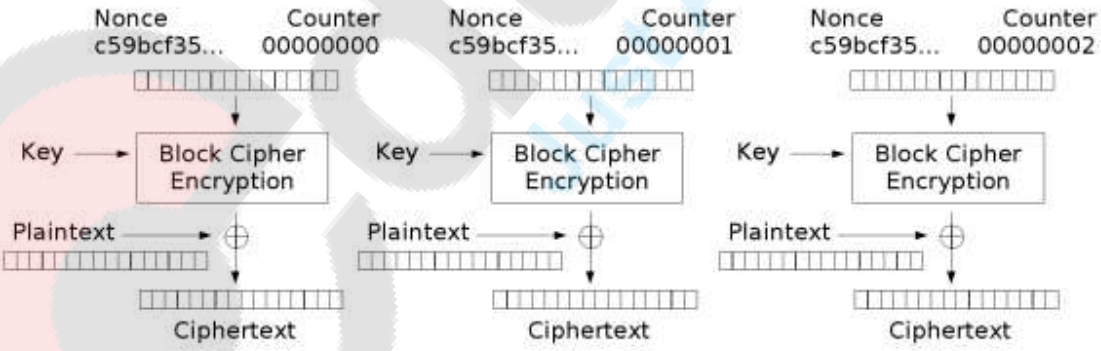


Output Feedback (OFB) mode decryption

The possible drawback with the OFB is that an attacker can make necessary changes to the cipher text and the checksum o the message in a controlled fashion, these causes change the cipher text without it getting detected. In other words, the attacker changes both the cipher text and the checksum at the same time, hence is no way to detect this change.

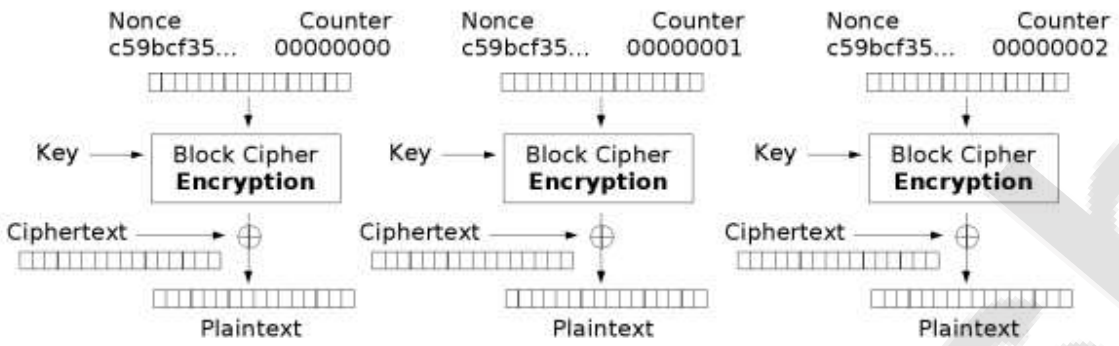
5.Counter (CTR) Mode

It uses sequence numbers called ad\s counter as the inputs to the algorithm. After each block is encrypted, to fill the register, the next counter value is used. Usually , a constant is used as the initial counter value and is incremented for every iteration. The size of the counter block is same as that of the plain text block.



Counter (CTR) mode encryption

CTR mode has similar characteristics to OFB, but also allows a random access property during decryption. CTR mode is well suited to operation on a multi-processor machine where blocks can be encrypted in parallel. The IV/nonce and the counter can be concatenated, added, or XORed together to produce the actual unique counter block for encryption.



Counter (CTR) mode decryption

Q7) Give a overview of DES. Explain DES round. What are weak and semi-weak keys used in DES.
OR
Explain following steps in DES algorithm.

1. **Basic principle**
2. **Initial Permutation**
3. **Rounds**

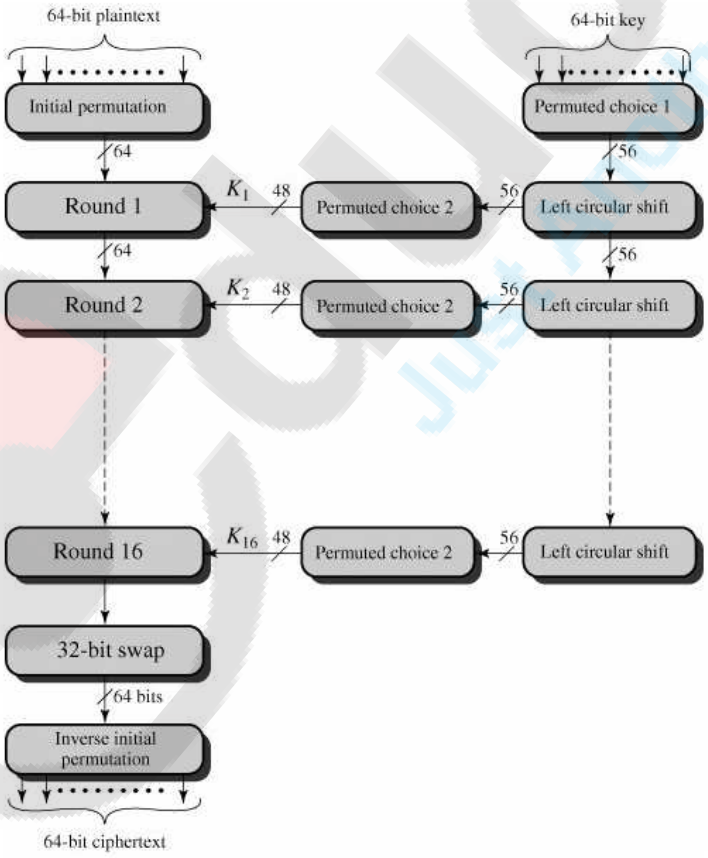
Ans:

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). The algorithm itself is referred to as the Data Encryption Algorithm (DEA).[6] For DES, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

DES Encryption

The overall scheme for DES encryption is illustrated in Figure 3.4. As with any encryption scheme, there are two inputs to the encryption function: the plaintext to be encrypted and the key. In this case, the plaintext must be 64 bits in length and the key is 56 bits in length.

Figure 3.4. General Depiction of DES Encryption Algorithm



Looking at the left-hand side of the figure, we can see that the processing of the plaintext proceeds in three phases. First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the permuted input. This is followed by a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce the pre-output. Finally, the pre-output is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function, to produce the 64-bit cipher text.

The right-hand portion of Figure 3.4 shows the way in which the 56-bit key is used. Initially, the key is passed through a permutation function. Then, for each of the 16 rounds, a *sub-key* (K_i) is produced by the combination of a left circular shift and a permutation. The permutation function is the same for each round, but a different sub-key is produced because of the repeated shifts of the key bits.

Initial Permutation

The initial permutation and its inverse are defined by tables. The tables are to be interpreted as follows. The input to a table consists of 64 bits numbered from 1 to 64. The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the position of a numbered input bit in the output, which also consists of 64 bits.

Details of Single Round

Figure 3.5 shows the internal structure of a single round. Again, begin by focusing on the left-hand side of the diagram. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right). As in any cipher, the overall processing at each round can be summarized in the following formulas:

$L_i = R_{i-1}$

$R_i = L_{i-1} \times F(R_{i-1}, K_i)$

The round key K_i is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the R bits. The resulting 48 bits are XORed with K_i. This 48-bit result passes through a substitution function that produces a 32-bit output, which is then permuted.

The role of the S-boxes in the function F is illustrated in Figure 3.6. The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output. These transformations are interpreted as follows: The first and last bits of the input to box S_i form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S_i. The middle four bits select one of the sixteen columns. The decimal value in the cell selected by the row and column is then converted to its 4-bit representation to produce the output. For example, in S_1 for input 011001, the row is 01 (row 1) and the column is 1100 (column 12). The value in row 1, column 12 is 9, so the output is 1001.

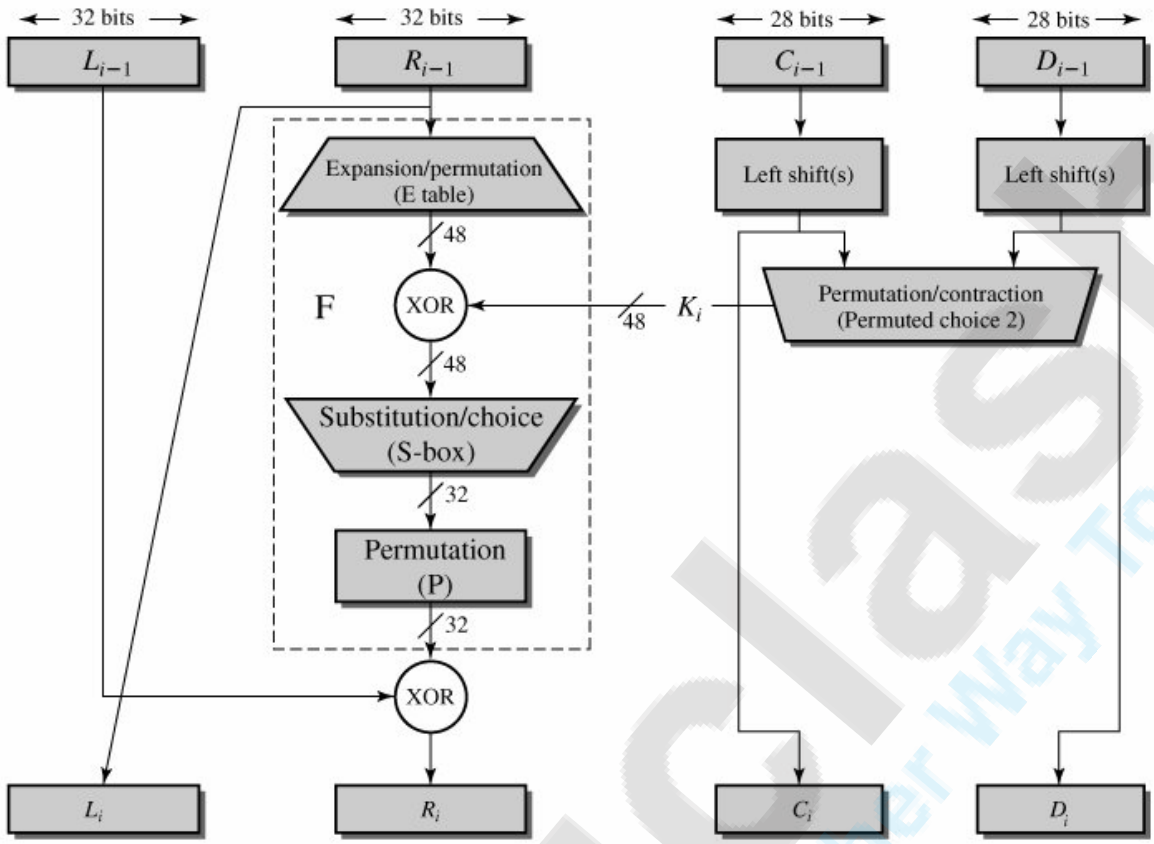
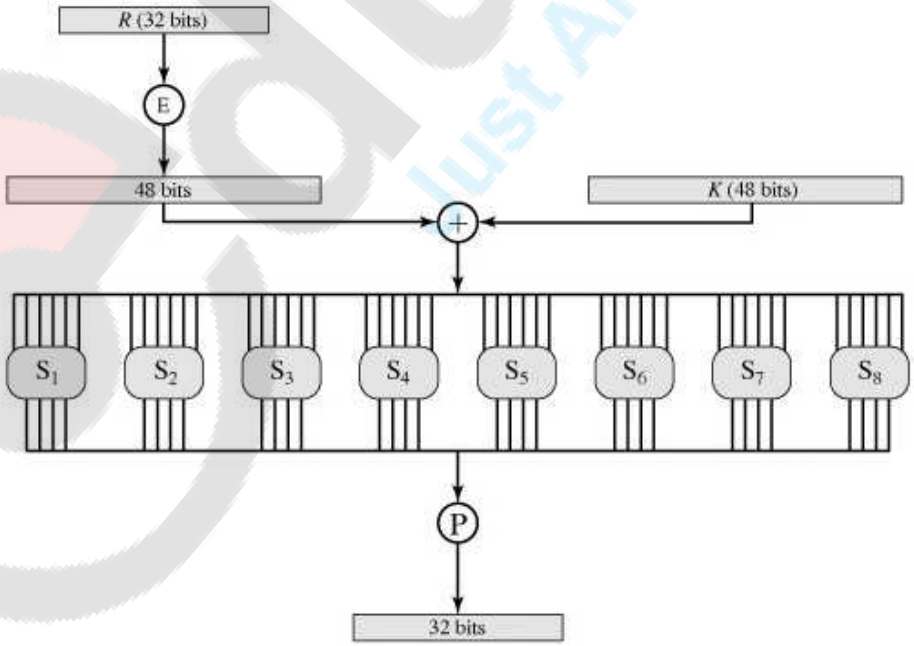


Figure 3.6. Calculation of F(R, K)

(This item is displayed on page 78 in the print version)



Weak and Semi-weak Keys

There are 16 DES keys that the security community warns people against using, because they have starnge properties. But the probability of randomly generating one of these keys is only $16/2^{56}$, which in our opinion is nothing to worry about. Its probably equally insecure to use a key with a value less than 1000, since an attacker might be likely to start searching for a keys from the bottom.

Generating the per-round keys that the key is subjected to an initial permutation to generate two 28-bit quantities, C_0 and D_0 . The 16 suspect keys are ones for which C_0 and D_0 are one of the four values: All ones, all zeros, alternatives ones and zeros, alternating zeros and ones. Since there are 4 possible values for each half, there are 16 possibilities in all. The 4 weak keys are the ones for which each of C_0 and D_0 are all ones or all zeros. Weak keys are their own inverses. The remaining 12 keys are the semi-weak keys. Each is the inverse of one of the others.

Q 8) Differentiate between:

- **DES & IDEA**
- **ECB Mode & CBC mode of DES**

Ans:

DES	IDEA
1. DES is efficient to implement in h/w but relatively slow if implemented in s/w. 2.56 bit key is used to generate 16, 48 bit round key. 3. Same key are used in reverse order deriving decryption. 4. Each DES S-box maps a 6-bits qty into a 4-bits qty. 5. DES has 16 round. 6.56 bits key is expanded to generate 16 keys.	1. IDEA was designed to compute in s/w. 2. Uses 128 bit key. 3. Decryption odd round mathematical inverses of keys is used. Even round same key used while encrypt are used No inverse of key. 4. Each Primitive maps 16-bits qty into a 16-bits qty. 5. Has 17-round odd no round are diff from even numbered round. 6.128-bit key is expanded into 52 keys.

ECB	CBC
1. Each block is encrypted to cipher text block. 2. Does not use IV. 3. Error is transmitted cipher text. 4.Repeated plain text, cipher text repeated 5. Less Expensive.	1. Each block is encrypted to cipher text block. Each cipher text block used to modify the encryption of next block. 2. Uses IV. 3. Error propagates in both cipher and plain text msg. 4. Different Cipher text block are three. 5. Overhead of transmitting and generating IV.

Q. 9) Explain the symmetric key encryption algorithm IDEA.

Ans. The International Data Encryption Algorithm (IDEA) is one of the strongest cryptographic algorithms. It was launched in 1990. Technically, IDEA is a block cipher. It works on 64-bit plain text blocks. The key size is 128 bits. IDEA is a symmetric key algorithm, that is, the same algorithm is used for encryption as well as decryption.

Working of IDEA:

The 64-bit input plain text block is divided into four portions of plain text, say P1 to P4. Thus, P1 to P4 are the inputs to the first round. There are eight such rounds. For the first round, we will have keys from Z1 to Z6. For the second round, we will have keys from Z7 to Z12. Finally, for the eighth round, the keys would be from Z43 to Z48. The final step consists of an Output Transformation, which has four sub-keys. The final output is produced by the Output Transformation step, which is four blocks of cipher text which are combined to form the final 64-bit cipher text block (C1 to C4).

Rounds:

Each round out of the eight rounds is performed as per the steps given below:

1. Multiply P1 and Z1
2. Add P2 and Z2
3. Add P3 and Z3
4. Multiply P4 and Z4
5. XOR the results of step 1 and step 3
6. XOR the results of step 2 and step 4
7. Multiply the results of step 5 with Z5
8. Add the results of step 6 and step 7
9. Multiply the results of step 6 with Z6
10. Add the results of step 7 and step 9
11. XOR the results of step 1 and step 9
12. XOR the results of step 3 and step 9
13. XOR the results of step 2 and step 10
14. XOR the results of step 4 and step 10

Sub-key Generation for a Round:

First round- The initial key consists of 128 bits from which 6 sub-keys Z1 to Z6 are generated for the first round. Since these keys consist of 16 bits each, the first 96 bits are used for the first round.

Second round- In the second round, the 32 unused bits (97-128) are used. For each round, 96 bits are required. Hence, to acquire the next 64 bits, the technique of key shifting is used.

Output Transformation:

This is a one-time operation. It takes place at the end of the eighth round. Output of eighth round is the input to this round. The 64-bit value is divided into four sub-blocks (say R1 to R4, each consisting of 16 bits).

The steps in Output Transformation are as follows:

1. Multiply R1 and Z1
2. Add R2 and Z2
3. Add R3 and Z3
4. Multiply R4 and Z4

We can show the entire working of IDEA as given below:

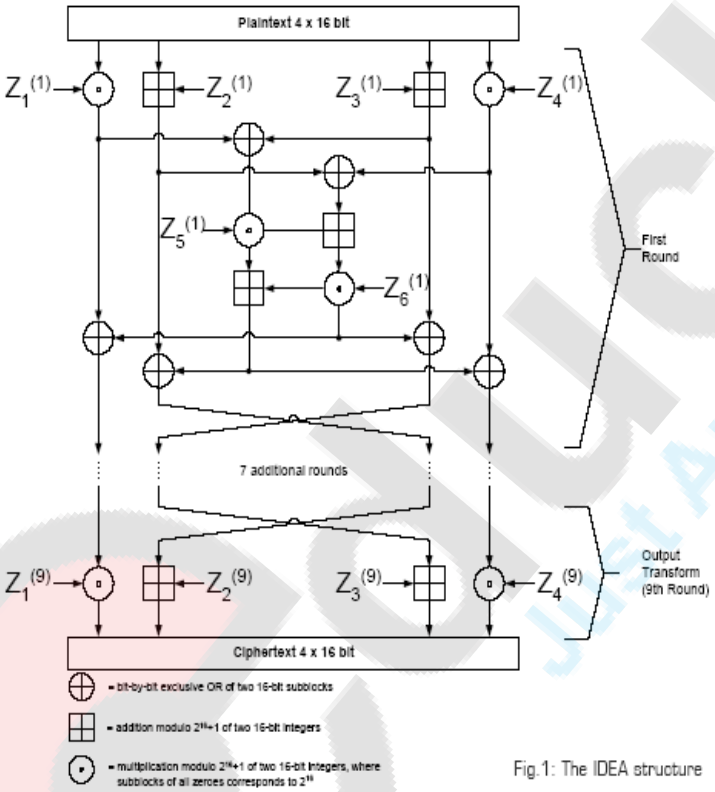


Fig.1: The IDEA structure

The Strength of IDEA:

IDEA uses a 128-bit key which is double the key size of DES. Thus, to break into IDEA, 2^{128} operations are required. Hence, more than 54×10^{23} years are required to break IDEA!

Q 10) What is the importance of message digest ? Explain MD2 and MD4 message digest scheme?

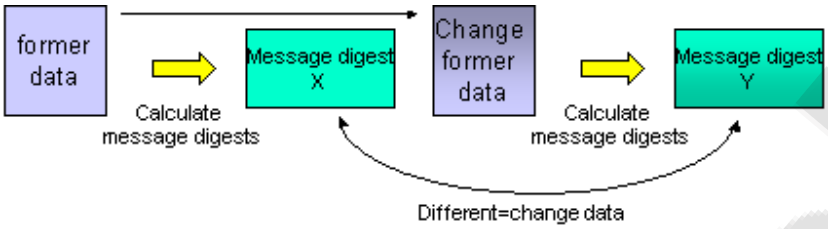
Ans:

1) A message digest is a compact digital signature for an arbitrarily long stream of binary data
2) Never generate the same signature for two different sets of input
3) Achieving such theoretical perfection would require a message digest as long as the input file
4) Compromise in favor of a digital signature of modest and usually fixed size created with an algorithm designed to make preparation of input text with a given signature computationally infeasible
5) Chances of two message digests being the same for two different inputs extremely remote
6) Message digest algorithms have much in common with techniques used in encryption

Importance of message digest:

1) Message digest functions also called *hash functions* , are used to produce digital summaries of information called message digests. Message digests (also called *hashes*) are commonly 128 bits to 160 bits in length and provide a digital identifier for each digital file or document. Message digest functions are mathematical functions that process information to produce a different message digest for each unique document. Identical documents have the same message digest; but if even one of the bits for the document changes, the message digest changes.
2) Because message digests are much shorter than the data from which the digests are generated and the digests have a finite length, duplicate message digests called *collisions* can exist for different data sets. However, good message digest functions use one-way functions to ensure that it is mathematically and computationally infeasible to reverse the message digest process and discover the original data. Finding collisions for good message digest functions is also mathematically and computationally infeasible but possible given enough time and computational effort. However, even if an attacker discovers a collision, it is highly improbable that the collision could be useful.
3) Message digests are commonly used in conjunction with public key technology to create digital signatures or "digital thumbprints" that are used for authentication, integrity, and nonrepudiation. Message digests also are commonly used with digital signing technology to provide data integrity for electronic files and documents
4) The generation of a digest is very fast and the digest itself is very small and can easily be encrypted and transmitted over the internet
5) It is very easy and fast (and therefore cheap) to check some data for validity

6) The algorithms are well known and implemented in most major programming languages, so they can be used in almost all environments

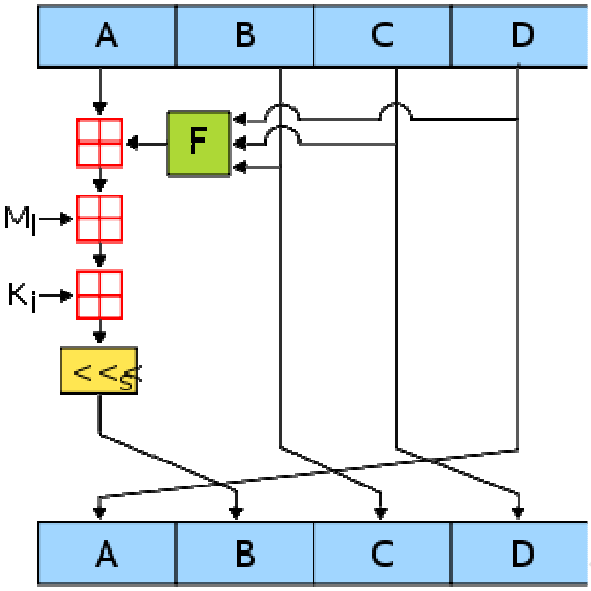


MD2:

1) **Message Digest Algorithm 2 (MD2)** is a cryptographic hash function developed by Ronald Rivest in 1989. The algorithm is optimized for 8-bit computers. MD2 is specified in RFC 1319. Although other algorithms have been proposed since, such as MD4, MD5 and SHA, even as of 2010 MD2 remains in use in public key infrastructures as part of certificates generated with MD2 and RSA
2) The 128-bit hash value of any message is formed by padding it to a multiple of the block length on the computer (128 bits or 16 bytes) and adding a 16-byte checksum to it. For the actual calculation, a 48-byte auxiliary block and a 256-byte S-table generated indirectly from the digits of the fractional part of pi are used (see nothing up my sleeve number). The algorithm runs through a loop where it permutes each byte in the auxiliary block 18 times for every 16 input bytes processed.
3) Once all of the blocks of the (lengthened) message have been processed, the first partial block of the auxiliary block becomes the hash value of the message.

MD4:

1) MD4 (Message-Digest algorithm 4) is a message digest algorithm (the fourth in a series) designed by Professor Ronald Rivest of MIT in 1990. It implements a cryptographic hash function for use in message integrity checks. The digest length is 128 bits. The algorithm has influenced later designs, such as the MD5, SHA-1 and RIPEMD algorithms.
2) The 128-bit (16-byte) MD4 hashes (also termed *message digests*) are typically represented as 32-digit hexadecimal numbers.



3) One MD4 operation : MD4 consists of 48 of these operations, grouped in three rounds of 16 operations. *F* is a nonlinear function; one function is used in each round. M_i denotes a 32-bit block of the message input, and K_i denotes a 32-bit constant, different for each operation.

Q.11) What is the minimal & Maximum amount of padding that would be required in each message digest function MD2, MD4, MD5 , SHA, SHA-1?

Ans:

Padding for MD2:

The message is "padded"(extended) so that its length is congruent to 0, modulo 16. That is, the message is extended so that it is multiple of 16 byte long. Padding is always performed, even if the length of the message is already Congruent to 0, modulo 16. Padding is performed as follows : a single "i" bit is appended to the message so that the length is in byte of the padded message become congruent to 0,modulo 16. At this point the resulting message has length what it is exact multiple of 16 byte.

Let M[0…….N-1] denotes the bytes of the resulting message. Where N is a multiple of 16.

Padding For MD4:

The message is "padded"(extended) so that its length is congruent to 448 modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512. Padding is performed as follows : a single "1" bit is appended to the message, and then "0" bit appended so that the length is in bit of the padded message become congruent to 448,modulo 512. In all of at least one bit and at most 512 bits are appended.

Padding For MD5:

The message is "padded"(extended) so that its length is similar to 448 modulo 512.that is, the message is extended so that it is just 64 bits timid of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is always similar to 448,modulo 512. Padding is performed as follows : a single "1" bit is appended to the message, and then "0" bit appended so that the length is in bit of the padded message become congruent to 448,modulo 512. In all at least one bit and at most 512 bits are appended.

Padding For SHA:

Like MD5, the first step in SHA is to add padding to the end of original message in such a way that the length of the message is 64 bits short of a multiple of 512. Like MD5, the padding is always added, even if the message is already 64 bits short of a multiple of 512.

Padding For SHA-1:

SHA-1 pads message in same manner as MD4 and MD5, except that SHA-1 is not different for a message that is longer than 2^64 bits.If there were such a message it would take several hundred years to transmit it at 10 Gigabits per second, and it would take even longer to compute the SHA-1 message digest for it at 1000 MBPS.

Q 12) What is a message Digest? How is a message digest calculated using MD5 algorithm?

Ans: The electronic equivalent of the document & fingerprint pair is the message & message digest pair it is fingerprint or the summary of a message. It is similar to the concepts of Longitudinal Redundancy Check (LRC) or Cyclic Redundancy Check (CRC).

To preserve the integrity (i.e. to ensure that a message has not been tampered with after it leaves the sender but before it reaches the receiver) of a message, the message is passed through a message digest algorithm also called as hash function.

Thus we perform a hashing function (or message digest algorithm) over block of data to produce its hash or message digest, which is smaller in size than the original message. This concept is shown in figure.

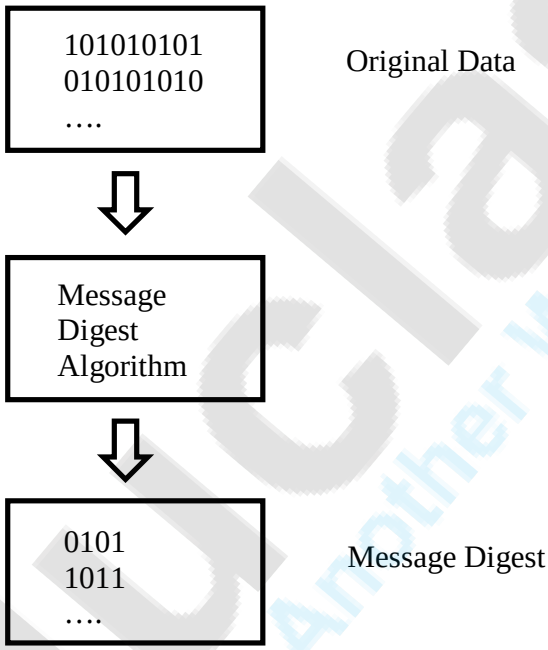


Figure: Message Digest Concept.

MD5 :

- MD5 is a message digest algorithm developed by Ron Rivest.
- MD5 actually has its roots in a series of message digest algorithm, which were predecessors of MD5, all developed by Ron Rivest.
- The original message digest algorithm was called as MD. He soon came up with next version MD2, MD3 & MD4 but those were quite weak, failure etc. consequently Ron Rivest released MD5.
- MD5 is quite fast & produces 128 bit message digest. MD5 has been able to successfully defend itself against collisions.

Working of MD5 :

Step 1: Padding The first step in MD5 is to add padding bits to the original message. The aim of this step is to make the length of the original message equal to a value, which is 64 bits less than an exact multiple of 512.

Thus, after padding, the original message will have a length of 448 bits (64 bits less than 512), 960 bits (64 bits less than 1024), 1472 bits (64 bits less than 1536), etc.

The padding consists of a single 1-bit, followed by as many 0-bits, as required. Note that padding is always added, even if the message length is already 64 bits less than a multiple of 512. Thus, if the message were already of length say 448 bits, we will add a padding of 512 bits to make its length 960 bits. Thus, the padding length is any value between 1 and 512.

The padding process is shown in below:

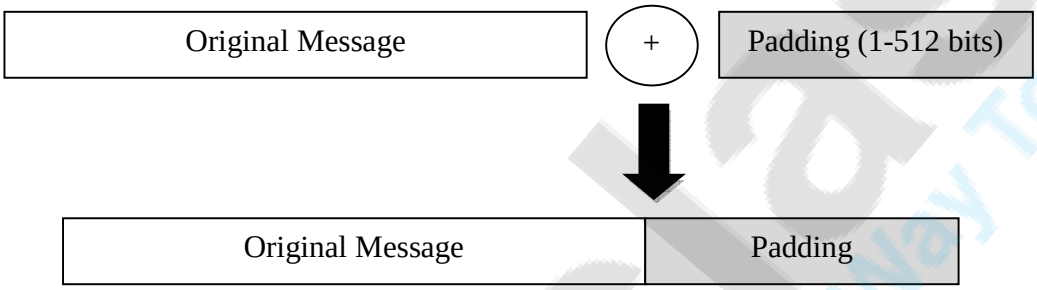


Figure: Padding Process

Step 2: Append length After padding bits are added, the next step is to calculate the original length of the message and add it to the end of message, after padding.

The length of the message is calculated, excluding the padding bits (i.e. it is the length before the padding bits were added). For instance, if the original message consisted of 1000 bits and we added a padding of 472 bits to make the length of the message 64 bits less than 1536 (a multiple of 512), the length is considered as 1000 and not 1472 for the purpose of this step.

This length of the original message is now expressed as a 64-bit value and these 64 bits are appended to the end of the original message + padding. This is shown in Figure below. Note that if the length of the message exceeds 2^{64} bits (i.e. 64 bits are not enough to represent the length, which is possible in the case of a really long message), we use only the low-order 64 bits of the length. That is, in effect, we calculate the length mod 2^{64} in that case.

We will realize that the length of the message is now an exact multiple of 512. This now becomes the message whose digest will be calculated.

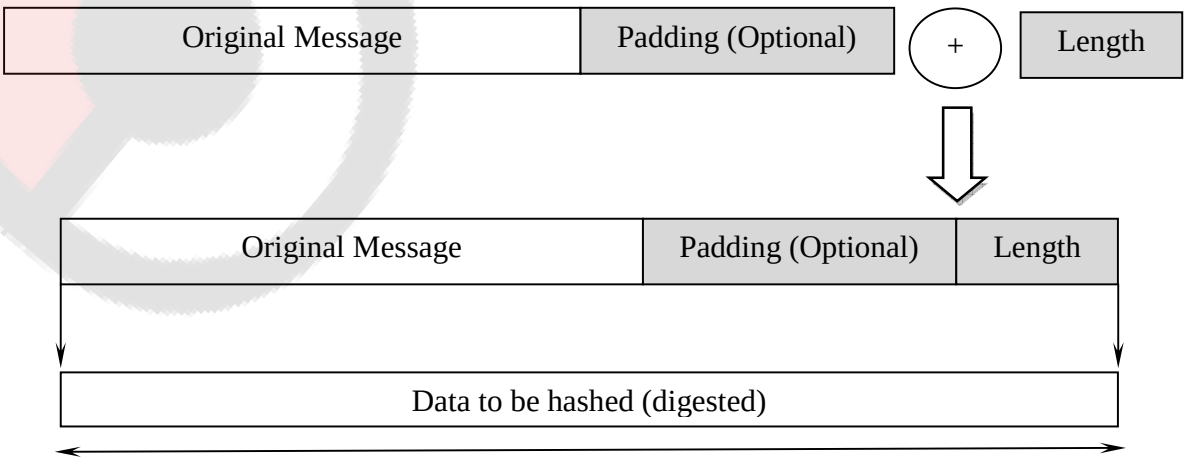


Figure: Append Length

Step 3: Divide the input into 512-bit blocks Now, we divide the input message into blocks, each of length 512 bits. This shown below:

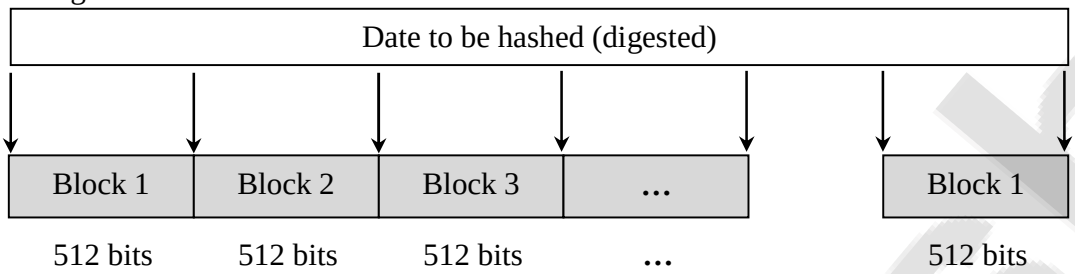


Figure: Data is divided into 512-bits blocks

Step 4: Initialize chaining variables In this step, four variables (called as **chaining variables)** are initialize. They are called as A, B, C and D. Each of these is a 32-bit number. The initial hexadecimal values of these chaining variables are shown below:

A	Hex	01	23	45	67
B	Hex	89	AB	CD	EF
C	Hex	FE	DC	BA	98
D	Hex	76	54	32	10

Figure: Chaining Variables

Step 5: Process blocks After all the initializations, the real algorithm begins. It is quite complicated and we shall discuss it step-by-step to simplify it to the maximum extent possible.

There is a loop that runs for as many 512-bit block as are in the message.

Step 5.1: Copy the four chaining variables into four corresponding variables, a, b, c and d (note the smaller case). Thus, we now have a = A, b = B, c = C and d = D. This is shown below:

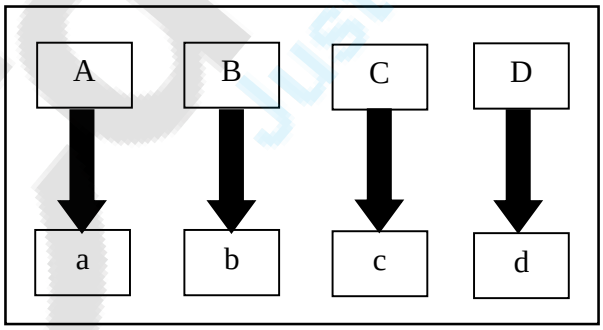


Figure: Copying chaining variables into temporary variables

Actually, the algorithm considers the combination of a, b, c and d as a 128-bit single register (which we shall call as abcd). This register (abcd) is useful in the actual algorithm operation for holding intermediate as well as final results. This is shown below:

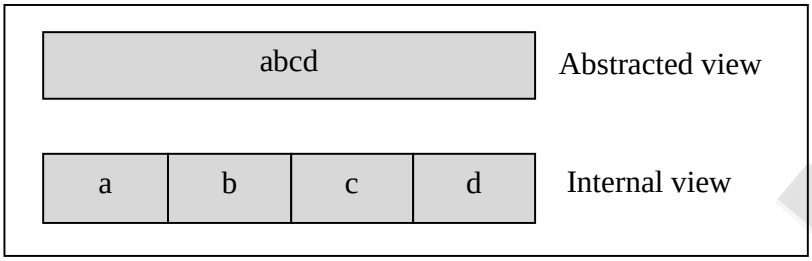


Figure: Abstracted view of the chaining variables

Step 5.2: Divide the current 512-bit block into 16 sub-blocks. Thus, each sub-block contains 32 bits, as shown below:

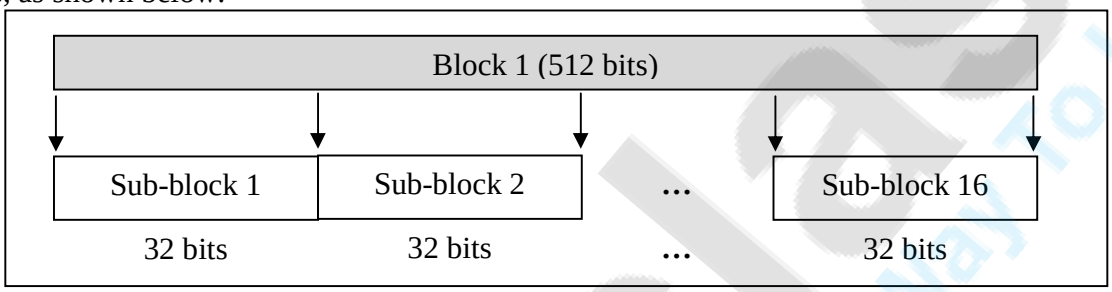


Figure: Sub-blocks within a block

Step 5.3: Now, we have four rounds. In each round, we process all the 16 sub-blocks belonging to a block. The inputs to each round are: (a) all the 16 sub-blocks, (b) the variables a, b, c, d and (c) some constants, designated as t. This is shown as below:

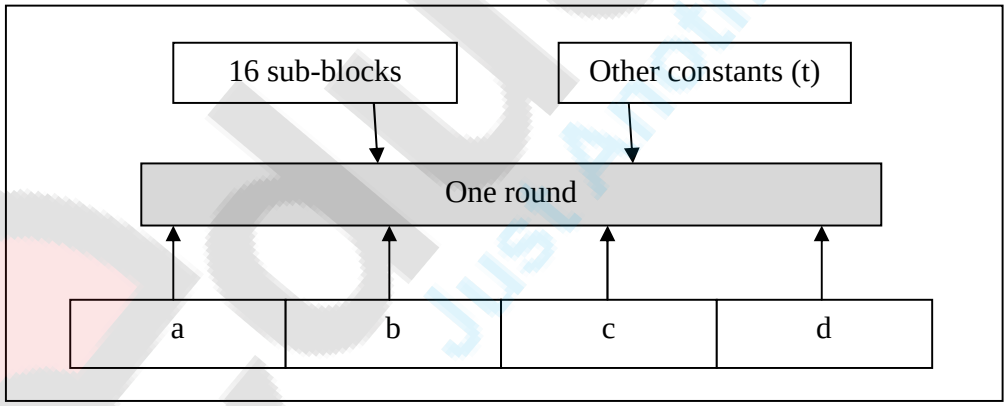


Figure: Conceptual process within a round

All the four rounds vary in one major way: Step 1 of the four round has different processing. The other steps in all the four rounds are the same.

- In each round, we have 16 input sub-blocks, named M[0], M[1], …,M[15] or in general, M[i], where I varies from 0 to 15. As we know, each sub-block consists of 32 bits.
- Also, t is an array of constants. It contains 64 elements, with each element consisting of 32 bits.

We denote the elements of this array t as t[1], t[2],.., t[64] or in general as t[k], where k varies from 1 to 64. Since there are four rounds, we use 16 out of the 64 values of t in each round.

Let us summarize these iterations of all the four rounds. In each case, the output of the intermediate as well as the final iteration is copied into the register abcd. Note that we have 16 such iterations in each round.

1. A process P is first performed on b, c and d. This process P is different in all the four rounds.
2. The variable a is added to the output of the process P (i.e. to the register abcd).
3. The message sub-block M[i] is added to the output of Step 2 (i.e. to the register abcd).
4. The constant t[k] is added to the output of Step 3 (i.e. to the register abcd).
5. The output of Step 4 (i.e. the contents of register abcd) is circular-left shifted by s bits. (The value of s keeps changing, as we shall study).
6. The variable b is added to the output of Step 5 (i.e. to the register abcd).
7. The output of Step 6 becomes the new abcd for the next step

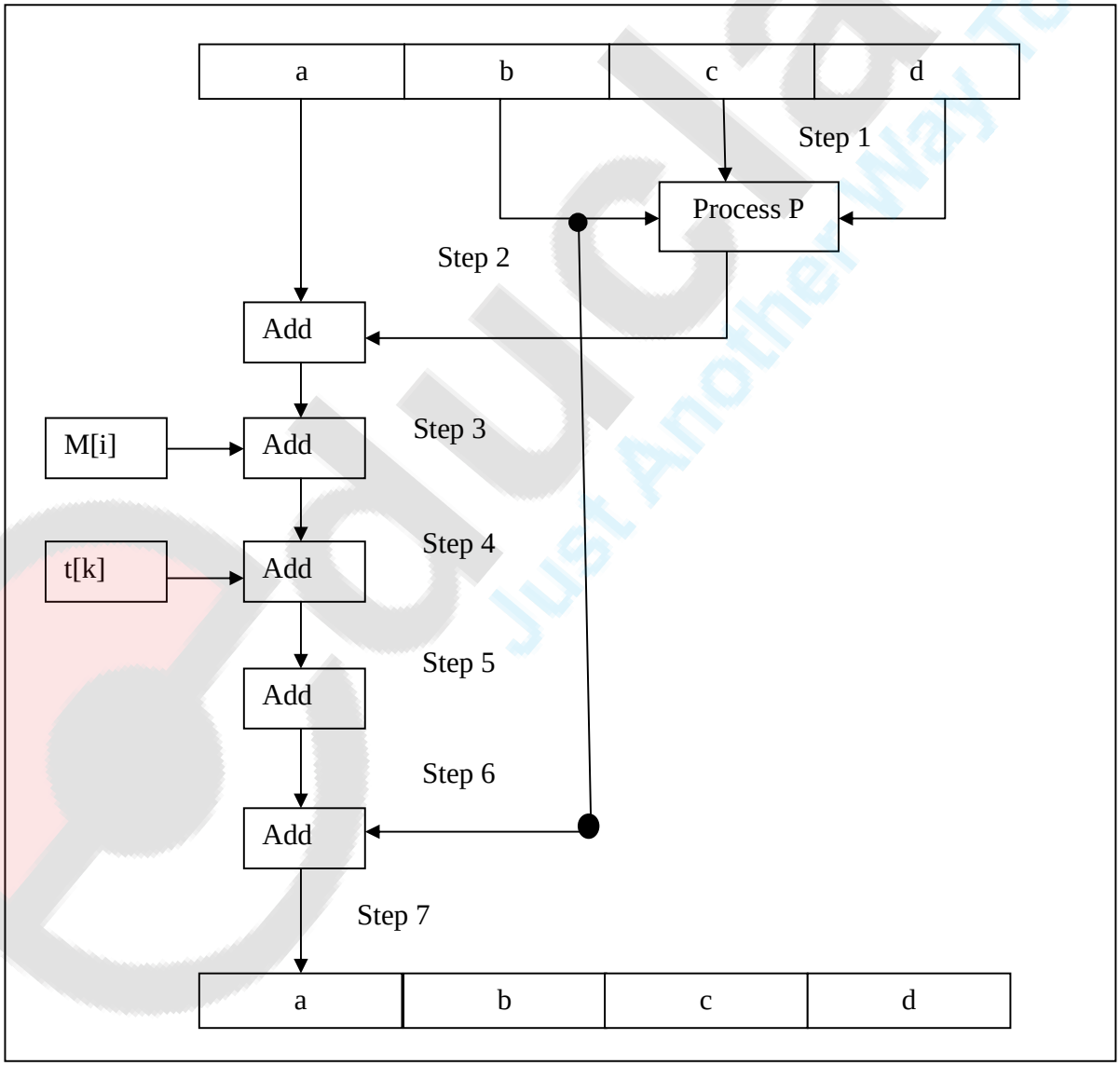


Figure: One MD5 operation

We can mathematically express a single MD5 operation as follows:

a = b + ((a + Process P (b, c, d) + M[i] + T[k]) <<< s)

Where,

a, b, c, d	= Chaining variables, as described earlier
Process P	= A non-linear operation, as described subsequently
M[i]	= M[q x 16 + i], which is the ith 32-bit word in the qth 512-bit block of the message
t[k]	= A constant, as discussed subsequently
<<< s	= Circular-left shift by s bits

Understanding the process P

The process P is nothing but some basic Boolean operations on b, c and d. This is shown below in the table.

Note that in the four rounds, only the process P differs. All the other steps remain the same. Thus, we can substitute the actual details of process P in each of the round and keep everything else constant.

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

Table: Process P in each round

Q13) What is Hash? Discuss briefly SHA-1 and compare it with predecessor MD5?

Ans:

Hash

- A Hash is a transformation of data into distilled forms that are unique to the data and is a one way function.
- In Hashing, a fixed length message digest is created out of a variable length message. A digest is normally much smaller than message.
- A hash function accept a variable size message M as input and produce fixed size output, referred to as Hash Code H(M).
- Hash code is a function of all the bits of message and provides an error detection capability: A change to any bit or bits in message result in a change to the hash code.

SHA-1

- SHA-1(Secure hash algorithm) was proposed by NIST as a message digests function.
- SHA-1 takes a message of length at most 2^{64} bits and produces a 160 bit output.
- It is similar to the MD5 message digest function, but it is a little slower to execute and presumably more secure.
- MD5 made four passes over each block of data; SHA-1 makes five. It also produce a 160 bit digest as opposed o the 128 of the MDs.

SHA-1 Message Padding

SHA-1 pads message in same manner as MD4 and MD5, except that SHA-1 is not different for a message that is longer than 2^{64} bits. If there were such a message it would take several hundred years to transmit it at 10 Gigabits per second, and it would take even longer to compute the SHA-1 message digest for it at 1000 MBPS.

Overview of SHA-1 Message Digest

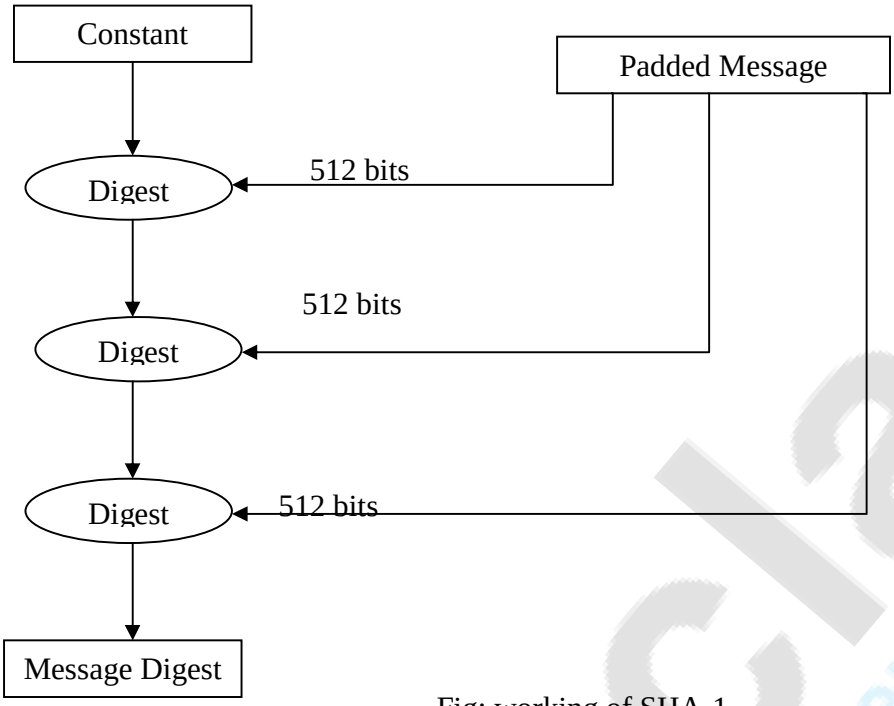


Fig: working of SHA-1

Like MD5, SHA-1 operates in stages. Each Stage mangles the pre-stage message digest by a sequence of operations based on the current message block.

At the end of the stage, each word of the mangled message digest is added to its pre-stage value to produce the post-stage value.

Therefore, the current value of the message digest must be saves at the beginning of the stage so that it can be added in the end of the stage.

The 160 bit message digest consist of 5 32-bit words. Consider these words are A,B,C,D and E.

Before first stage the values are A=67452301_{16} , B=$efcdab89_{16}$, C= $98badcfe_{16}$,

D= 10325476_{16}, E=$c3d2e1f0_{16}$.After last stage, the value of A|B|C|D|E is the message digest for entire message.

SHA-1 Operation on a 512-bit Block

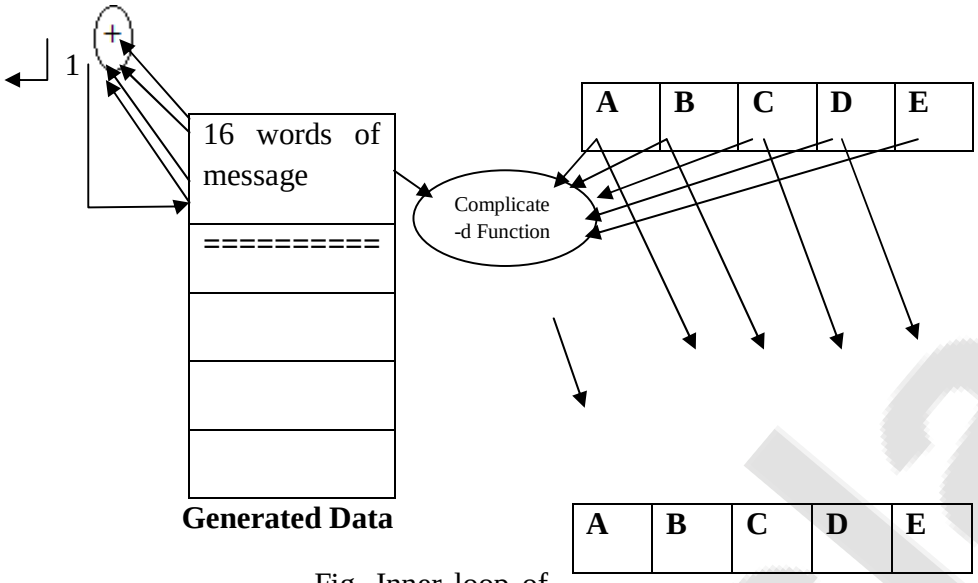


Fig. Inner loop of SHA-1 80 iteration per block

At the start of each stage, the 512-bit message block is used to create a 5×512-bit chunk as shown in above fig.

The first 512-bits of the chunk consist of the message block. The rest get filled in, a 32-bit word at a time, according to the bizarre rule that the n^{th} word is the XOR of words n-3, n-8, n-14 and

n-16. In SHA-1 XOR of words n-3, n-8, n-14 and n-16 is rotated left one bit before stored as word n; this is only modification from the original SHA.

We have a buffer of eighty 32-bit words i.e. W_0 to W_{79}.

For t=0 through 79,modify A,B,C,D and E as follows:

B=old A, C= old B$\leftarrow$30, D= old C , E= old D

Therefore new A depends on old A , B, C, D, and E.

A= E + (A $\leftarrow$5) + W_t+K_t+ f(t,B,C,D)

SHA-1 Features:

- The SHA1 is used to compute a message digest for a message or data file that is provided as input.
- The message or data file should be considered to be a bit string.
- The length of the message is the number of bits in the message (the empty message has length 0).
- If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex.
- The purpose of message padding is to make the total length of a padded message a multiple of 512.
- The SHA1 sequentially processes blocks of 512 bits when computing the message digest.
- The 64-bit integer is l, the length of the original message.
- The padded message is then processed by the SHA1 as n 512-bit blocks.

Comparison of MD5 and SHA-1

Points	MD5	SHA-1
Message Digest Length in bits	120	160
Attack to try and find the original message digest	Requires 2^{128} operations to break in	Requires 2^{160} operations to break in, therefore more secure
Attack to try and find two message producing the same message digest	Requires 2^{64} operations to break in	Requires 2^{80} operations to break in
Successful attack so far	There have been reported attempts to some extent	No such claim so far
Speed	Faster (64 iterations and 128 bit buffer)	Slower(80 iterations and 160 bit buffer)
Software Implementation	Simple, does not need any large programs or complex tables.	Simple, does not needs any large programs or complex tables.

Q14) Discuss various PKI trust models?
Ans:

Diagram of a public key infrastructure

Introduction

Public Key Infrastructure (**PKI**) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. n cryptography, a **PKI** is an arrangement that binds public keys with respective user identities by means of a certificate authority (**CA**). The user identity must be unique within each CA domain. The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. The PKI role that assures this binding is called the Registration Authority (**RA**). For each user, the user identity, the public key, their binding, validity conditions and other attributes are made unforgettable in public key certificates issued by the CA.

PKI trust models

Monopoly Model

- In this model, the world chooses one organization, universally trusted by all companies, countries, universities, and other organizations to be the single CA for the world.
- The key of that one organization is embedded in all software and hardware as the PKI trust anchor. Everyone must get certificate from it. This model is a wonderfully simple

model, mathematically. This is favored by organizations hoping to be monopolist. However, there are problems with it:

- There is no one universally trusted organization.
- Given that all software and hardware would come preconfigured with the monopoly organization's key, it would be infeasible to ever change that key in case it were compromised, since that would involve reconfiguration of every piece of equipment and software.
- It would be expensive and insecure to have a remote organization certify your key. How would they know it was you? How would you be able to securely send them your public key? Although transmission of the public key does not require secrecy, it requires integrity. Otherwise the CA could be tricked into certifying the public key as yours.
- Once enough software and hardware was deployed so that it would be difficult for the world to switch organizations, the organization would have monopoly control, and could charge whatever it wanted for granted certificates.
- The entire security of the world rest on that one organization never having an incompetent or corrupt employee who might be bribed or tricked into issuing bogus certificates or divulging the CA's private key.

Monopoly plus Registration Authorities (RAs)

- This model is just like Monopoly model except that the single CA chooses other organization (Know as RAs) to securely check identities and obtains and vouch for public keys. The RA then securely communicates with the CA, perhaps by sending signed email with the information that would go into the certificate, and the CA can than issue a certificate because it trusts the RA.
- This model's advantage over the Monopoly model is that it is more convenient and secure to obtain certificates, since there are more places to go to get certified. However, all the other disadvantages of the monopoly model apply.

Delegated CAs

- In this model the trust anchor CA can issue certificates to other CAs, vouching for their keys and vouching for their trustworthiness as CAs. Users can then obtain certificates from one of the delegated CAs instead of having to go to the trust anchor CA.
- The difference between a delegated CA and an RA is whether Alice sees a chain of certificates from a trusted anchor to Bob‘s name, or sees a single certificate. Assuming a monopoly trust anchor, this has security and operational properties similar to Monopoly plus Registration Authorities (RAs). Chains of certificates through delegated CAs can be incorporated into any of the models.

Oligarchy

- This is the model commonly used in browsers. In this model, instead of having products preconfigured with one single key, the products come configured with many trust anchors, and a certificate issued by any one of them is accepted. Usually in such a model it is possible for the user to examine and edit the list of trust anchors, adding or deleting trust anchors.
- It has the advantage over the monopoly models that the organizations chosen as trust anchors will be in competition with each other, so the world might be spared monopoly pricing.

Anarchy Model

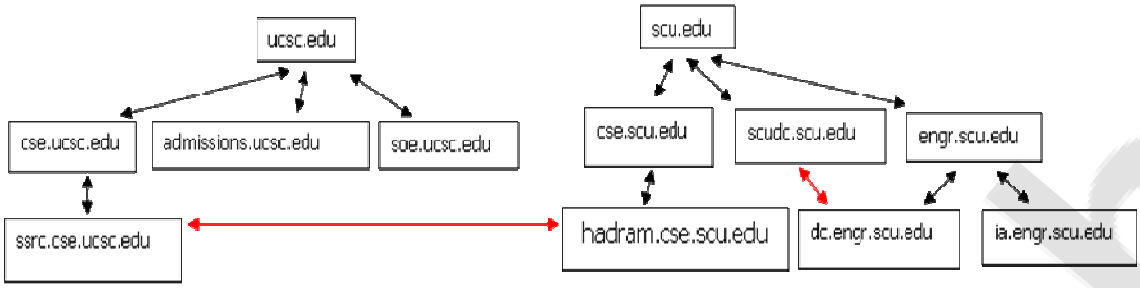
- This is the model used by PGP (pretty good privacy). Each user is responsible for configuring some trust anchors, for instance, public keys of people he has met and who have handed him a business card with PGP fingerprint (the message digest of the public key), and sent him email containing a public key with that digest. Then anyone can sign certificates for anyone else.
- Some organizations (for instance, MIT does this today) volunteer to keep a certificate database into which anyone can deposit certificates. To get key of someone whose key is not in your set of trust anchors, you can search through the public database to see if you can find a path from one of your trust anchors, you can search through the public database to see if you can find a path from one of your trust anchors to the name you want. This absolutely eliminates the monopoly pricing, but it is really unworkable on a large scale.

Top-Down with Name Constraints

- This model is similar to the monopoly model in that everyone must be configured with a preordained, never changing root key, and that root CA delegates to other CAs. However, the delegated CAs are only allowed to issue certificates for their portions of the namespace.
- In this model it is easy to find the path to a name (just follow the namespace from the root down). But it has the other problems of the monopoly model, in that everyone has to agree upon a root organization, and that organization and its key would be prohibitively expensive to ever replace

Bottom-Up with Name Constraints

- Each organization can create its own PKI and link it to others.
- Parent certifies child (downlink), but child certifies parent as well (uplink).
- Cross links are links from one node to another one.



Bottom-Up PKI model with two cross links (red)

- Even with a single cross-link from one tree into the other, we can establish a path from all nodes of the first tree to all of the other.
- Maintaining cross-links can be expensive.
- Root service providers can provide these cross-links (for a fee).

Q16) Diffie-Hellman neither encrypts nor signs. What does it do? What is man in middle attack? Distinguish what are different public key encryption algorithms? Discuss Diffie-Hellman crypto system.

Ans. Diffie-hellman neither encrypts nor signs.

There is a problem of key distribution or key exchange in a symmetric key cryptography in which one key is used for encryption and decryption of a plain text.

- Whitefield Diffie and Martin Hellman came up with solution to the problem of key agreement or key exchange in 1976.
- The beauty of this scheme is that the two parties, who want to communicate securely, can agree on asymmetric key using this technique.
- This key can be used for encryption and decryption of text.
- The Diffie-Hellman key exchange algorithm is used for only key agreement but not for actual encryption or decryption of messages.
- Once both parties agree upon the key to be used, they need to use other symmetric key encryption algorithm for actual encryption and decryption of messages.
- Thus Diffie-Hellman key exchange /agreement algorithm is used only for key agreement between two parties and not for encryption or signing of document.

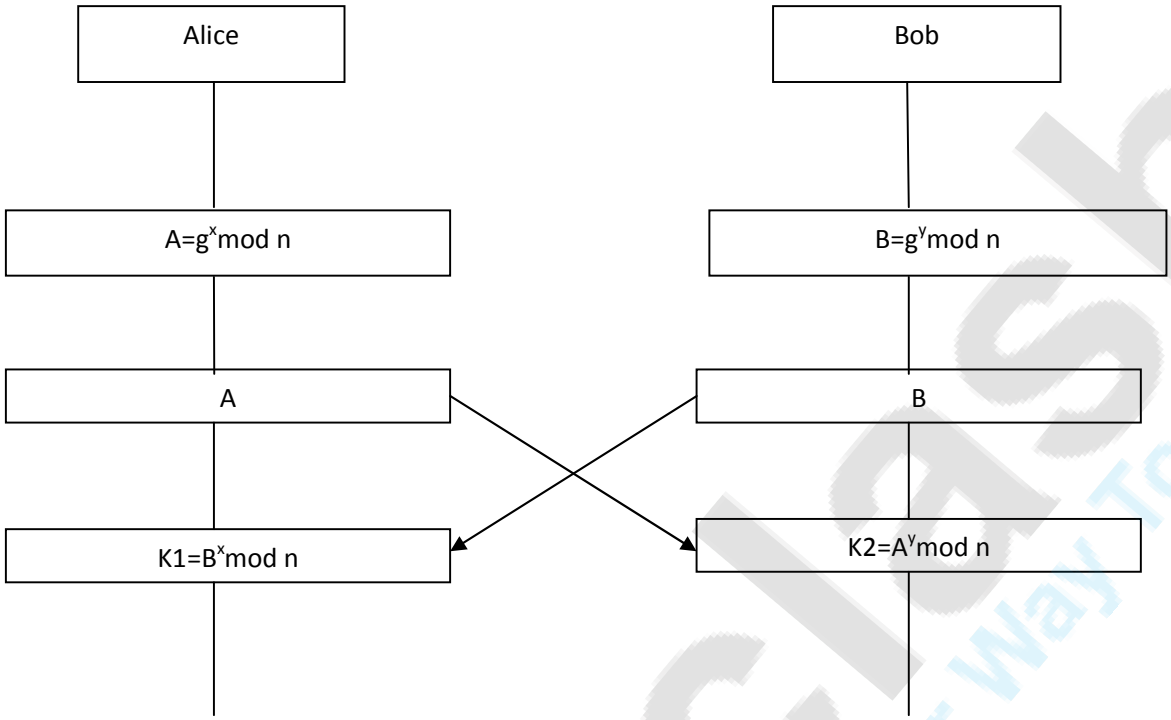
Diffie-hellman crypto system

The Diffie-Hellman key exchange algorithm is based on mathematical principles.

Description

Let us assume that Alice and Bob want to agree upon a key to be used for encrypting/decrypting messages that would be exchanged between them.

1. Alice and Bob agree upon two large prime numbers n and g. These two integers need not be kept secret. Alice and Bob can use an insecure channel to agree on them.
2. Alice chooses another large random number x, and calculate A such that:
   $A= g^x \bmod n$
3. Alice then sends the number A to Bob.
4. Bob independently chooses another large integer y and calculates B such that :
   $B=g^y \bmod n$
5. Bob then sends the number B to Alice.
6. Alice now computes the secret key k1 as follows:
   $K1=B^x \bmod n$
7. Bob now computes the secret key k2 as follows:
   $K2=A^y \bmod n$



K1 is equal to k2i.e k1=K2=k is the Symmetric key which Alice and Bob must keep secret and can use for encryption and decryption of their messages.

- Diffie-Hellman key exchange algorithm gets it security from the difficulty of calculating logarithms in a finite field as compared with the ease of calculating exponentiation in the same field.
- The idea behind this algorithm is quite simple.
- The final shared key is made up of three parts: g, x and y.
- The first part is a public member, known to all whereas the other two numbers must be available with Alice and Bob.
- Alice adds second part while Bob adds third.
- When Alice receives two third completed key from bob, she adds the remaining one-third part (i.e. x). This completes Alice keys.
- Similarly Bob gets the complete key.
- Although Alice key is made up of sequence g-x-y and Bob key is made up of sequence g-y-x, the two keys are same then also Alice cannot find Bobs part because the computation is done using modulus n. similarly Bob cannot derive Alice's part.

e.g.

1. Alice and Bob agree upon two large prim numbers

N=11,g=7

2. Alice chooses another large prime number x= 3 and calculates A:

$A=7^3 \bmod 11= 343 \bmod 11=2.$

3. Alice sends 2 to Bob.

4. Bob independently chooses another random number y=6 then calculate B:

$B=7^6 \bmod 11=117649 \bmod 11=4.$

5. Bob sends 4 to Alice,

6. Alice now computes key k1 :

$K1= 4^3 \bmod 11=64 \bmod 11=9.$

7. B now computes key K2:

$K2=2^6 \bmod 11=64 \bmod 11=9.$

8. Therefore

K1=K2=K.

<u>Man in the middle Attack</u>

In cryptography, the man-in-the-middle attack (often abbreviated MITM), bucket-brigade attack, or sometimes Janus attack, is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

Let us discuss man in the middle Attack using following e.g. of Diffie-Hellman Key exchange algorithm.

Diffie-Hellman key exchange algorithm can fall prey to the man-in-the middle attack.

- Alice wants to communicate with Bob securely and therefore uses the Diffie-hellman key exchange algorithm.
- For this purpose she chooses n=11 and g=7 and sends to Bob.

- Alice doesn't realize that the Attacker tom is listening to the conversation between Alice and Bob.
- Tom simply picks the values of n and g and also forwards them to Bob as they were originally.
- Now, Alice, tom and Bob select random number x and y
  Alice:- x=3
  Tom: - x=8, y=6
  Bob: - y=9
- Now, Alice calculates A Bob calculates b and Tom calculates A and B.
  Alice: - $A=g^x \bmod n=7^3 \bmod 11=2$.
  Bob: - $B=g^y \bmod n=7^9 \bmod 11=8$.
  Tom: - $A=g^x \bmod n=7^8 \bmod 11=9$.
  $B=g^y \bmod n=7^6 \bmod 11=4$.
- Alice sends her A to Bob, Tom intercepts it and sends his A to Bob. Bob has no idea that Tom has hijacked Alice's A.
- In return Bob sends his B to Alice, Tom again intercepts it and instead sends his b to Alice.
- Based on these values all three person calculates the key :
  Alice: - $k1=B^x \bmod n=4^3 \bmod 11=9$.
  Bob: - $K2=A^y \bmod n= 9^9 \bmod 11=5$.
  Tom: - $k1= B^x \bmod n=8^8 \bmod 11=5$.
  $K2= A^y \bmod n=2^6 \bmod 11=9$.
- Therefore Alice (k1) =Toms (K2) =k.
  Bob (K2) =Toms (k1) =k.

Tom needs two keys because at one side Tom wants to communicate with Alice securely using shared Symmetric key(9) and on other hand , he wants to communicate with Bob securely using different shared symmetric key(5). Only then can he receive messages from Alice view, manipulate them and forward them to Bob and vice-versa.

Alice and Bob believe that they are directly communicating with each other.

This is the man in the middle attack.

<u>Public key encryption algorithms</u>

Different algorithms for public key encryption are:

1. RSA algorithm
2. Elliptic Curve Cryptography (ECC).
3. EIGamal.

1. **RSA**

- The RSA algorithm is the most popular and proven asymmetric key cryptography algorithm.
- It is based on the mathematical fact that is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product.
- The private key and public keys in RSA are based on very large prime numbers.
- The algorithm itself is quiet simple.
- Real challenge in the case of RSA is the selection and generation of the public key and private key.
- The RSA algorithm
    1. Choose two large prime numbers p and q.
    2. Calculate n=p x q.
    3. Select public key e such that it is not a factor of (p-1) and (q-1).
    4. Select the private key d such that the following equation is true:
       (d x e) mod (p-1) x (q-1) =1.
    5. For encryption calculate the cipher text ct from plain text pt as follows:
       $Ct=pt^e \bmod n$.
    6. Send ct as cipher text to the receiver.
    7. For decryption , calculate the plain text pt from cipher text ct as follows:
       $pt^d \bmod n$.

  It would take more than 70 years to find p and q if n is a 100 digit number.

2. **Elliptic Curve Cryptography**

- It is gaining popularity in the last few years.
- The main difference between RSA and ECC is that ECC offers the same level of security for smaller key sizes.
- ECC is highly mathematical in nature.
- ECC algorithm
    1. An elliptic curve is similar to a curve drawn as graph on x and y axes.
    2. Consider an elliptic curve (e) with a point p.
    3. Now generate a random number d.
    4. Let q=d x p.
    5. Thus e, p, q together form public key and d is the corresponding private key.

  Mathematics says that e, p and q are the public values and the challenge is to find d. this is called elliptic curve discrete logarithm problem.

  As long as curve is big enough, it is impossible to find d.

3. EIGamal

- The EIGamal technique is a public key algorithm which can be used for both digital signatures as well as encryption.
- Its security is based on the difficulty of computing discrete logarithms in a finite field.
- EIGamal algorithm
    1. To generate a key pair, first select a prime number p and two random number g and x, so that both g and x are less than p.
    2. Then find out $y=g^x \bmod p$.
    3. The public key becomes y, g, p. both g and p are shred in a group of users.
    4. The private key is x.
    5. For encrypting the plain text m, first select random number k such that k is relatively prime to p-1.
    6. Then find out following:

       $A=g^k \bmod p$.

       $B= y^k m \bmod p$.

    7. Here $m= (ax+kb) \bmod (p-1)$, then the pair (a,b) becomes cipher text.
    8. To decrypt (a, b) to find out plain text m, calculate $m=b/a^x \bmod p$.

Q 17) What is Man-In-The- Middle attack? Alice and Bob establish a secret key using Diffie-Hellman key exchange using g=7, n=13.Alice takes x as 3 and Bob takes y as 9. Tom an intruder selects x as 8 and y as 6. Show working of Man-In-The-Middle attack.

Ans:

- A **man-in-the-middle** attack (also called as Bucket Brigade attack) is an active internet attack where the person attacking attempts to intercept, read or alter information moving between two computers.
- The Diffie-Hellman key exchange is vulnerable to a man-in-the-middle attack.
- This vulnerability is present because Diffie-Hellman key exchange does not authenticate the participants.

The way this happens is as follows:

1. Alice wants to communicate with Bob securely and therefore, she first wants to do a Diffie-Hellman key exchange with him. For this purpose, she sends the values of n and g to Bob. Let n=13 and g=7 (These values will form the basis of Alice's A and Bob's B, which will be used to calculate the symmetric key k1=k2=k.)
2. Alice does not realize that the attacker Tom is listening quietly to the conversation between her and Bob. Tom simply picks up the values of n and g and also forwards them to Bob as they originally were (i.e. n=13 and g=7). This is shown in the Fig.17.1.

Alice	Tom	Bob
n=13, g=7	n=13, g=7	n=13, g=7

Fig. 17.1 Man-in-the-middle attack – Part I

3. Now, let us assume that Alice, Tom and Bob select random numbers x and y as shown in Fig.17.2.

Alice	Tom	Bob
x=3	x=8, y=6	y=9

Fig. 17.2 Man-in-the-middle attack – Part II

4. Now, based on these values, all the three persons calculate the values of A and B as shown in Fig.17.3. Note that Alice and Bob calculate only A and B, respectively. However, Tom calculates both A and B.

Alice	Tom	Bob
$A = g^x \bmod n$	$A = g^x \bmod n$	$B = g^y \bmod n$
$= 7^3 \bmod 13$	$= 7^8 \bmod 13$	$= 7^9 \bmod 13$
$= 343 \bmod 13$	$= 5764801 \bmod 13$	$= 40353607 \bmod 13$
$= 5$	$= 3$	$= 8$
	$B = g^y \bmod n$	
	$= 7^6 \bmod 13$	
	$= 117649 \bmod 13$	
	$= 12$	

Fig. 17.3 Man-in-the-middle attack – Part III

5. Now, the real drama begins
   a) Alice sends her A (i.e. 5) to Bob. Tom intercepts it and instead, sends his A (i.e. 3) to Bob. Bob has no idea that Tom hijacked Alice's A and has instead given his A to Bob.
   b) In return, Bob sends his B (i.e. 8) to Alice. As before, Tom intercepts it and instead, sends his B (i.e. 12) to Alice. Alice thinks that this B came from Bob. She has no idea that Tom had intercepted the transmission from Bob and changed B.
   c) Therefore, at this juncture, Alice, Tom and Bob have the values of A and B as shown in Fig.17.4.

Alice	Tom	Bob
A=5, B=12*	A=5, B=8	A=3*, B=8

(Note: * indicates that these are the values after Tom hijacked and changed them.)

Fig. 17.4 Man-in-the-middle attack – Part IV

6. Based on these values, al the three persons now calculate their keys as shown in Fig.17.5

Alice	Tom	Bob
$K1 = B^x \bmod n$	$K1 = B^x \bmod n$	$K2 = A^y \bmod n$
$= 12^3 \bmod 13$	$= 8^8 \bmod 13$	$= 3^9 \bmod 13$
$= 1728 \bmod 13$	$= 16777216 \bmod 13$	$= 19683 \bmod 13$
$= 12$	$= 1$	$= 1$
	$K2 = A^y \bmod n$	
	$= 5^6 \bmod 13$	
	$= 15625 \bmod 13$	
	$= 12$	

Fig. 17.5 Man-in-the-middle attack – Part V

- Tom requires two keys is because, Tom wants to communicate with Alice securely using a shared symmetric key (12) and on the other hand, he wants to communicate with Bob securely using a different shared symmetric key (1).
- Only then he can receive messages from Alice, view/manipulate them and forward them to Bob and vice versa.
- Unfortunately for Alice and Bob, both will (incorrectly) believe that they are directly communicating with each other.
- As we can see, the man-in-the-middle attack can work against the Diffie-Hellman key exchange algorithm, causing it to fail. This is plainly because the man-in-the-middle makes the actual communicators believe that they are talking to each other, whereas they are actually talking to the man-in-the-middle, who is talking to each of them.
- This attack can be prevented if Alice and Bob authenticate each other before beginning to exchange information.

Q18) What are advantages and disadvantages of symmetric and asymmetric algorithm?Discuss how advantages of both can be used to ensure secure key exchange

Ans:

Symmetric key algorithm:

* Faster and easier to implement
* Lower overhead on system resources

Asymmetric key algorithm

* Scalable and does not require much administration
* Easier for users to use

Symmetric Cryptography:

Symmetric cryptography uses a single private key to both encrypt and decrypt data. Any party that has the key can use it to encrypt and decrypt data. They are also referred to as block ciphers.

Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data.

The disadvantage of symmetric cryptography is that it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. This is a significant challenge. Symmetric algorithms are usually mixed with public key algorithms to obtain a blend of security and speed.

Asymmetric, or Public Key, Cryptography:

Public-key cryptography is also called asymmetric. It uses a secret key that must be kept from unauthorized users and a public key that can be made public to anyone. Both the public key and the private key are mathematically linked; data encrypted with the public key can be decrypted only by the private key, and data signed with the private key can only be verified with the public key.

The public key can be published to anyone. Both keys are unique to the communication session.

Public-key cryptographic algorithms use a fixed buffer size. Private-key cryptographic algorithms use a variable length buffer. Public-key algorithms cannot be used to chain data together into streams like private-key algorithms can. With private-key algorithms only a small block size can be processed, typically 8 or 16 bytes.

Asymmetric vs. Symmetric

At the most basic level, each method of encryption can be categorised as being asymmetric or symmetric - either you need one key to encrypt and another to decrypt, or the same key works both ways. Each has their own advantages and disadvantages, as per usual.

Asymmetric encryption, also known as public key encryption, means that you have one key for encryption (a public key) and another key for decryption (a private key). The public key can be given away freely to whoever wants it - after all, with just the public key your data cannot be

read, because people would need the private key to decrypt your data. For example, if you sent out your public key to everyone who emails you, they would all be able to send you encrypted emails that only you would be able to decrypt and read.

Symmetric encryption, also known as secret key encryption, means that you have one key for encryption that works also as the key for decryption. The easiest symmetric encryption method to understand is called ROT13, and legend would have us believe it was first used by Julius Caesar to encrypt orders going to Roman troops. ROT13 works by shifting all letters in the alphabet thirteen places to the right - A becomes N, B becomes O, Z becomes M, etc. Because there are twenty-six characters in the English alphabet, performing ROT13 a second time results in the original text again - N becomes A, O becomes O, and M becomes Z.

Symmetric encryption is very easy to use, and usually very fast too. On the other hand, symmetric encryption keys must be kept secure - you would need to make sure each person who needs the key gets it without any risk of it getting out.

Asymmetric encryption is generally slower than symmetric encryption, however the public keys they use are safe to be published anywhere (even on the Internet) because to get the private key from a public key could take hundreds of years of work.

Advantages of Asymmetric-key cryptography
-Only the private key must be kept secret
-The administration of keys on a network requires the presence of only a functional trusted TTP as opposed to an unconditionally trusted TTP.
-A private/public key pair key pair main remain unchanged for considerable long periods of time (depending on the usage)
-There are many relatively efficient digital signature mechanisms as a result of asymmetric-key schemes
-In a large network the number of keys necessary may be considerably smaller than in the symmetric-key scenario

Disadvantages of Asymmetric-key cryptography
-Slower throughput rates than the best known symmetric-key schemes
-Large key size
-No asymmetric-key scheme has been proven to be secure
-Lack of extensive history

Symmetric-Key Management

Advantages:
*It is easy to add and remove entities from the network
*Each entity needs to store only one long-term secret key

Disadvantages:
*All communications require initial interaction with the TTP
*The TTP must store n long-term secret keys
*The TTP has the ability to read all messages
*If the TTP is compromised, all communications are insecure

Asymmetric-Key Management

Advantages:
*No TTP is required
*The public file could reside on each entity
*Only n public keys need to be stored to allow secure communications between any pair of
*entities, assuming the only attack is that by a passive adversary

Disadvantages:
*If the signing key of the TTP is compromised, then all communications become insecure
*All trust is placed with one entity

Q 19) Explain RSA with eg In a RSA system the public key of a given user is c=31,n=3599.What is a private key of the user? Perform encryption and Decryption using RSA for foll.P=3 q=11 e=7 m=5

Ans:

RSA algorithm:

The RSA Algorithm is the most popular and proven asymmetric key cryptographic algorithm. The RSA algorithm is based on the mathematical fact that it is easy to find and multiply large prime numbers together, but it is extremely difficult to factor their product. The private and public keys is RSA is based on very large (made up of 100 or more digits) prime numbers. The algorithm itself is quite simple (unlike the symmetric key cryptographic algorithm).However the real challenge in the case of RSA is the selection and generation of the public and private keys.

Algorithm steps:

1. Choose two large prime numbers P and Q.
2. Calculate N= P x Q
3. Select the public key (i.e. the encryption key) E such that it is not a factor of (P -1) and (Q-1).
4. Select the private key (i.e. the decryption key) D such that the following equation is true : (D X E) mod (P -1) X (Q-1) =1
5. For encryption calculate the cipher text CT from the plain text PT as follows : $CT = PT^E \mod N$
6. Send CT as the cipher text to receiver.
7. For decryption calculate the plain text PT from the cipher text CT as follows : $PT = CT^D \mod N$

Example:

For Example:

1. Take P=2 & Q=5
2. N=P.Q = 10
3. (P-1) (Q-!)=1.4 =4
4. Factors of 4 are:2,2
   Choose 'E' so that it is not a factor of '2'
5. Let E=3
6. To find D solve
   D x 3 mod 4 =1
   D = Q x 4+1/3
   D = 3 when Q =2

7. At the sender let ‘PT = y’
   CT = PT^E mod N
   CT= 25^3 mod 10
   CT= 5
8. At the receiver
   PT = CT^D mod N
   PT = 5^3 mod 10
   PT = 5

Solution: c=31, n=3599

The private key can be written as {d, n}

To calculate d, we have to 1st calculate Φ(n), where Φ(n) is the number of prime factors of n

c= 31, n = 3599

It means that e is relatively prime to Φ(n), i.e. gcd (Φ(n), e) = 1

To find Φ(n), we have to first check whether n is prime or not, and if it is not, what are the prime factors of n.

Step1: To find prime factors of n

By using trial and error we find that n is not prime and the prime factors of n are

n = 3599 = 61 * 59

Thus p = 61 and q = 59 are both prime numbers.

Step 2: To calculate Φ(n)

Here we find the Euler’s totient function written as Φ(n), where Φ(n) is the number of positive integers less than n and relatively prime to n.

It is clear that for a prime number p,

$$\Phi(p) = p-1$$

Now since we have two prime numbers p and q, with $p \neq q$, Then for n= pq,

$$\Phi(n) = \Phi(pq) = \Phi(p) * \Phi(q) = (p-1) * (q-1)$$

Therefore

$$\Phi(3599) = \Phi(61) * \Phi(59) = 60 * 58 = 3480$$

Step 3: Calculate d

We know that if gcd (Φ(n), e) = 1, then e has a multiplicative inverse modulo Φ(n)

Therefore,

$$d \equiv e^{-1} \bmod \Phi(n)$$

i.e. d is multiplicative inverse of e mod Φ(n)

$d = (1 + k\,\varphi(n))/e = (1 + 3480k)/31 = -13921/31 = -449$ (for k = -4)
$d = -449 \pmod{3480} = 3031 \pmod{3480}$

Hence it is proved that d = -449 is multiplicative inverse of e = 31 mod Φ(n)

So the private key of this user will be {-449, 3599}

p=3; q=11; e=7; M=5

Answer:

n = p * q = 3 * 11 = 33

f(n) = (p-1) * (q-1) = 2 * 10 = 20

Now, we need to compute $d = e^{-1} \bmod f(n)$ by using backward substitution of GCD algorithm:

According to GCD:

$$20 = 7 * 2 + 6$$

$$7 = 6 * 1 + 1$$

$$6 = 1 * 6 + 0$$

Therefore, we have:

$$1 = 7 - 6$$

$$= 7 - (20 - 7 * 2)$$

$$= 7 - 20 + 7 * 2$$

$$= -20 + 7 * 3$$

Hence, we get $d = e^{-1} \bmod f(n) = e^{-1} \bmod 20 = 3 \bmod 30 = 3$

So, the public key is {7, 33} and the private key is {3, 33}, RSA encryption and decryption is following:

Encryption:

For encryption calculate the cipher text CT from the plain text m as follows:

$CT=M^E \bmod n;$

$CT=5^7 \text{ Mod } 33= 14$

Now send CT as cipher text to the receiver i.e. send 14 as the cipher text to the receiver.

Decryption:

For decryption calculate the plain text m from the cipher text CT as follows:

$M=CT^D \bmod n$

$M=14^3 \bmod 33=5$

Q 20) Explain EI-Gamal Signatures

Ans: The **ElGamal signature scheme** is a digital signature scheme which is based on the difficulty of computing discrete logarithms.

The ElGamal signature scheme allows that a verifier can confirm the authenticity of a message *m* sent by the signer sent to him over an insecure channel.

EIGamal

- The EIGamal technique is a public key algorithm which can be used for both digital signatures as well as encryption.
- Its security is based on the difficulty of computing discrete logarithms in a finite field.
- EIGamal algorithm
    9. To generate a key pair, first select a prime number p and two random number g and x, so that both g and x are less than p.
    10. Then find out $y=g^x \bmod p$.
    11. The public key becomes y, g, p. both g and p are shred in a group of users.
    12. The private key is x.
    13. For encrypting the plain text m, first select random number k such that k is relatively prime to p-1.
    14. Then find out following:

        $A=g^k \bmod p$.

        $B= y^k m \bmod p$.

    15. Here $m= (ax+kb) \bmod (p-1)$, then the pair (a,b) becomes cipher text.
    16. To decrypt (a, b) to find out plain text m, calculate $m=b/a^x \bmod p$.

Q 21)What is KDC? How is it different from CA ? How does a KDC work with multiple domain If there is a revocation mechanism why do certificates need an expiry date

Ans: In cryptography, a **key distribution center** (**KDC**) is part of a cryptosystem intended to reduce the risks inherent in exchanging keys.

KDCs often operate in systems within which some users may have permission to use certain services at some times and not at other

For instance, an administrator may have established a policy that only certain users may use the tape backup facility. (Perhaps the administrator has concerns that unrestricted use might result in someone smuggling out a tape containing important information; but the precise reason does not matter for the purpose of explaining the functioning of the key distribution center.)

Many operating systems can control access to the tape facility via a 'system service'. If that system service further restricts the tape drive to operate on behalf only of users who can submit a service-granting ticket when they wish to use it, there remains only the task of distributing such tickets to the appropriately permitted users.

If the ticket consists of (or includes) a key, we can then term the mechanism which distributes it a KDC. Usually, in such situations, the KDC itself also operates as a system service

A typical operation with a KDC involves a request from a user to use some service. The KDC will use cryptographic techniques to authenticate requesting users as themselves. It will also check whether an individual user has the right to access the service requested. If the authenticated user meets all prescribed conditions, the KDC can issue a ticket permitting access.

KDCs mostly operate with symmetric encryption.

In most (but not all) cases the KDC shares a key with each of all the other parties.

The KDC produces a ticket based on a server key.

The client receives the ticket and submits it to the appropriate server.

The server can verify the submitted ticket and grant access to the user submitting it.

Security systems using KDCs include Kerberos

Benefits

- Easier key distribution
- Scalability

Drawbacks:

- A KDC can become a **single point of failure**
- Everybody must trust the KDC
- Vulnerable to **replay attack**

Certificate Authority

In cryptography, a **certificate authority** or **certification authority** (**CA**) is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. In this model of trust relationships, a CA is a trusted third party that is trusted by both the subject (owner) of the certificate and the party relying upon the certificate. CAs are characteristic of many public key infrastructure (PKI) schemes

Revocation necessity and expiration of certificate

A malicious (or erroneous) revocation of some or all of the keys in the system is likely, or in the second case, certain, to cause a complete failure of the system.

If public keys can be revoked individually, this is a possibility. However, there are design approaches which can reduce the practical chance of this occurring.

For example, by means of certificates we can create what is called a "compound principal"; one such principal could be "Alice and Bob have Revoke Authority". Now only Alice and Bob (in concert) can revoke a key, and neither Alice nor Bob can revoke keys alone. However, revoking a key now requires both Alice and Bob to be available, and this creates a problem of reliability. In concrete terms, from a security point of view, there is now a single point of failure in the public key revocation system. A successful Denial of Service attack against either Alice or Bob (or both) will block a required revocation. In fact, any partition of authority between Alice and Bob will have this effect, regardless of how it comes about.

Because the principle allowing revocation authority for keys is very powerful, the mechanisms used to control it should involve both as many participants as possible (to guard against malicious attacks of this type), while at the same time as few as possible (to ensure that a key can be revoked without dangerous delay). Public key certificates which include an expiry date are unsatisfactory in that the expiry date may not correspond with a real world revocation need, but at least such certificates need not all be tracked down system wide, nor must all users be in constant contact with the system at all times.

Q 22) Discuss concept of certificate revocation.

Ans: - Reasons for the revocation of digital certificate->

- The holder of the digital certificate reports that the private key corresponding to the public key specified in the digital certificate is compromised.
- The CA realizes that it had made some mistake while issuing a certificate.
- The certificate holder leaves a job and the certificate was issued specifically while the person was employed in that job.

For this reasons, the certificate must be revoked. Just like a credit card loss or theft, a certificate holder should report that a certificate be revoked. If user leaves an organization or indulges in an illegal act because of which the certificate needs to be revoked, the organization should initiate the process. Finally, if the CA realizes its own mistake in providing a wrong certificate, the CA initiates revocation process.

The CA must come to know about this certificate revocation request. Also, the CA must authenticate revocation requester before accepting this revocation request.

Let us assume that Alice want to use Bob's certificate for securely communicating with him. However, before she uses Bob's certificate, Alice want to answers to the following question:

- Does this certificate really belong to Bob?
- Is this certificate valid or is it revoked?
- For this CA provides the facilities as shown in following figure.

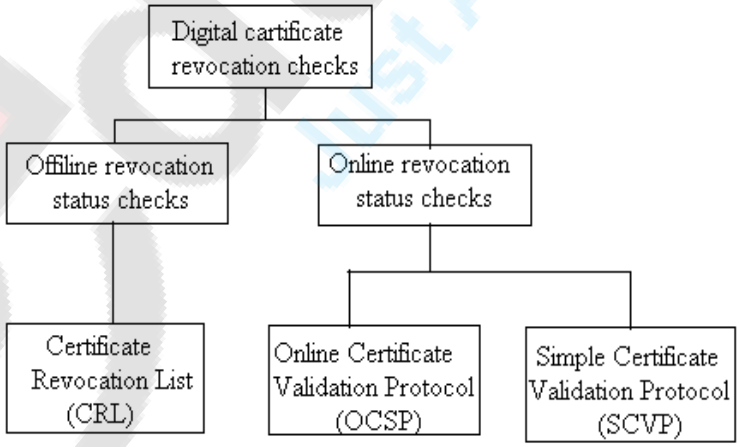


Fig. Certificate revocatoin status mechanism.

Offline Certificate Revocation Status Checks The Certificate Revocation List (CRL) is the primary means of checking the status of a digital certificate offline. In its simplest form, a CRL is a list of certificate published regularly by each CA, identifying all the certificates that have been revoked through the life of the CA. A CRL lists only those certificates whose validity period is still within the acceptable range, but they are revoked for some other reason.

A CRL is a sequential file that grows over time to include all the certificates that have not expired, but lists have been revoked. Thus, it is a superset of all the previous CRLs issued by that CA. Each CRL entry lists the certificate serial number, the date and time on which the certificate was revoked and the reason behind the revocation.

Thus, when Alice receives Bob's certificate and wants to see if she should trust it, she should do the following, in the given sequence:

- **Certificate expiry check:** compare the current date with the validity in term of the signature by his CA ensure that certificate has not expired.
- **Signature check:** check that Bob's certificate can be verified in term of the signature by His CA.
- **Certificate revocation check:** consult the latest CRL issued by Bob's CA to ensure that Bob's certificate is not listed there as a revoked certificate.

Only after Alice is assured of all these three aspects that she can trust Bob's certificate.

A CRL can become really quite big over time. The general assumption is that about 10% of unexpired certificate will be revoked every year. That's why CA can send a one time full up-to-date CRL to the users who want to use the CRL service. This is called as the base CRL. However, at the time of the next update, the CA need not send the entire CRL file once again. This mechanism makes the CRL file size small and therefore, its transmission easier. The changes to the CRL are called as delta CRL. The difference between issuing a complete CRL every time versus only a delta CRL is shown in following figure.

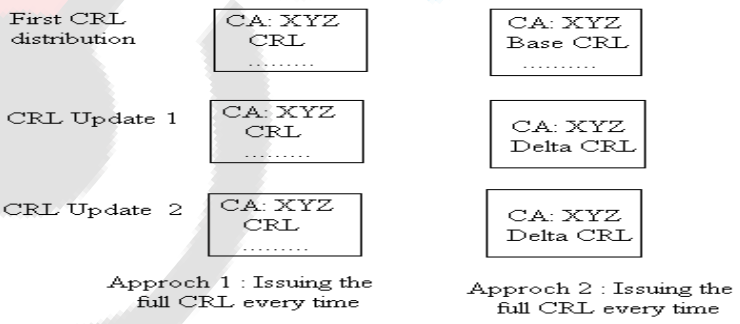


Fig. Delta CRL

The CRL is called offline because the CRL is issued periodically by the CA. This period can vary from few hours to a few weeks. CA always issue new CRLs periodically even if there are no change to the previous CRL.

Online Certificate Revocation Status Checks:-

Two new protocols were developed for checking the status of a certificate online, namely the Online Certificate Status Protocol (OCSP) and Simple Certificate Validation Protocol (SCVP).

- Online Certificate Status Protocol (OCSP):- The Online Certificate Status Protocol(OCSP) can be used to check if a given digital certificate is valid at a particular moment. Thus, it is an online check. OCSP allows the certificate validators to check for the status of certificates in real time, thus providing for quicker, simple and more efficient mechanism for digital certificate validation. Lets us understand how OCSP works, step-by-step.

1) The CA provides a server, called as an OCSP responder. This server contains the latest certificate revocation information. The requestor (client) has to send a query about a particular certificate to check whether it is revoked or not. In practice, the OCSP request contains the OCSP protocol version, the service requested and one or more certificate identifiers.
2) The OCSP responder consults the servers X.500 directory to see if the particular certificate is valid or not.
3) Based on the result of the status check from X.500 directory lookup, the OCSP responder sends back a digitally signed OCSP response for each of the certificates in the original to the client. The OCSP response may also include the date, time and reason for revocation, if the certificate is revoked. The client has to determine the action to be taken accordingly. The recommendation is to consider the certificate as valid if the OCSP response is good.

- Simple certificate Validation Protocol (SCVP):- The SCVP is in the draft stage of the current writing. SCVP is an online certificate status reporting protocol, designed to deal with the drawbacks of OCSP. Since SCVP is conceptually similar to OSCP with minor differences.

Q 23) What is authentication? How can be achieved with the help of token.

Ans:

Authentication:- One of the key aspects of cryptography and network/Internet security is authentication. Authentication helps trust by identifying who a particular user / system is. Authentication can be defined as determining an identity to the required level of assurance. It is the first step in any cryptographic solution.

Authentication tokens:

An **authentication token** is an extremely useful alternative to a password. An authentication token is a small device that generates a new random value every time it is used. This random value becomes the basis for authentication. The small devices are typically of the size of small key chains, calculators or credit cards. Usually an authentication token has the following features:

- Processer
- Liquid Crystal Display(LCD) for displaying outputs
- Battery
- (Optionally) a small keypad for entering information
- (Optionally) a real-time clock

Each authentication token (i.e. each device) is pre-programmed with a unique number, called as a random seed, or just seed. The seed forms the basis for ensuring the uniqueness of the output produced by the token.

Step 1:-Creation of a token:-

Whenever an authentication token is created, the corresponding random seed is generated for the token by the authentication server. This seed is stored or pre-programmed inside the token, as well as its entry is made against that user's record in the user database. Conceptually, think about this seed as the user's password (although this is technically completely different from a password). Also the user does not know about the value of the seed, unlike a password. This is because the seed is used automatically by the authentication token.

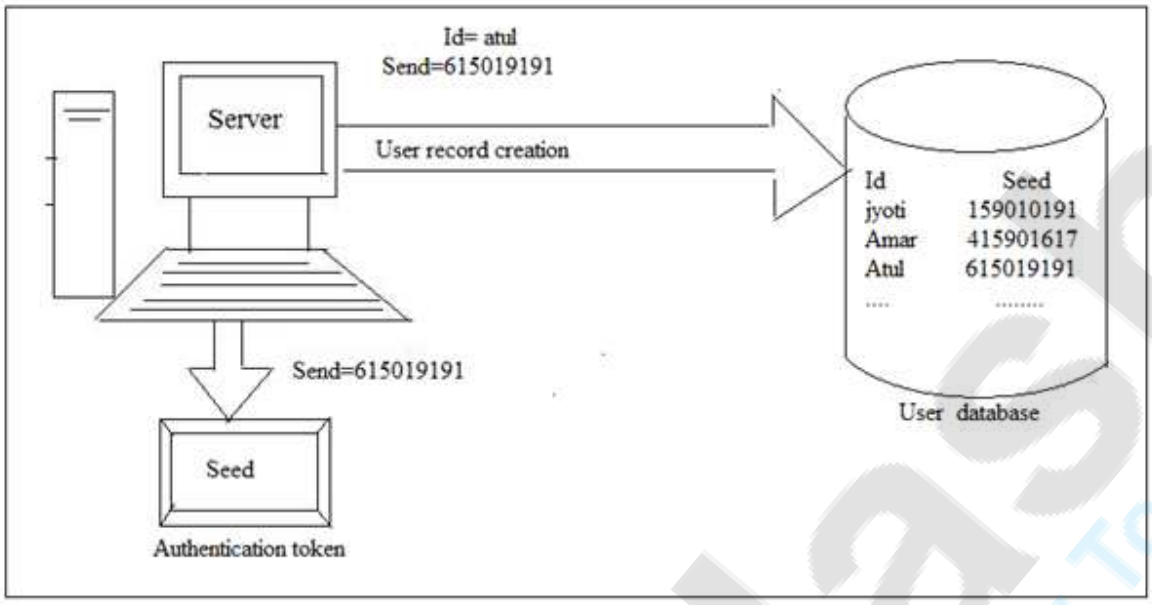
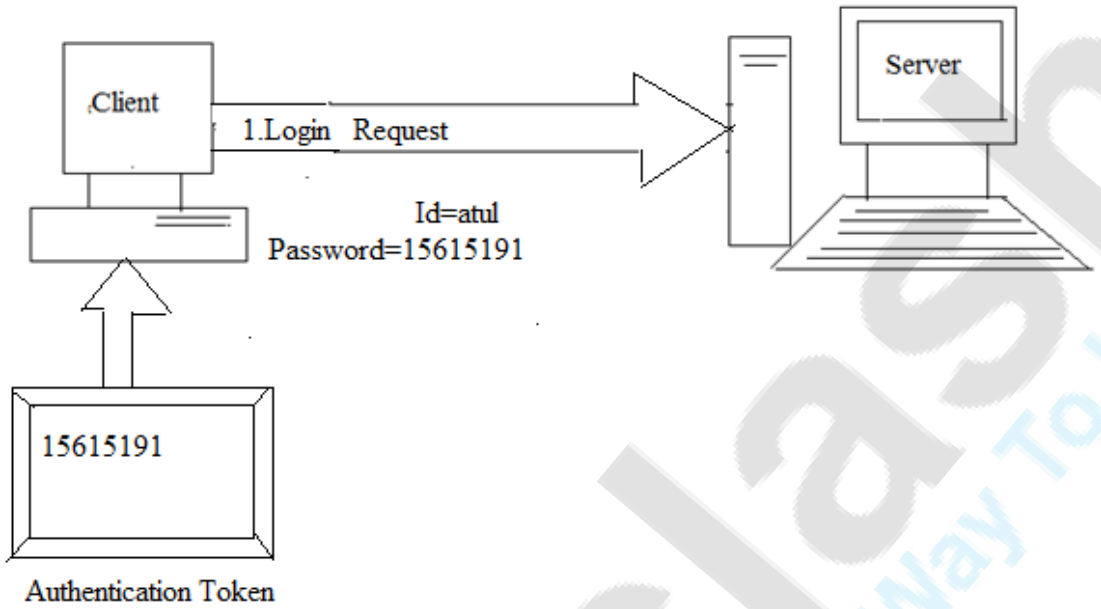


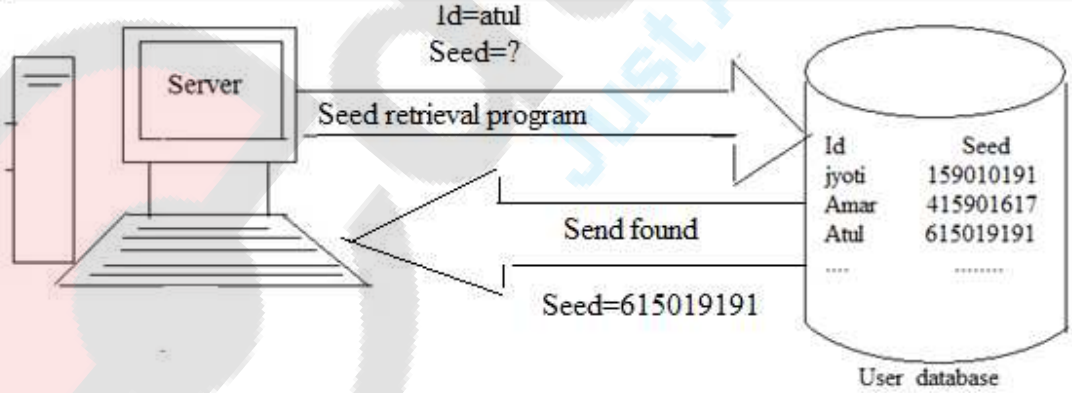
Fig:- Random seed storage in the database and the authentication token

Step 2:-**Use of token**:- An authentication token automatically generates pseudorandom numbers, called as **one time passwords** or **one-time passcodes**. One-time passwords are generated randomly by an authentication token, based on the seed value that they are pre-programmed with. They are one-time because they are generated, used once , and discarded for ever. When a user wants to be authenticated, the user will be get a screen to enter the user id and the latest one-time password. For this, the user will enter the user id and the one-time password obtained from the authentication token. The user id and password travel to the server as a part of the login request. The server obtains the seed corresponding to the user id from the user database, using a Seed retrieval program. It then calls another program called as Password validation program, to which the server gives the seed and the one-time password. This program knows how to establish the relationship between the seed and the one-time password. How this is done beyond the scope of the current text. , but to explain it in simple terms, the program use synchronization techniques, to generates the same one-time password as was done by the authentication token. However, the main point to be noted here is that the authentication server can use this program to determine if a particular seed value relates to a particular one-time password or not.

Step 1: The user's id and the one-time password obtained from the authentication token are sent to the server



Step 2: The server 's seed retrieval program now retrieves the seed for this user from the user database



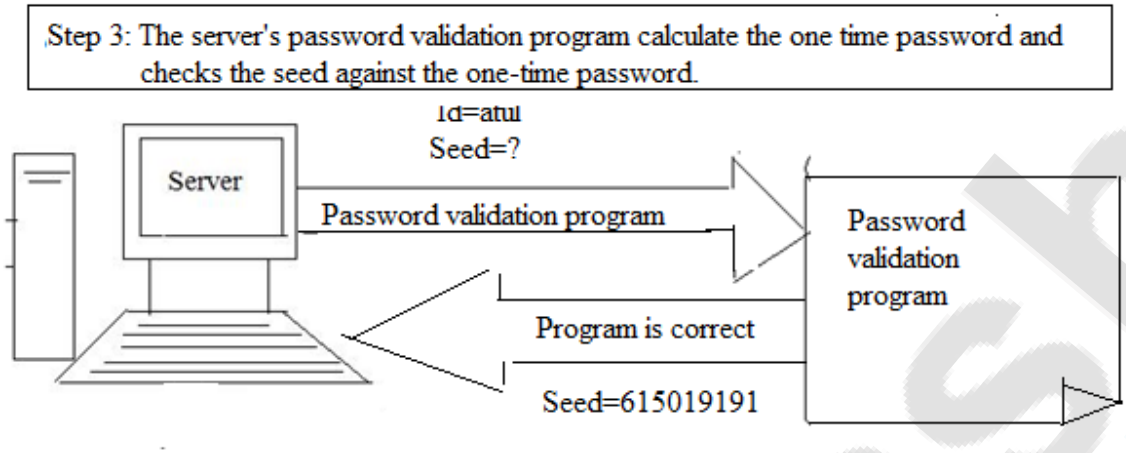


Fig:- Server validations the one-time password

A question at this stage could be, what would happen if a user loses an authentication token? Can another user simply grab it and use it? To deal with such situations, the authentication token is generally protected by a password or a 4-digit pin. Only when this PIN is entered can the one-time password be generated. This is also the basis for what is called as multi-factor authentication. What are this factors? There are three most common factors:

- Something that you know, e.g. a password or PIN
- Something you have , e.g. a credit card or an identity card

- Something you are ,e.g. your voice or finger print

Based on these principles, we can see that a password is a 1-factor authentication, because it is only something that you know. In contrast, authentication tokens are examples of 2-factor authentication, because here you must have something (the authentication token itself) and you must also know something (the PIN used to protect it). Someone only knowing the PIN or only having the cannot use it – both the factors are required for the authentication token to be used.

Step 3: server returns an appropriate message back to the user :- Finally, the server sends an appropriate message back to the user, depending on whether the previous operations yielded success or failure. This is shown in figure.

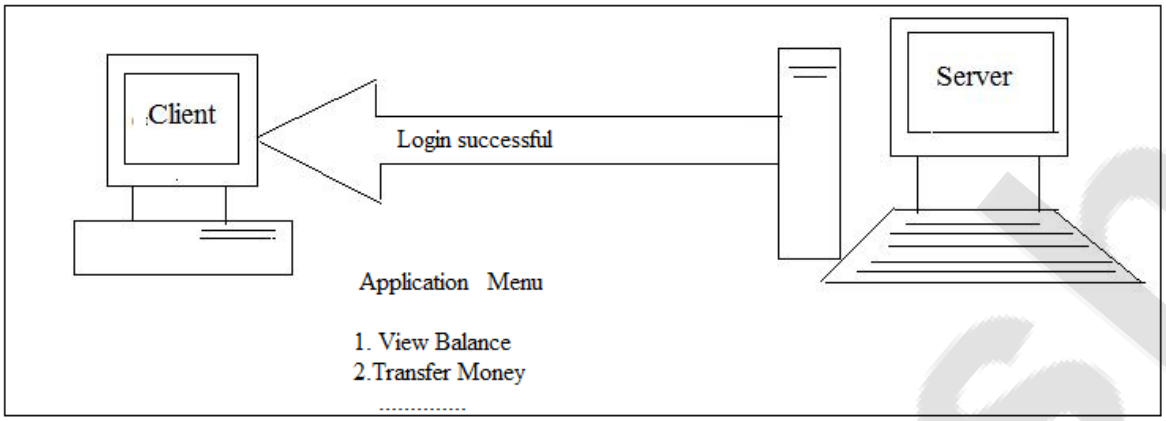


Fig:- Server sends an appropriate message back to the user

- Authentication token type

There are two main types of authentication token:-

1. Challenge/Response Tokens
2. Time-based Tokens

Q 24) How Symmetric Key Algorithm RSA used to get Digital Signature?

Ans: Digital signature is a mechanism by which a message is authenticated i.e. proving that a message is effectively coming from a given sender, much like a signature on a paper document. For instance, suppose that Alice wants to digitally sign a message to Bob. To do so, she uses her private-key to encrypt the message; she then sends the message along with her public-key (typically, the public key is attached to the signed message). Since Alice's public-key is the only key that can decrypt that message, a successful decryption constitutes a Digital Signature Verification, and meaning that there is no doubt that it is Alice's private key that encrypted the message.

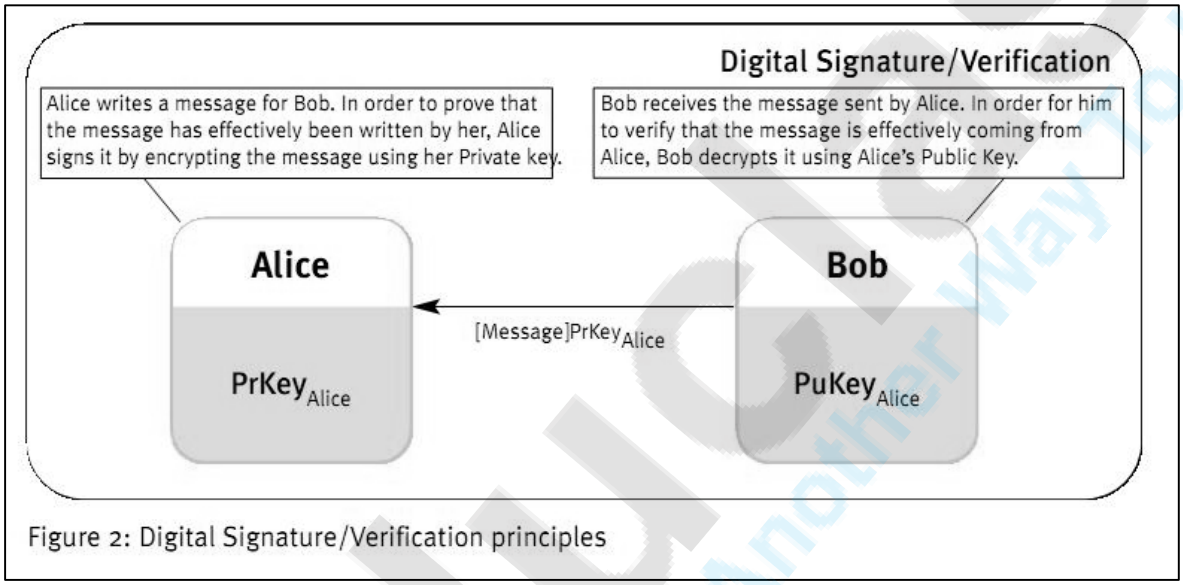


Figure 2: Digital Signature/Verification principles

Both encryption and digital signature can be combined, hence providing privacy and authentication. Symmetric-key plays a major role in public-key encryption implementations. This is because asymmetric-key encryption algorithms are somewhat slower than symmetric-key algorithms.

RSA Digital Signatures/Verification Scheme :

- Digital signatures are always computed with private key. This makes them easily verifiable publicly with the public key.
- The raw message m is never signed directly. Instead it is usually hashed with hash function and the message digest is signed. This condition usually also means that the message m in fact is not secret to the parties so that each party can compute the message digest separately. It is also possible to use so called redundancy function instead of hash function. This function is reversible which makes it possible to sign secret messages since

the message can be retrieved by the party verifying the signature. In practice hash function is often used.

If the m is not hashed or run through redundancy function several attacks exist against RSA signatures which may make it possible to forge signatures. Also if the redundancy function is insecure it may be possible to forge signatures.

Steps For Signing And Encrypting A Message :

1. **Computing signature:**

First run the message through the hash function (or redundancy function): m' =H(m), then compute s= m^d mod n, where the n is the modulus (from public key) and d is the private key. The end result is s which is the signature.

Figure 3 below shows the set of operations required when Alice wants to send a signed and encrypted message to Bob.

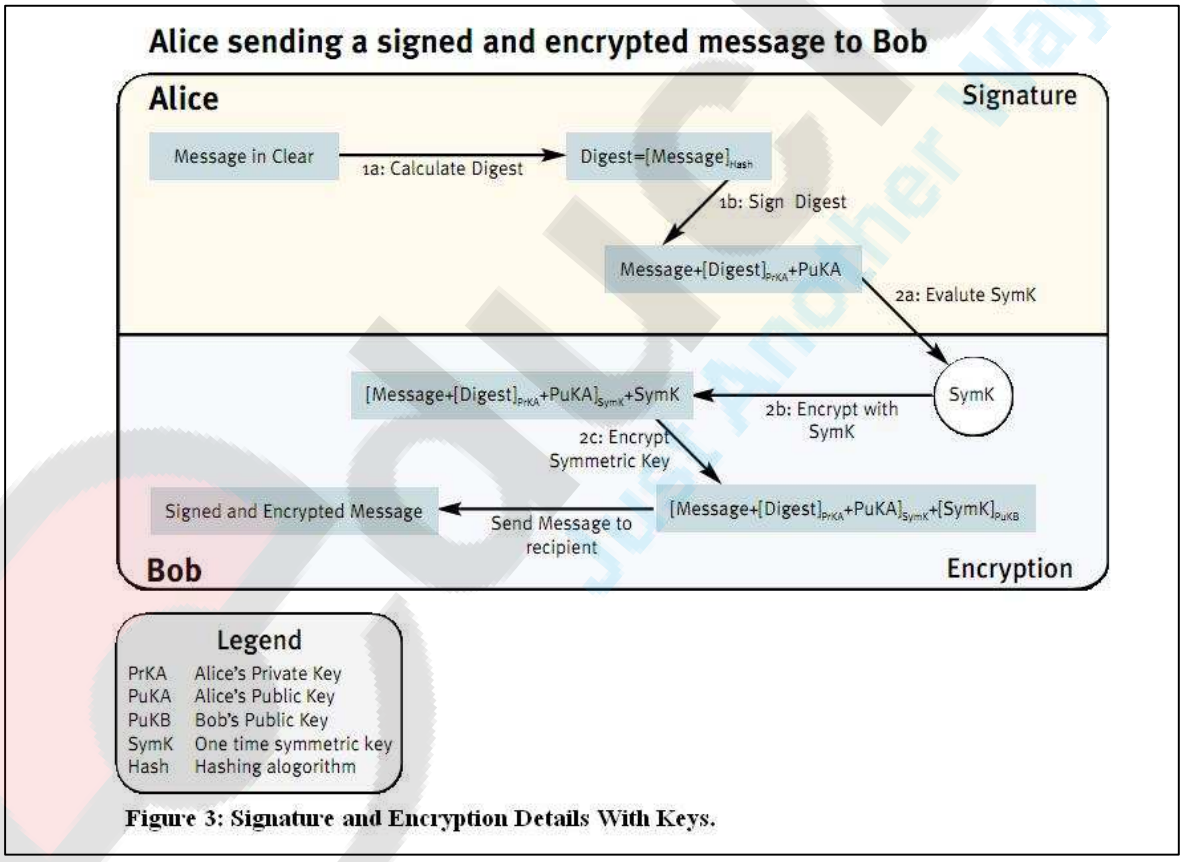


Figure 3: Signature and Encryption Details With Keys.

Message signature:

Digital signature includes two steps:

i) **Message digests evaluation.** The main purpose for evaluating a digest is to ensure that the message is kept unaltered; this is called message integrity.
ii) **Digest signature.** A signature is in fact an encryption using the issuer's (Alice in this case) private-key. Included in the signature is also the hashing algorithm name used by the issuer. The issuer's public-key is also appended to the signature. Doing so lets anyone decrypt and verify the signature using the issuer's public-key and hashing algorithm. Given the properties of public-key encryption and hashing algorithms, the recipient has proof that:
   (1) The issuer's private-key has encrypted the digest.
   (2) The message is protected against any alteration.

b) **Message encryption**. Encryption includes the following 3 steps:
   i) Creation of a onetime symmetric encryption/decryption key. Remember that encryption and decryption algorithms using asymmetric-keys are too slow to be used for long messages; symmetric-key algorithms are very efficient and are therefore used.
   ii) Message encryption. The whole message (the message itself and the signature) is encrypted using SymK, the symmetric-key evaluated above.
   iii) Symmetric-key encryption. SymK is also used by the recipient to decrypt the message. SymK must therefore be available to the recipient (Bob) only. The way to hide the Symk from everybody except the recipient is to encrypt it using the recipient's public-key. Since SymK is a small piece of information compared to a message (that could be very long), the performance penalty associated with the relative inefficiency of asymmetric-key algorithms is acceptable.

[One interesting point to mention is that if Alice wants to send the same message to more than one recipient, say Bob and John for instance, the only additional operation to be performed is to repeat 'step 2) c)' for John. Hence, the message that both Bob and John would receive would look like:

[Message+[Digest]PrKA+PuKA]SymK+[SymK]PuKB+[SymK]PuKJ . Notice that the exact

same SymK will be used by Bob and John to decrypt the message.]

2. Verifying the signature:

$m' = s^d \bmod n$. If hash function was used then the m is run through the hash function and the message digest is verified against m'. If the verification fails the

signature is not authentic. If redundancy function was used then the redundancy function defines how the m' is verified. In this case also the m may be retrieved from m', which is not possible when using hash functions.

Figure 4 below shows the set of operations required when Bob wants to decrypt and verify the message sent by Alice.

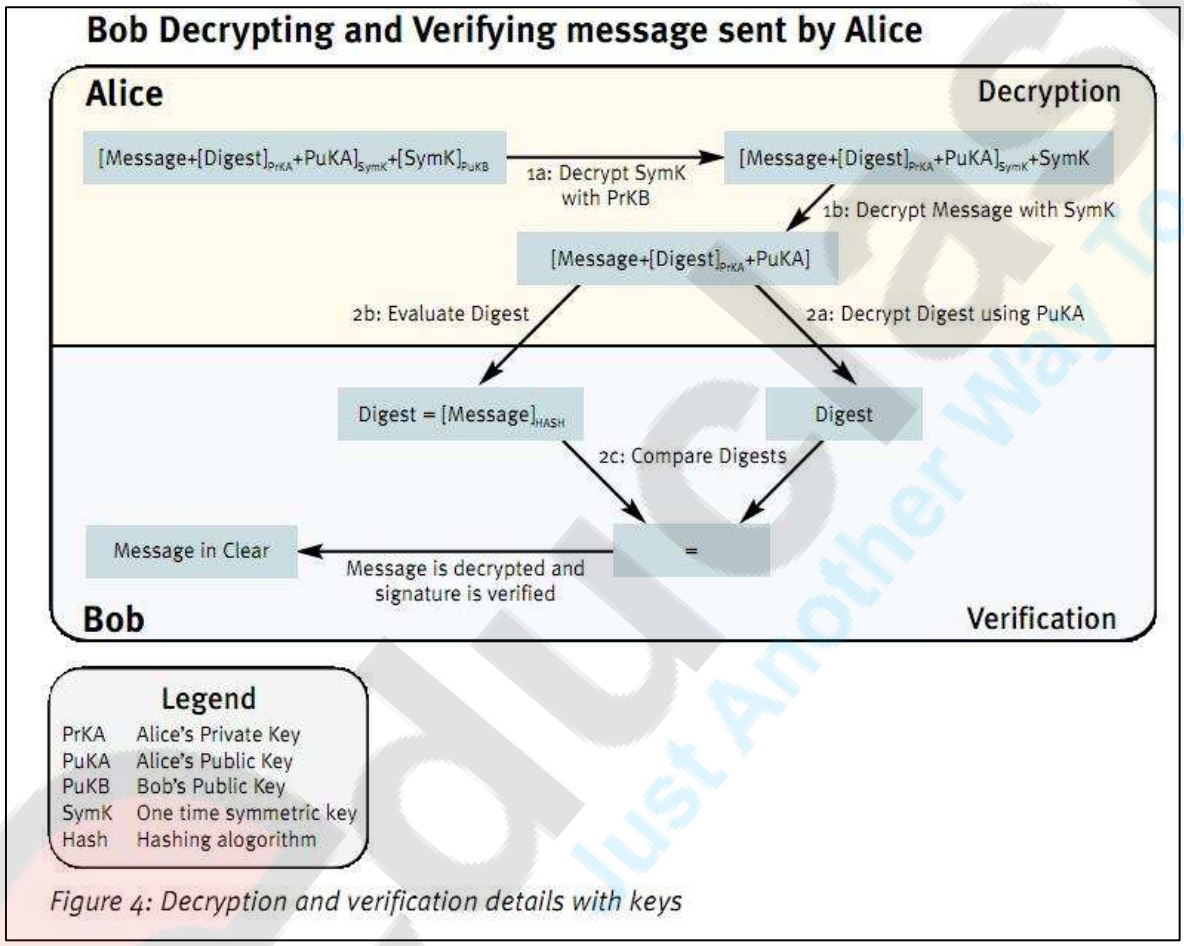


Figure 4: Decryption and verification details with keys

a) **Message decryption**. The decryption includes the following steps:
   i. Symmetric-key decryption. The one time symmetric-key has been used to encrypt the message. This key (SymK) has been encrypted using the recipient's (Bob) public-key. Only Bob can decrypt SymK and use it to decrypt the message.
   ii. Message decryption. The message (which includes the message itself and the signature) is decrypted using SymK.

b) **Signature verification**. The signature verification includes the following 3 steps:

I. Message digests decryption. The digest has been encrypted using the issuer's (Alice) private-key. The digest is now decrypted using the issuer's public-key included in the message.

II. Digest evaluation. Since hashing is a one-way process i.e. the message cannot be derived from the digest itself, the recipient must re-evaluate the digest using the exact same hashing algorithm the issuer used.

III. Digests comparison. The digest decrypted in a) and the digest evaluated in b) are compared. If there is a match, the signature has been verified, and the recipient can accept the message as coming unaltered from the issuer. If there is a mismatch this could mean that:

i. The message has not been signed by the issuer or
ii. The message has been altered.
iii. In both cases, the message should be rejected.

Q 25) Explain in brief the security mechanism used in an Electronic transaction?

Ans: Security Mechanism**:**

Security mechanisms are means to achieve the security services i.e

- Confidentiality
- Integrity
- Authentication,
- Non-repudiation
- Access control

There is no single mechanism that can provide all the security services. However, there is one main class of techniques that underlies most of the security mechanisms in use, namely cryptographic mechanisms.

Cryptographic Mechanism:

1. Asymmetric cryptography
2. Symmetric cryptography

1. Asymmetric cryptography:

- The concept of asymmetric cryptography, or public key cryptography, was first introduced in 1976 by Diffie and Hellman [12]. An asymmetric cryptosystem is a cryptographic scheme in which two distinct keys, known as the public key and the private key, are used.
- The concept of a Public Key Infrastructure (PKI) has been introduced as a means to generate, distribute and manage `public key certificates' which are used to bind the identifier of a party to that party's public key.

1.1 Asymmetric encryption:

- Asymmetric encryption schemes use public keys for encryption and private keys for decryption.
- The best known algorithm for public key encryption is RSA, which was proposed in 1978 by Rivest, Shamir, and Adleman
- It can be used to provide data confidentiality.

1.2 Digital Signature:

- Digital signature mechanisms are asymmetric cryptographic techniques which can be used to provide entity authentication, data origin authentication, data integrity and non-repudiation services'.

2. Symmetric cryptography:
Symmetric cryptography is a cryptographic scheme in which either the same key (secret key) or two keys that can be easily computed from each other are used.
There are a number of symmetric cryptographic schemes, including encryption schemes, message authentication codes, and cryptographic hash functions.

2.1 Symmetric encryption:

- It uses a single key for both the encryption and decryption transformation

Types of symmetric encryption scheme are

- **stream ciphers** : an encryption that encrypts a plaintext in bit-wise manner.
- **block ciphers:** an encryption scheme which breaks up the plaintext messages to be transmitted into strings (called blocks) of a fixed length and encrypts one block at a time

2.2 Cryptographic hash function:

- `A hash function is a function which maps strings of bits to fixed-length strings of bits'

It must also satisfy the following three properties.

- It must be computationally infeasible to find for a given output, an input which maps to this output, and
- It must be computationally infeasible to find for a given input, a second input which maps to the same output.
- It must be computationally infeasible to find two different inputs which map to the same output.

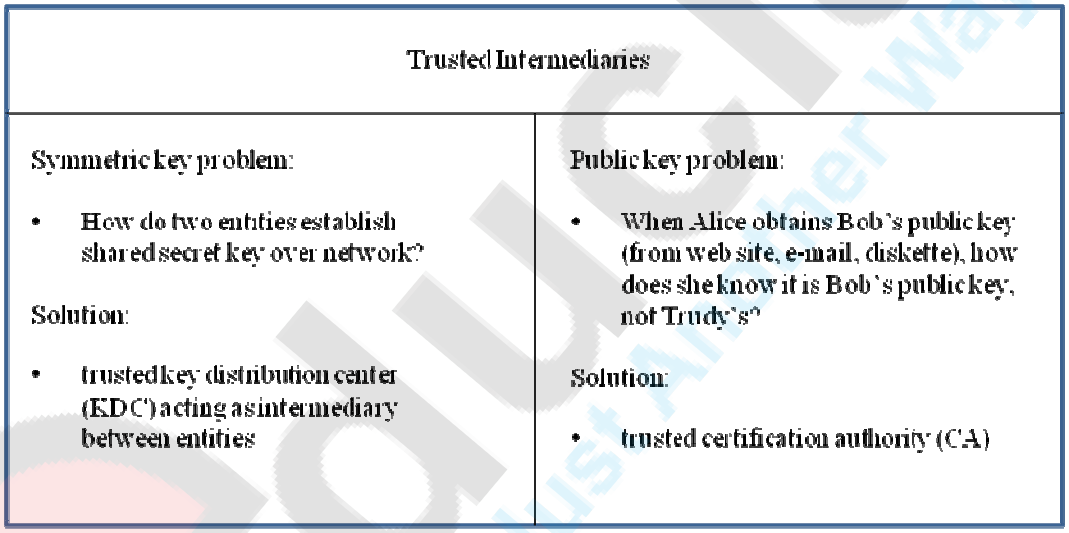
2.3 Message Authentication Code (MAC):

- Used to provide data origin authentication and data integrity services.
- The originator of data inputs the data to be protected into a MAC function, together with a secret key - the resulting output (a short fixed-length bit string) is known as the MAC.
- The verifier of the MAC simply uses the same secret key to recompute a MAC value on the data, and the data is accepted as valid if and only if the recomputed MAC agrees with the value sent or stored with the data.

Q26) What are Certification Authorities? How are they different from Key Distribution Center? Which of the two is preferable and why?

Answer: - **Certificate authority** or certification authority (CA) is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. In this model of trust relationships, a CA is a trusted third party that is trusted by both the subject (owner) of the certificate and the party relying upon the certificate. CAs is characteristic of many public key infrastructure (PKI) schemes.

KDC: - In cryptography, a key distribution center (KDC) is part of a cryptosystem intended to reduce the risks inherent in exchanging keys. KDCs often operate in systems within which some users may have permission to use certain services at some times and not at others.



Certification Authorities are different from Key Distribution Center:-

Symmetric Key Distribution:

Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages. Symmetric-key cryptography, however, needs a shared secret key between two parties. The number of keys and distribution of keys is another problem.

Practical solution is the use of trusted third party, refereed as KDC. A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.

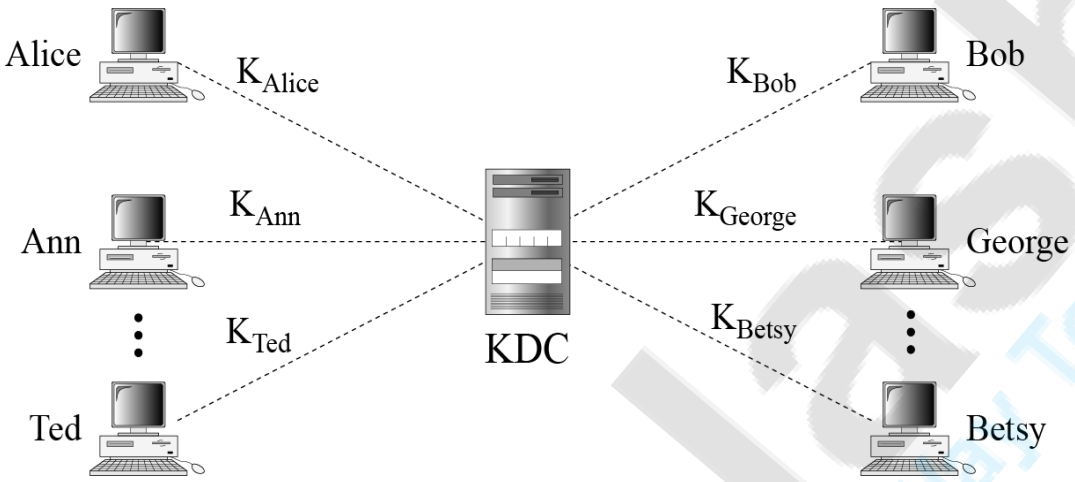


Figure Key-*distribution center (KDC)*

Benefits

- Easier key distribution
- Scalability

Drawbacks

- A KDC can become a single point of failure
- Everybody must trust the KDC
- Vulnerable to replay attack

Features

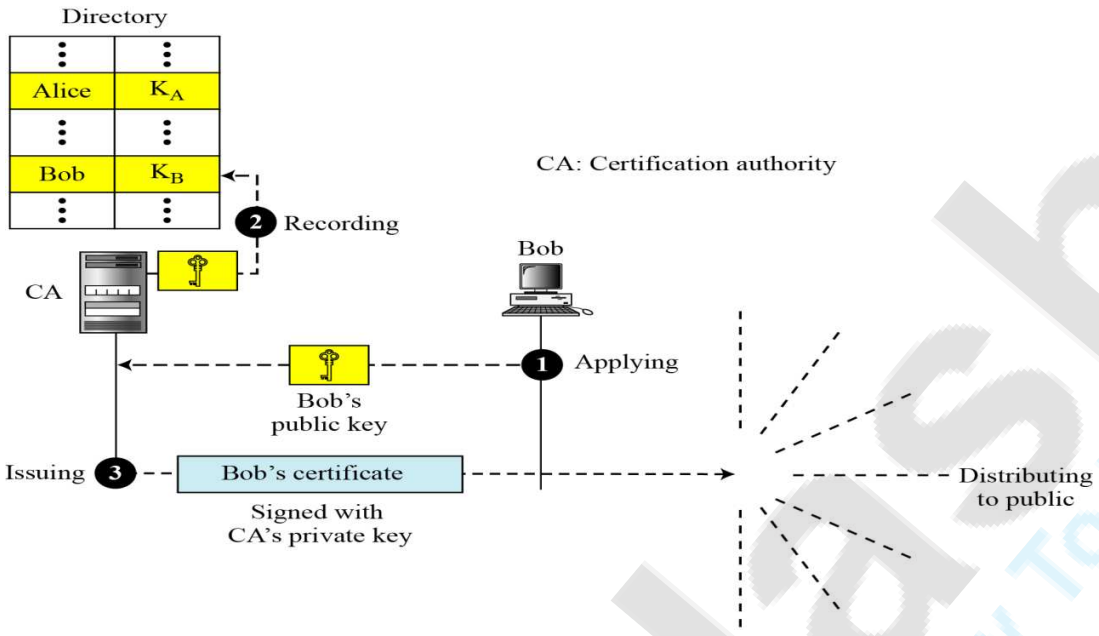
1) All users' secret keys in one database.

2) Must be available all time.

3) Need replication for performance, availability.

Public-Key Distribution:

In asymmetric-key cryptography, people do not need to know a symmetric shared key; everyone shields a private key and advertises a public key.



Features

1) Simple, need not be online all time.

2) High processing complexity.

CA is preferred over KDC because of various reasons.

1) CA does not need to be online.

2) Since CA does not have to be on-line or perform network protocols, it can be vastly simpler device, and therefore it might be more secure.

3) If CA were to crash, the network would not be disabled as would be the case with KDC.

4) A compromised CA cannot decrypt conversations, whereas a compromised KDC can decrypt the conversation.

5) Certificates are not security-sensitive, since only the CA can generate signatures.

Q 27) Explain Smart Cards

Ans: A Better form of authentication token is the smart card. This is a device about the size of the credit card but with an embedded CPU and memory When inserted in a smart reader , the card carries on a conversion with a device.

There are various forms of smart cards:

1) PIN protected memory card
2) Cryptographic challenge/ response cards
3) Cryptographic calculator

1) PIN protected memory card:
With this card, there is information in the memory of the card that can only be read after a PIN is input to the card.

Usually , after some number of wrong PIN guesses , the card "locks" itself and will not give the information to anyone . Information stored on such a smart card is safer than that stored on a magnetic strip card because a stolen card is useless

Without the PIN . These cards are more difficult to duplicate than magnetic strip cards, but it is still possible given the PIN.

2) Cryptographic challenge/response cards:

With this card, there is a cryptographic key in memory , the card is willing to encrypt or decrypt using the key but will not reveal the key even after the PIN is entered. A computer that know the key in the card can authenticate the user by creating a random challenges and "challenging" the card to encrypt or decrypt it. If the correct answer is returned , the computer can have the confidence that the smart is present and the correct PIN was entered. These cards can be constructed so as to be nearly impossible to duplicate or to extract the key from.

Since there is no way to directly extract the key , it can only be done by disassembling or inserting probes into the card. There is a reciprocating escalation in the technologies for probing the card and for packaging it to be unreadable. For most pratical purposes the cards are unreadable. Like keys and magnetic strip cards , the serious pratical problems with smart cards are the need for readers at every access point and the need for recovery wheb a card is lost or forgotten. The cryptographic card offers substantial protection againt eavesdropping.

3) Cryptographic calculator(sometimes called a readerless smart card):

A cryptographic calculator is like a smart card. In that it performs cryptographic calculations using a key that it will not disclose. It is unlike a smart card in that it requires no electrical connection to the terminal.

It has a display and usually a keyboard , and all interaction is through the user. One way it could work is by stimulating a smart card . The user enters a PIN to unlock the device; the computer wishing to authenticate the user generate a random challenges and displays it to the users , the users types it into calculator: the calculator encrypts the values and display the result ; the user enters the result on the terminal ; the computer does the same calculation and compares the results. An alternative protocol that cuts the typing in half is for the calculator to encrypt the current time and display the result . The user types in this number in places of a password . There's a little more work for the computer since it will not be sure of the exact time that the calculator thinks it is ; it will have to do the calculation on several candidate times values to verify what the device said .It might then record the accumulated clock skew to make the next such calculation easier. It's possible to eliminate the keyboard from the calculator by having a PIN or password sent to the computer instead. It is important to have some form of PIN to prevent someone who steals the calculator from impersonating its owner . In addition to saving typing, another advantage of the time encryption protocol is that it fits the "form factor" of protocols designed for passwords . If a protocol has a "password" field and no way for the authenticating application to send a challenge , the encrypted time variant can still be made to work.

Biggest advantage of these readerless smart cards is that they can be used from ordinary terminals with no special hardware. This popularity is growing among companies that want to let their employees log in from home using laptops and modems but are afraid of opening their networks up to intruders.

Q28)What is mutual authentication? What is reflection attack, Suggest method for fixing this?

Ans: **MUTUAL AUTHENTICATION:** When before communication the sender and receiver both authenticate each other and then start the actual communication then it is called Mutual authentication. Mutual authentication can be implemented in three different ways.

I. Shared secret

II.Public keys

III.Timestamps-based

I.**SHARED SECRET:**This protocol assumes that A and B have a shared symmetric key KAB.The protocol works as follows.

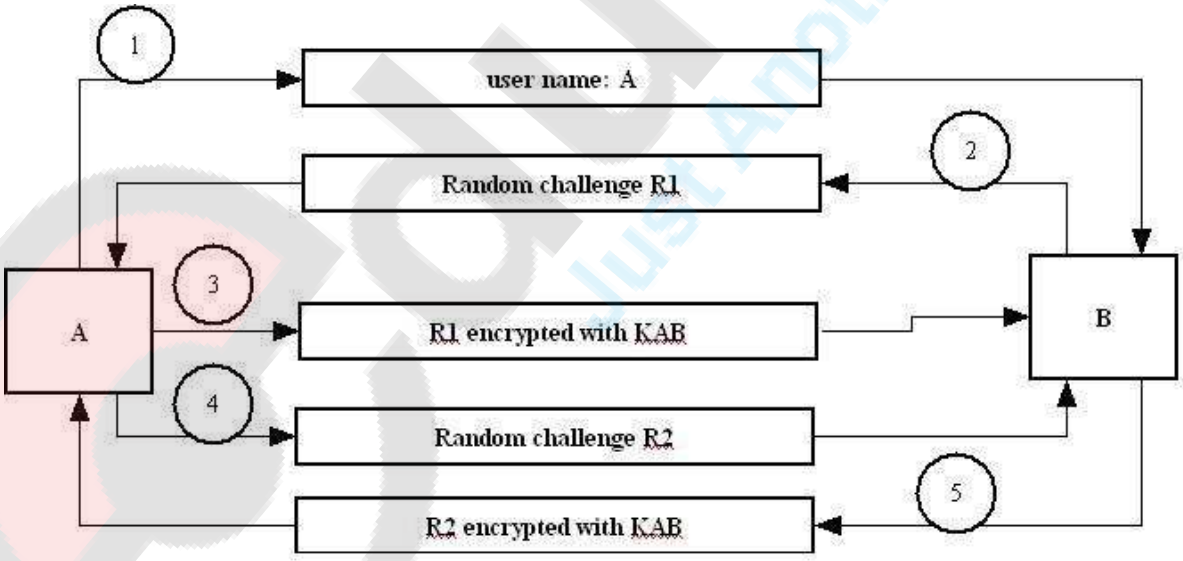
1.A sends her user name to B.

2.B sends a random challenge R1 to A.

3.A encrypts R1 with KAB and sends to B.

4.Asends a different random challenge R2 to B.

5.B encrypts R2 with KAB and sends it to A.



Now ,B authenticates A by step 2 and 3 and A authenticates B in step 4 and 5.Thus mutual authentication takes place.

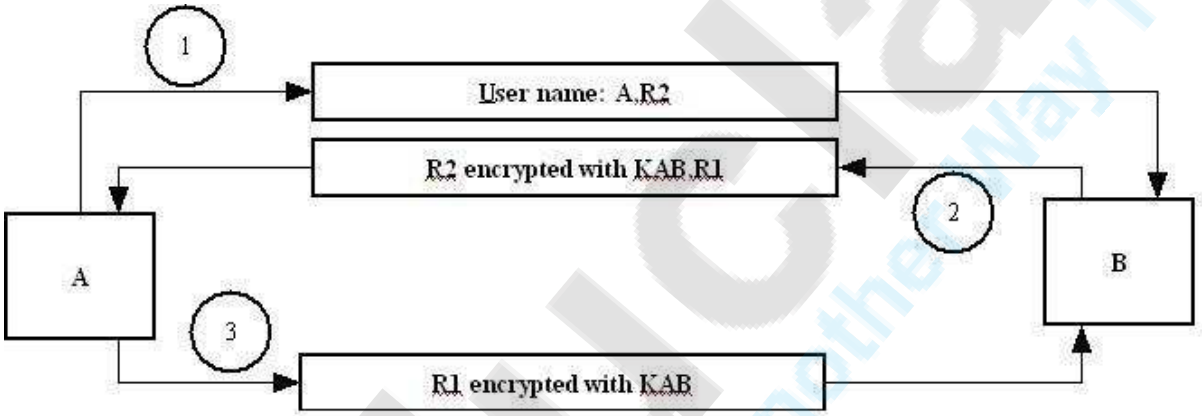
We can see that too many messages are exchanged ,making this protocol inefficient.We can reduce this messages ,by putting more information in those three messages.This modified approach is described as follows:

1.A sends the user name and a random challenge R2 to B.

2.B encrypts R2 with the shared symmetric key KAB,generates a new random challenge R1 and sends these two to A.

3.A verifies R2 ,encrypts R1 with the shared key KAB and sends it to B ,B verifies R1.

This process is shown in the figure below.



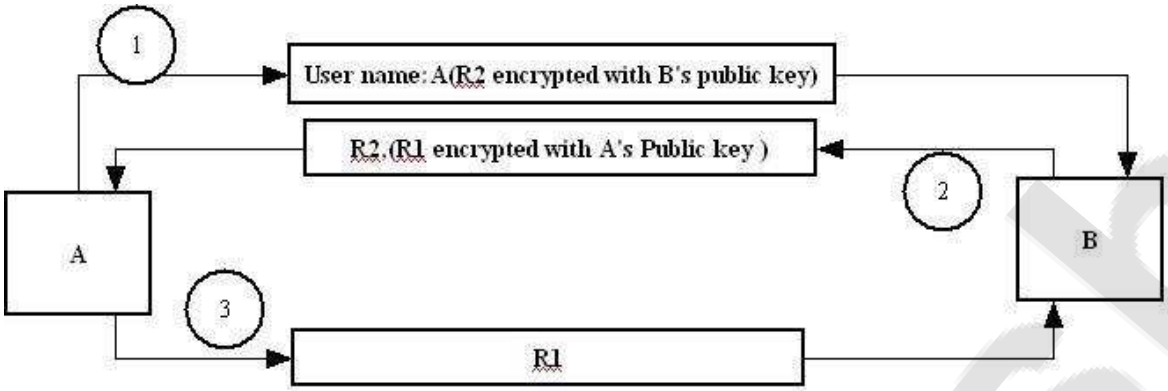
II.**PUBLIC KEYS:** Mutual authentication can also be accomplished by using the public key technology.If A and B know each other's public key,three messages are required to complete the mutual authentication process, as follows

1.A sends her user name and a random challenge (R2) encrypted with B's public key.

2.B decrypts the random challenge (R2) with his private key.B creates a new random challenge (R1) and encrypts it with A's public key .B sends these two things (decrypted R2 and encrypted R1) to A.

3.A decrypts the random challenge (R1) with her private key and sends it to B.B verifies R1.

This process is shown in fig below



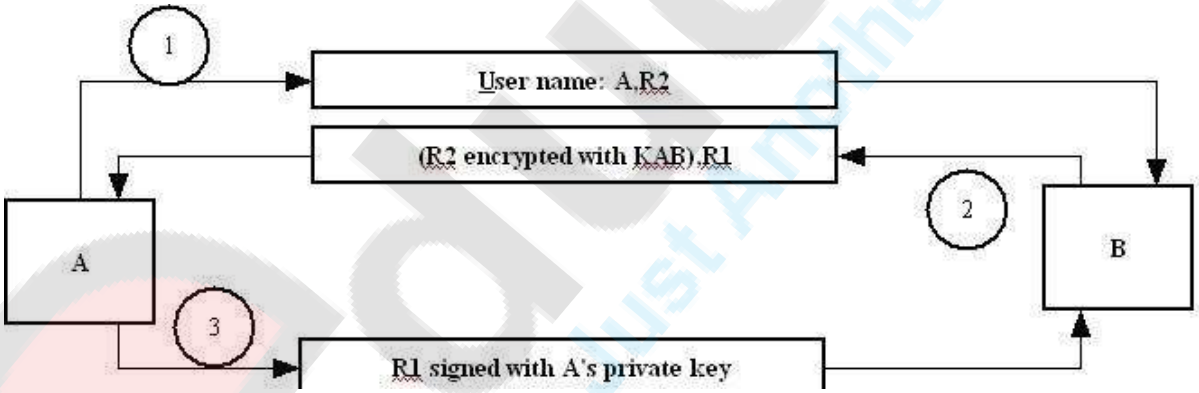
As usual we have a variation of this scheme, where:

1.A sends her user name and R2 to B.

2.B encrypts R2 with his private key and sends it and R2 to A.

3.A signs R1 and returns it back to A.

This is shown in fig below

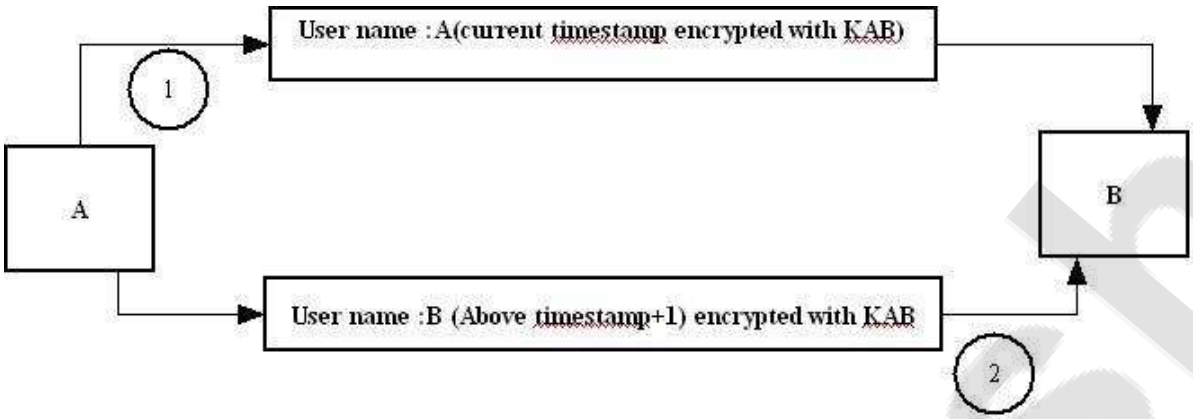


III.**TIMESTAMPS :**We can reduce the mutual authentication process to just two steps by using timestamps, instead of random numbers as challenges. This would work as follows:

1.A sends her user name and the current timestamp encrypted with a shared symmetric key(KAB) to B.

2.B retrieves the timestamp by decrypting the block using KAB and adds one to the timestamp encrypts the result with KBA (not KAB!) and sends it to A,along with his user name.

This approach is shown in fig below

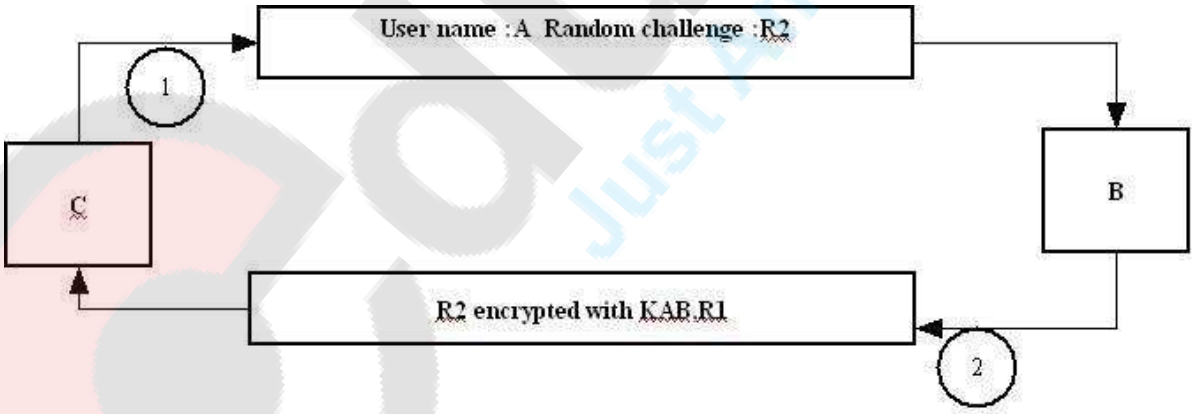


REFLECTION ATTACK: The reduced version of shared key for mutual authentication suffers from a problem called reflection attack.Suppose that attacker C wants to pose as A to B .First ,the attacker C starts the protocol as follows:

1.C sends a message to B containing user id of A and Random challenge R2.

2.B encrypts R2 with the shared symmetric key KAB ,generates a new random challenge R1, and sends these two to C.B thinks he is sending these to A.

This is shown in fig below.

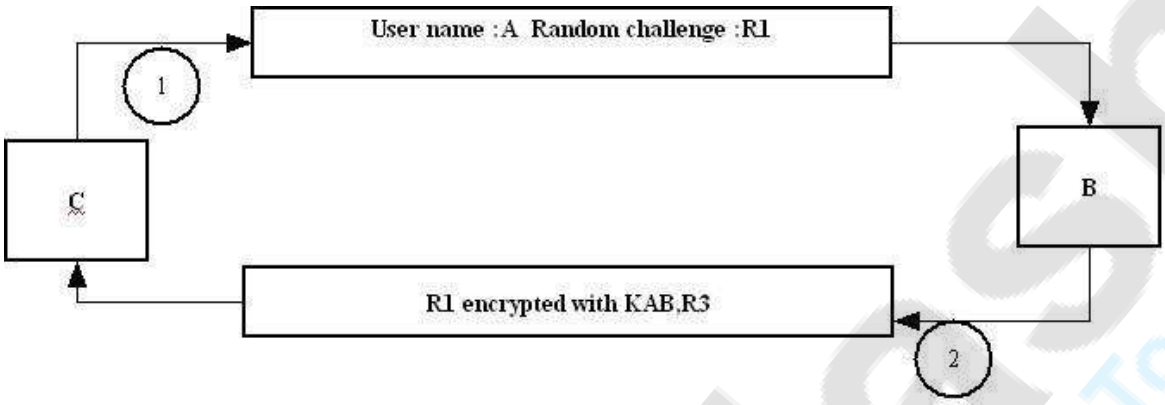


The attacker C cannot encrypt R1 with KAB .However ,she has managed to have B encrypt R2.Now the attacker C opens a second session with B ,distinct from the first session,which is still active. Now the following happens.

1.C sends a message to B containing user id of A and random challenge R1.

2.B encrypts R1 with the shared symmetric key KAB ,generates a new random challenge R3 and sends these two to C.B thinks that he is sending these to A.

This is shown below in fig.



The attacker C cannot proceed with this second session, since she cannot encrypt the new random challenge R3.however,she need not anyways proceed with this session. Instead, she can go back to her first session opened with B earlier. Remember she could not encrypt r1 with KAB in that session, and was hence waiting? Now C has R1 encrypted with KAB thanks to this second session. She sends it to B and completes authentication!

This is shown in fig below

Thus C is able to convince B that she is A!

This problem of Reflection attack can be handled by using two different keys KAB and KBA.KAB should be used when A wants to encrypt something and send it to B .KBA should be used in the other direction that is used by B when he wants to encrypt something. Therefore, B will not be able to encrypt R1 using KAB .This means that C cannot misuse it later, as happens in the case of the reflection attack.

Q 29) Explain Digital Signatures

DIGITAL SIGNATURE

A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged. Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact. A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real.

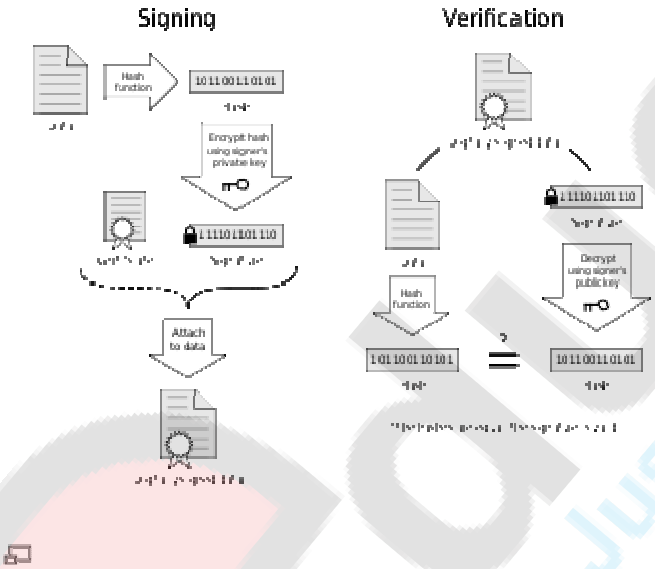


Diagram showing how a simple digital signature is applied and then verified

How It Works

Assume you were going to send the draft of a contract to your lawyer in another town. You want to give your lawyer the assurance that it was unchanged from what you sent and that it is really from you.

1. You copy-and-paste the contract (it's a short one!) into an e-mail note.
2. Using special software, you obtain a message hash (mathematical summary) of the contract.

3. You then use a private key that you have previously obtained from a public-private key authority to encrypt the hash.
4. The encrypted hash becomes your digital signature of the message. (Note that it will be different each time you send a message.)

At the other end, your lawyer receives the message.

1. To make sure it's intact and from you, your lawyer makes a hash of the received message.
2. Your lawyer then uses your public key to decrypt the message hash or summary.
3. If the hashes match, the received message is valid.

Uses of digital signatures

As organizations move away from paper documents with ink signatures or authenticity stamps, digital signatures can provide added assurances of the evidence to provenance, identity, and status of an electronic document as well as acknowledging informed consent and approval by a signatory. The United States Government Printing Office (GPO) publishes electronic versions of the budget, public and private laws, and congressional bills with digital signatures.Below are some common reasons for applying a digital signature to communications:

Authentication

Although messages may often include information about the entity sending a message, that information may not be accurate. Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user. The importance of high confidence in sender authenticity is especially obvious in a financial context. For example, suppose a bank's branch office sends instructions to the central office requesting a change in the balance of an account. If the central office is not convinced that such a message is truly sent from an authorized source, acting on such a request could be a grave mistake.

Integrity

In many scenarios, the sender and receiver of a message may have a need for confidence that the message has not been altered during transmission. Although encryption hides the contents of a message, it may be possible to *change* an encrypted message without understanding it. (Some encryption algorithms, known as nonmalleable ones, prevent this, but others do not.) However, if a message is digitally signed, any change in the message after signature will invalidate the signature. Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions

Non-repudiation

Non-repudiation, or more specifically *non-repudiation of origin*, is an important aspect of digital signatures. By this property an entity that has signed some information cannot at a later time deny having

signed it. Similarly, access to the public key only does not enable a fraudulent party to fake a valid signature. This is in contrast to symmetric systems, where both sender and receiver share the same secret key, and thus in a dispute a third party cannot determine which entity was the true source of the information.

Q 30) What is Certificate Revocation? What are the broad level differences between CRL, OCSP and SCVP?

Ans:

Digital Certificate:

The key exchange or key agreement suffers from the man-in-the-middle attack. This problem is resolved with the use of Digital Certificate.

The digital certificate is issued by Certificate Authority(CA).

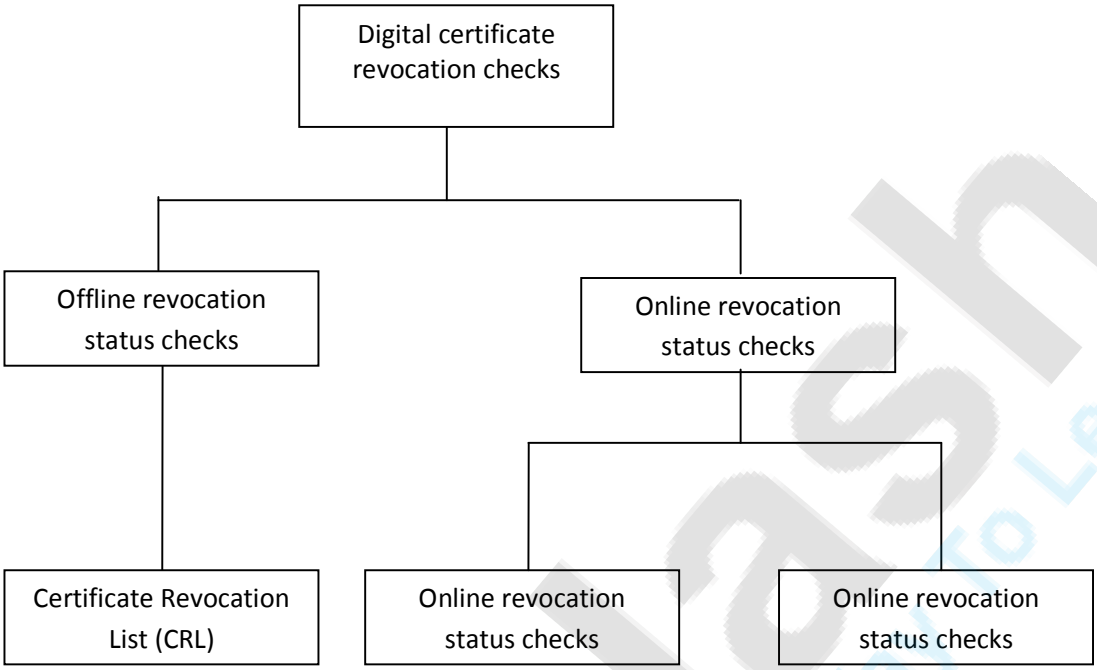
Certificate Revocation:

A digital certificate can be revoked like in situations like cancelling the credit card when it is stolen or lost. The reasons for the revocation of a digital certificate are as follows:

- The holder of the digital certificate reports that the private key corresponding to the public key specified in the digital certificate is compromised(i.e. someone has stolen it)
- The CA realizes that it has made some mistake while issuing a certificate.
- The certificate holder leaves a job and the certificate was issued specifically while the person was employed in that job.

In such cases, the digital certificate of the user must be treated as invalid. For this purpose, the certificate must be revoked. If the user leaves the organization or indulges in an illegal at because of which the certificate needs to be revoked, the organization should initiate the process. In any of the above cases, the CA must come to know about the certificate revocation process. Also, the CA must authenticate the certificate revocation requester before accepting the revocation request. Otherwise, someone can misuse the certificate revocation process to potentially request for the revocation of a certificate that belongs to another user.

The facilities provided by the CA for checking the validity of the digital certificate are shown in the diagram below:



Offline Certificate Status Checks:

Certificate Revocation List (CRL):

- It is the primary means of checking the status of the certificate offline.
- In its simplest form, a CRL is a list of certificates published by each CA, identifying all the certificates whose validity period s over. A CRL lists only those certificates whose validity period is still within the acceptable range, but they are revoked for some other reason.
- Each CA issues its own CRL. The respective CA signs each CRL. Therefore, a CRL can be easily verified.
- The CRL is a sequential file that grows over time to include all the certificates that have not expired, but have been revoked. It is a superset of all previous CRLs issues by that CA.
- Each CRL entry lists the certificate serial number, the date and time on which the certificate was revoked and the reason behind the revocation.
- At the top level, the CRL also includes the information such as date and time this CRL was published and when the next CRL will be published.
- When Alice receives Bob's certificate and wants to see if she should trust it, she should do the following in the given sequence.
    1. Certificate expiry check: compare the current date with the validity period of the certificate to ensure that the certificate has not expired.

2. Signature check: check that Bob's certificate can be verified in terms of the signature by his CA.
3. Certificate revocation check: consult the latest CRL issued by Bob's CA to ensure that Bob's certificate is not listed there as a revoked certificate.

- CRL can become really large overtime. To avoid this problem, the concept of Delta CRL was introduced.
- In this, initially a CA can send a one-time full up-to-date CRL to the users who want to use the CRL services(this is the Base-CRL). At the time of next update, the CA simply issues changes (called Delta) to the CRL since the last update. This mechanism reduces the file size and therefore, its transmission becomes easier.

Online Certificate Revocation Status Checks:

Realizing that CRL may not always be the best way to check the revocation of certificates because of its size as well as of its likelihood of being stale, 2 new protocols were developed- Online Certificate Status Protocol (OCSP) and Simple Certificate Validation Protocol (SCVP).

Online Certificate Status Protocol (OCSP):

- It can be used to check if a given digital certificate is valid at a particular moment.
- It is an online check.
- It allows the certificate validators to check for the status of the certificates in real time. This provides a quicker, simpler and more efficient mechanism for digital certificate validation.

The working of OCSP is as follows:

1. CA provides a server, called OCSP Responder. This server contains the latest certificate revocation information. The requestor has to send a query about a particular certificate. Mostly, HTTP is used but, sometimes the OCSP provides its own protocol
2. The OCSP responder consults the server's X.500 directory to see if the particular certificate is valid or not.
3. Based on the result of the status check form X.500, the OCSP responder sends back a digitally signed OCSP Response for each of the certificates in the original request to the client. The response can take of the 3 forms, namely, Good, Revoked or Unkown. Generally, the recommendation is to consider the certificate as valid if the OCSP response is Good.

Simple Certificate Validation Protocol (SCVP):

- It is the draft stage as of the current writing.
- It is an online certificate status reporting protocol.
- It is designed to deal with the drawbacks of OCSP like- OCSP does not check the validity of the chain of certificates associated with the current certificate.
- Conceptually SCVP is similar to OCSP.

Difference between OCSP and SCVP

Point	OCSP	SCVP
1. Client request	The client sends just the certificate serial number to the server.	The clients sends the entire certificate to the server. Consequently, the server can perform many more checks
2. Chain of trust	Only the given certificate is checked.	The client can provide a collection of the intermediate certificates which the server can check.
3. Checks	The only check is whether the certificate is revoked or not.	The client can request for additional checks, type of revocation information to be considered etc.
4. Returned information	Only the status of the certificate is retuned by the server.	The client can specify what additional information it is interested in.
5. Additional features	None	The client can request for a certificate to be checked for a backdated event.

The main difference between CRL and OCSP & SCVP is that CRL is an Offline Check whereas OCSP and SCVP are online checks.

Q 31) What is authentication and the different methods by which authentication can be done?Expalin types .

Ans: Authentication is any process by which you verify that someone is who they claim they are. This usually involves a username and a password, but can include any other method of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints. Authentication is equivalent to showing your drivers license at the ticket counter at the airport.

Authorization is finding out if the person, once identified, is permitted to have the resource. This is usually determined by finding out if that person is a part of a particular group, if that person has paid admission, or has a particular level of security clearance. Authorization is equivalent to checking the guest list at an exclusive party, or checking for your ticket when you go to the opera.

Different Authentication Methods :

Introduction: Authentication can be accomplished in many ways. The importance of selecting an environment appropriate Authentication Method is perhaps the most crucial decision in designing secure systems. Authentication protocols are capable of simply authenticating the connecting party or authenticating the connecting party as well as authenticating itself to the connecting party. This overview will generalize several Authentication Methods and Authentication Protocols in hopes of better understanding a few options that are available when designing a security system.

Passwords: Passwords are the most widely used form of authentication. Users provide an identifier, a typed in word or phrase or perhaps a token card, along with a password. In many systems the passwords, on the host itself, are not stored as plain text but are encrypted. Password authentication does not normally require complicated or robust hardware since authentication of this type is in general simple and does not require much processing power. Password authentication has several vulnerabilities, some of the more obvious are:

- Password may be easy to guess.
- Writing the password down and placing it in a highly visible area.
- Discovering passwords by eavesdropping or even social engineering.
- The risk of eavesdropping can be managed by using digests for authentication. The connecting party sends a value, typically a hash of the client IP address, time stamp, and additional secret information. Because this hash is unique for each accessed URI, no other documents can be accessed nor can it not be used from other IP address without detection. The password is also not vulnerable to eavesdropping because of the hashing.

The system is, however, vulnerable to active attacks such as the-man-in-the middle attack.

One-time passwords: To avoid the problems associated with password reuse, one-time passwords were developed. There are two types of one-time passwords, a challenge-response password

and a password list.

The challenge-response password responds with a challenge value after receiving a user identifier. The response is then calculated from either the response value (with some electronic device) or select from a table based on the challenge.

A one-time password list makes use of lists of passwords which is sequentially used by the person wanting to access a system. The values are generated so that it is very hard to calculate the next value from the previously presented values. For example, the S/Key system calculates values xi starting from initial value R: x1=f(R), x2=f(f(R)), ..., xn=f(xn-1).

The f() is chosen so that f -1(f raise to -1) is very difficult. First the xn is used, then the xn-1

is used.

It is important to keep in mind that Password systems only authenticate the connecting party. It does not provide the connecting party with any method of authenticating the system they are accessing, so it is vulnerable to spoofing or a man-in-middle attack.

Secure Sockets Layer

Secure Sockets Layer (SSL), developed by Netscape Communications, provides a secure method of communication for TCP connections, especially for HTTP connections. SSL work in this manner: after a TCP connection is established, the client sends a client hello message to which the server responds with a server hello message. The hello messages establish connection attributes which include the protocol version, a session identifier, the cipher suite used, and the compression method in addition to random values for both the server and the client. After the hello messages are exchanged, the server will send its certificate. When the server has been authenticated, depending on the cipher suite used, the server may then request a certificate from the client. After receiving the client hello, the server instructs the client to start using encryption and finishes the initial handshake. The application transfer can now take place.

When the client and the server decide to resume a previous session or duplicate an existing session, only the hello messages are exchanged. If the server does not find a matching session identifier, it will assume the connection is a new one. The advantage of resuming previous session is that it saves processing time, which may have a considerable effect on server performance.

Secure Shell

Secure Shell (SSH) is a protocol for providing secure remote login and other secure network services over an insecure network. With SSH (version 2) each host has a host key, during the connection establishment the client can verify he is talking to the right server. The server keys can be stored locally on the clients or they may be distributed by using a key distribution protocol. After a reliable byte stream is established between the client and the server, host authentication takes place using the transport layer functions. Both ends send version identification.

The key exchange begins with both the client and server sending a key exchange initialization packet. The initialization packet contains a list of algorithms for key exchange, keys, encryption, MAC, and the level of compression supported. The server and client may negotiate a different set of algorithms for each direction of data flow. For each category, the best algorithm is chosen that both the client and server support.

There are two types of authentication methods, one which is referred to as a **digital signature** (or plainly: signature), the other one often as **message authentication code** (MAC). These types can be described as follows:

- **Signatures:** A signature is an authentication on a document that can be verified by anyone **[1]** using the public key of the signer, the message signed, and the signature on the message. It is necessary to have the secret key corresponding to the public key in order to compute the signature on a given message. Signatures can be transferred, i.e., their validity can be checked by anybody, and they are therefore useful for contracts, receipts, etc. Signatures are characterized by being long (e.g., 1024 bits), and are not very quick to produce and verify, and for these reasons, should only be used when the functionality MACs do not offer is needed.
- **MACs:** A message authentication code corresponds to a short and quickly generated/verified **non-transferable** signature on a document. Since it cannot be transferred (i.e., verified by a participant other than the one it was intended for) it cannot be used for contracts or receipts (if these need to be saved in case of a conflict.) but can be used for participants to make sure that the message they obtain is from the person they expect. Since they are very efficient, this makes them very useful for individual, small messages in interactive protocols. Here, all of these messages can later be signed if a receipt is needed. MACs require that the sender and the receiver of the authenticated message both know a (symmetric) secret that is used both for generating and verifying the MAC. This secret can be produced by one of the participants, and sent over in an encrypted form to the other, using a public key encryption method. MAC's can be implemented using stream ciphers, e.g., RC5.

Q 33) Explain Stenography

Ans: Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity.

The advantage of steganography, over cryptography alone, is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

Steganographic techniques

Physical steganography

- Hidden messages on paper written in secret inks,
- Messages written on envelopes in the area covered by postage stamps.

Digital steganography

Digital steganography techniques include:

- Concealing messages within the lowest bits of noisy images or sound files.
- Concealed messages in tampered executable files, exploiting redundancy in the targeted instruction set.

Network steganography

All information hiding techniques that may be used to exchange steganograms in telecommunication networks can be classified under the general term of network steganography.

Network steganography covers a broad spectrum of techniques, which include, among others:

- Steganophony - the concealment of messages in Voice-over-IP conversations, e.g. the employment of delayed or corrupted packets that would normally be ignored by the receiver (this method is called LACK - Lost Audio Packets Steganography), or, alternatively, hiding information in unused header fields

Text steganography

Steganography can be applied to different types of media including text, audio, image and video etc. However, text steganography is considered to be the most difficult kind of steganography due to lack of redundancy in text as compared to image or audio but still has smaller memory occupation and simpler communication.

34) Expalin Biometric Authentication

Ans:-

Defination

Electronic identification of an individual on the basis of his or her unique biological or physiological characteristics(together called Biometric Signature) such as facial features, fingerprints, hand geometry, retinal patterns, voiceprint.

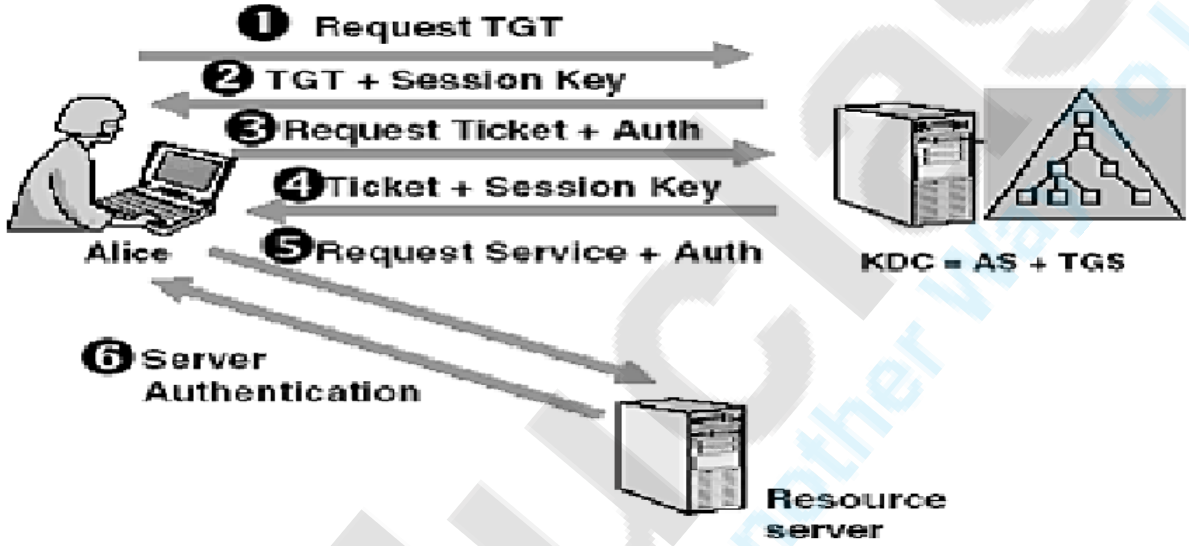
Explanation

- **Face recognition**
  Face recognition systems work by systematically analyzing specific features that are common to everyone's face - the distance between the eyes, width of the nose, position of cheekbones, jaw line, chin and so forth. These numerical quantities are then combined in a single code that uniquely identifies each person.
- **Fingerprint identification**
  Fingerprints remain constant throughout life. In over 140 years of fingerprint comparison worldwide, no two fingerprints have ever been found to be alike, not even those of identical twins. Good fingerprint scanners have been installed in PDAs like the iPaq Pocket PC; so scanner technology is also easy. Might not work in industrial applications since it requires clean hands.
  Fingerprint identification involves comparing the pattern of ridges and furrows on the fingertips, as well as the minutiae points (ridge characteristics that occur when a ridge splits into two, or ends) of a specimen print with a database of prints on file.
- **Hand geometry biometrics**
  Hand geometry readers work in harsh environments, do not require clean conditions, and forms a very small dataset. It is not regarded as an intrusive kind of test. It is often the authentication method of choice in industrial environments.
- **Retina scan**
  There is no known way to replicate a retina. As far as anyone knows, the pattern of the blood vessels at the back of the eye is unique and stays the same for a lifetime. However, it requires about 15 seconds of careful concentration to take a good scan. Retina scan remains a standard in military and government installations.
- **Iris scan**
  Like a retina scan, an iris scan also provides unique biometric data that is very difficult to duplicate and remains the same for a lifetime. The scan is similarly difficult to make (may be difficult for children or the infirm). However, there are ways of encoding the iris scan biometric data in a way that it can be carried around securely in a "barcode" format. (See the SF in the News article Biometric Identification Finally Gets Started for some detailed information about how to perform an iris scan.)
- **Signature**
  A signature is another example of biometric data that is easy to gather and is not physically intrusive. Digitized signatures are sometimes used, but usually have insufficient resolution to ensure authentication.
- **Voice analysis**
  Like face recognition, voice biometrics provide a way to authenticate identity without the subject's knowledge. It is easier to fake (using a tape recording); it is not possible to fool an analyst by imitating another person's voice.

Q 35) Explain Kerberos and its working? Explain Kerberos V5.How does V4 differ from V5?

Ans. Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It provides a centralized authentication server whose function is to authenticate users to servers and servers to users. Kerberos protocol messages are protected against eavesdropping and replay attacks. Kerberos is based on the secret-key distribution model that was originally developed by Needham & Schroeder. The first three versions are no longer in use, at present versions 4 and 5 are used.

Working-



There are four parties involved in the Kerberos protocol:

- Alice: The client workstation.
- Authentication server (AS): Verifies (authenticates) the user during login.
- Ticket granting server (TGS): Issues tickets to certify proof of identity.
- Bob: the server offering services such as network printing, file sharing or an application program.

The job of AS is to authenticate every use at the login time. AS shares a unique secret password with every user. The job of TGS is to certify to the services in the network that user is what she claims to be. For proving this, the mechanism of tickets is used.

Steps in the Kerberos protocol

Step 1: Login

To start with, Alice, the user enters her name at a public work station. The work station sends her name in plain text to the AS.

In response, the AS performs several actions. It first creates the package of the user name and a randomly generated session key (KS). It encrypts this message with the symmetric key that the AS shares with the TGS. The output of this step is called as the Ticket Granting Ticket (TGT). The AS then combines the TGT with the session key (KS), and encrypts the two together using a

symmetric key derived from the password of Alice (KA). After this message is received, Alice's workstation asks for the password. When Alice enters it, the workstation generates the symmetric key (KA) derived from the password and uses that key to extract the session key and the Ticket Granting Ticket. The workstation destroys the password of Alice.

Step 2: Obtaining a service granting ticket (SGT)
The user wants to make use of Bob-the email server, for some email communication. For this, Alice would inform her workstation that she needs to contact Bob. Therefore Alice needs a ticket to communicate with Bob. At this juncture, Alice's workstation creates a message intended for the Ticket Granting Server (TGS), which contains the following items:

- The TGT
- The id of the server
- The current timestamp, encrypted with the same session key.

Once the TGS is satisfied of the credentials of Alice, the TGS creates a session key KAB, for Alice to have secure communication with Bob. TGS sends it twice to Alice: once combined with Bob's id and encrypted with the session key and a second time, combined with Alice's id and encrypted with Bob's secret key (KB).

Step 3: User contacts Bob for accessing the server.

Alice can now send KAB to Bob in order to enter into a session with him. Alice forwards KAB encrypted with Bob's secret key to Bob. To guard against replay attacks, Alice also sends the timestamp, encrypted with KAB to Bob. Since only Bob has his secret key, he uses it to first obtain the information. From this it gets the key KAB, which he uses to decrypt the encrypted timestamp value. Bob now adds 1 to the timestamp sent by Alice encrypts the result with KAB and sends it back to Alice. Since only Alice and Bob know KAB, Alice can open this packet and verify that the timestamp incremented by Bob was indeed the one sent by her to Bob in the first place.
Since Alice needs to authenticate or sign on only once, this mechanism is called as Single Sign On(SSO).

<u>Kerberos Version 5</u>

- Version 5 of Kerberos overcomes some of the shortcomings of version 4.
- Version 5 allows the choice of other algorithms.
- The key salt algorithm has been changed to use the entire principal name i.e. the same password will not result in the same encryption key in different realms or with two different principals in the same realms.
- Kerberos V5 uses ASN.1 syntax with the basic encoding rules.
- V5 supports forwardable, renewable and postdatable tickets.
- Kerberos V5 tickets contain multiple ip addresses and addresses for different types of networking protocols.
- A generic crypto interface module is now used, so other encryption algorithm beside DES can be used.
- Supports replay caches, so authenticators are not vulnerable to replay.
- There is support for transitive cross-realm authentication.

V4 differs from V5 in the following ways:

- V4 demands the use of DES, whereas V5 allows flexibility in terms of allowing the choice of other algorithms.
- V4 depends on IP addresses as identifiers, however V5 allows the use of other types as well.
- V4 has a greater installed base, is simpler, and has better performance while V5 has greater functionality.
- Messages have a largely fixed layout with variable-length fields whereas in V5 fields can be of varying length.
- In V4, realms are DNS standard names, whereas in V5 they can be DNS standard names or X.500 names.
- In V4, the maximum lifetime of a ticket was about 21 hours, whereas in V5 tickets can be issued with virtually unlimited lifetimes.
- With V4 there is no authentication of the request to the KDC for a TGT, whereas in V5 preauthentication-data can be sent along with the request which proves that the requester is authentic.
- V4 does not allow credentials issued to one client to be forwarded to some other host and used by some other client. V5 provides this capability.

Q36) Explain various fields in a X.509 digital certificate (PKIX).

Ans: In cryptography, **X.509** is an ITU-T standard for a public key infrastructure (PKI) for single sign-on (SSO) and Privilege Management Infrastructure (PMI). X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm.

Structure of a certificate: The structure foreseen by the standards is expressed in a formal language, namely Abstract Syntax Notation One.

The structure of an X.509 v3 digital certificate is as follows:

Field	Description
Version	Identify a particular version of the X.509 protocol, which is used for this digital certificate. Currently this field can contain 1, 2 or 3.
Certificate Serial Number	Contains a unique integer number, which is generated by the CA.
Signature Algorithm Identifier	Identifies the algorithm used by the CA to sign this certificate.
Issuer Name	Identifies the **Distinguished Name** (DN) of the CA that created and signed this certificate.
Validity	Contains two date-time values (Not before and Not After), which specify the time frame within which the certificate should be considered as valid. These values generally specify the date and time up to seconds or milliseconds.
Subject Name	Identifies the Distinguished Name (DN) of the entity (i.e. the user or the organization) to whom this certificate refers. This field must contain an entry unless an alternative name is defined in version 3 extensions.
Subject Public Key Information	Contains the subject's public keys and algorithms related to that key. This key can never be blank.

37) Explain Packet Sniffing

Ans:-

A **packetsniffer** is a computer program or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network As data streams flow across the network, the sniffer captures each packet and, if needed, decodes and analyzes its content
Explanation

Packets sniffing is difficult to detect, but it can be done. But the difficulty of the solution means that in practice, it is rarely done.

The popularity of packet sniffing stems from the fact that it sees *everything*. Typical items sniffed include:

SMTP, POP, IMAP traffic

Allows intruder to read the actual e-mail.

POP, IMAP, HTTP Basic, Telnet authentication

Reads passwords off the wire in clear-text.

SMB, NFS, FTP traffic

Reads files of the wire.

SQL databse

Reads financial transactions and credit card numbers.

Not only can sniffing read information that helps break into a system, it is an intrusion by itself because it reads the very files the intruder is interested in.This technique can be combined with active transimission for even more effective attacks.

IP spoofingWhen the sniffing program is on a segment between two communicating end points, the intruder can impersonate one end in order to hijack the connection. This is often combined with a denial of service (DoS) attack against the forged address so they don't interfere anymore.raw transmit

Allows abonormal traffic to be generated, such as TCP SYN floods, overlapped fragments, illegal fragments, and TCP fingerprinting. The best attack is severe fragmentation, which fragments the TCP header in order to prevent firewalls from filtering by port number.

Packet sniffing tools are usually written by hackers. There are many extensions for pulling desired data off the network. The most popular are password sniffing programs.

Q 38) Explain in detail Email Security

Ans:

1) Email is one of the most widely used and regarded network services.
   Currently message contents are not secure so they must be inspected either in transit or by suitably privileged users or destination system

2) Email security enhancement:
   - Confidentiality-protect from disclosure.
   - Authentication-of sender of the message.
   - Non-reputation-protection of denial of sender

3) Pretty Good Privacy(PGP):
   - Pgp is widely used de facto secure email.
   - Developed by phil Zimmermann
   - It uses best available cryptography algorithm.
   - Its is available on unix,PC,Macintosh system.
   - It has now commercial versions available.

4) PGP Operation:
   a) Authentication:
   - sender creates a message.
   - SHA-1 used to generate 160-bit hashcode of message.
   - Hashcode is encrypted with RSA using the sender private key and result is attached to message.
   - Reciever generates new hash code for message and compares with decrypted hash code,if match,message is accepted as authentic.

   b) Confidentiality:
   - Sender generates message and random 128-bit number to be used as session key for this message only
   - Message is encrypted, using CAST-128/IDEA/3DES with session key.
   - Session key is encrypted using RSA with recipients public key ,then attached to message
   - Receiver uses RSA with its private key to decrypt and recover session key.
   - Session key is used to decrypt message.

   c) Confidentiality and Authentication:
   - Uses both services on same message

- Create signature and attach to message
- Encrypt both message and signature.
- Attach RSA encrypted session key

d) Compression :

- By default PGP compresses message after signing but before encrypting
- So can store uncompressed message and signature for later verification
- But because compression is non-deterministic
- Uses ZIP compression algo.

e) Email Compatibility:

- When using PGP will have binary data to send
- However email was designed only for text
- Hence PGP must encode raw binary data into printable ASCII characters
- Uses radix-64 algo which maps 3 bytes to 4 printable characters and also appends a CRC
- PGP also segments message if it is too long

5) PGP Session Keys:
   - Needs a session key for each message of varying sizes 56-bit DES, 128-bit CAST or IDEA
   - It is generated using ANSI X12.17 mode
   - Uses random i/ps taken from previous uses and from keystroke timing of uses

6) PGP Public and Private Keys:
   - Since many public/private keys may be in use, you need to identify which is actually used to encrypt session to identify which is actually used to encrypt session key in a message. It could send full public key with every message but this is inefficient.
   - Rather use a key identifier based on key which is least significant 64-bit key and will be unique
   - Also use key ID in signatures

7) S/MIME(Secure/Multipurpose Internet Mail Extension):
   - Security enhancement to MIME email
   - Original internet RFC822 email was text only
   - MIME provided support for varying content types and multi-part messages
   - With encoding of binary data to textual form
   - S/MIME added security enhancements

8) S/MIME Functions:

- Enveloped Data: Encrypted content and association keys
- Signed Data: Encoded message and signed digest
- Clear Signed Data: Clear text message and encoded signed digest
- Signed and Enveloped Data: Nesting of signed and encrypted entities

9) S/MIME Cryptographic Algos:

- Hash Functions: SHA-1 and MD5
- Digital Signnatures: DSS and RSA
- Session Key Encryption: Elgamal and RSA
- Message Encryption: Triple DES, RC2/40

10) S/MIME uses x.509 v3 certificates

- Managed using a hybrid of strict X.509 CA hierarchy and PGP's web of trust
- Each client has a list of trusted CA's certificate
- And own public/private key pairs and certificates
- Certificates must be signed by trusted CA's.

Q 39) Discuss security provided at IP layer in the TCP/IP protocol suite

Ans: Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session.

IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite. It can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host).

IP SEC

The IP Authentication header provides strong authentication and integrity for IP datagrams. Depending on the signing algorithm used, it may also provide non-repudiation, excluding those fields that are changed during transmit, like hop count or time to live. The authentication header has fields for the next header, payload length, security parameters index (SPI: identifies security association (SA) between two hosts), sequence number, and authentication data.

The authentication is transport-protocol independent, so there may be data from more than one different protocol, for instance TCP and UPD. The authentication data is calculated with a message digest algorithm.

To avoid replay attacks, the 32-bit sequence number is not allowed to wrap around; one must establish a new SA and generate new keys. This happens once in 232 packets so, if 1460 byte TCP segments are transferred one can transfer 5.7 TB of data using one SA.

Two levels of Security in this scheme

- Offer security at the IP packet scheme
- Implementing the higher level security mechanisms, depending on the requirements .

Cryptographic algorithms

Cryptographic algorithms defined for use with IPsec include:

- HMAC-SHA1 for integrity protection and authenticity.
- TripleDES-CBC for confidentiality
- AES-CBC for confidentiality.

Q 40) Explain IP security

Ans:

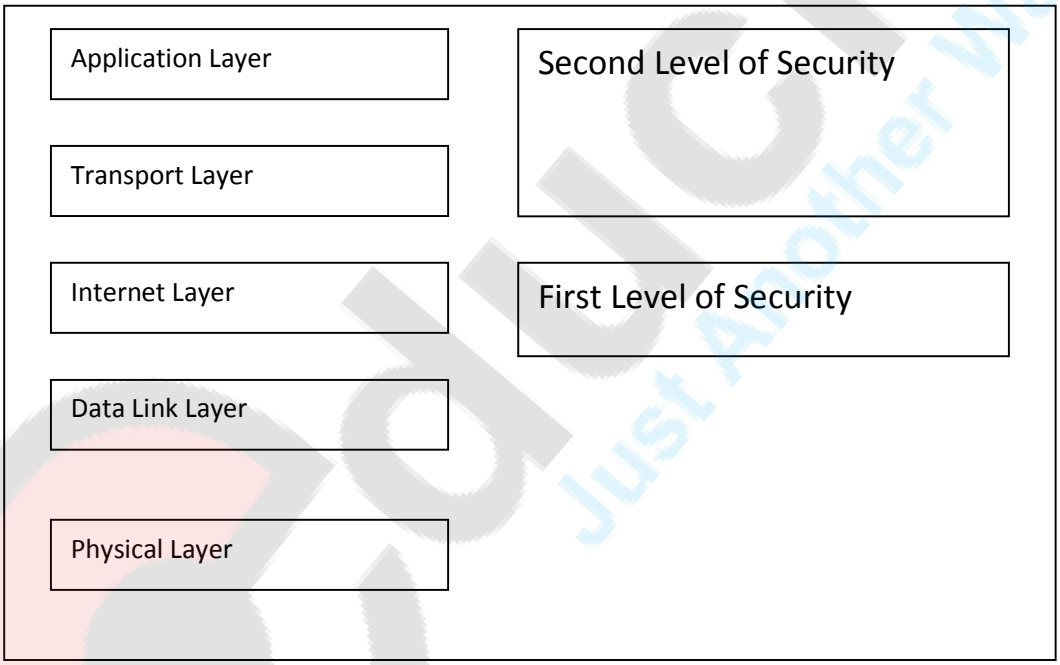
Introduction:

Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol(IP) communications by authenticating and encrypting each IP packet of a data stream. This security mechanism is very important because IP packets are transferred in a plain text format.

Two levels of security in this scheme:

1. Offer security at the IP packet level itself
2. Implementing the higher level security mechanisms depending on the requirements.

Levels of Security:



The Internet Architecture Board (IAB) prepared a report called as security in the Internet Architecture (RFC1636).

The outcome of the report and the study conducted by IAB, in 19995 IETF published five security based standards related to IPsec.

lpv4 supports all the standards mentioned by IETF

Security Architecture:

- Internet Key Exchange (IKE and IKEv2) to set up a security association (SA) by handling negotiations of protocols and algorithms and to generate the encryption and authentication keys to be used by IPsec.
- Authentication Header (AH) to provide connectionless integrity and data origin authentication for Ipdatagrams and to provide protection against replay attacks.
- Encapsulating Security Payload (ESP) to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality.

Security Parameters Index (SPI)

Identifies the security parameters, which, in combination with the IP address, then identify the security association implemented with this packet.

Sequence number

A monotonically increasing number, used to prevent replay attacks.

Authentication data

Contains the integrity check value (ICV) necessary to authenticate the packet, it may contain padding.

Encapsulating Security Payload

This provides data confidentiality. The ESP protocol also defines a new header to be inserted into the IP packet. ESP Processing also includes the transformation of the protected data into an unreadable, encrypted format. Normally the EXP will be inside the AH.

ESP packet format:

<table>
<tr><td>0-7 bit</td><td>8-15 bit</td><td>16-23 bit</td><td>24-31 bit</td></tr>
<tr><td colspan="4">Security parameters index (SPI)</td></tr>
<tr><td colspan="4">Sequence number</td></tr>
<tr><td></td><td colspan="2">Padding(0-255 bytes)</td><td></td></tr>
<tr><td></td><td></td><td>Pad Length</td><td>Next header</td></tr>
<tr><td colspan="4">Authentication data (variable)</td></tr>
</table>

Field meanings:

1. Security parameters index (SPI) → Identifies the security parameters, which, in combination with the IP address, then identify the security association implemented with this packet.
2. Sequence number →A monotonically increasing number, used to prevent replay attacks.
3. Authentication data → Contains the integrity check value (ICV) necessary to authenticate the packet, it may contain padding.
4. Payload data →The data to be transferred.
5. Padding → Used with some block ciphers to pad the data to the full length of the block.
6. Pad lengt →Size of padding in bytes
7. Next Header →Identifies the protocol of the payload data. The value of this field is chosen from the set of IP protocol numbers defined in the most recent "Assigned numbers"
8. Authentication data → Contains the data used to authenticate the packet

Modes of Operation

IPsec can be implemented in a host-to-host transport mode, as well as in a network tunnel mode.

1. Tunnel Mode: In Tunnel Mode, an encrypted tennel is established between two hosts. In this mode IPsec protects the entire IP datagram. It takes an IP datagram, adds the IPsec header and trailer and encrypts the whole thing. Its then adds new IP header to this encrypted datagram.

2. Transport Mode: In Transport mode, only the payload (the data you transfer) of the IP packet is encrypted and/or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however when the authentication header is used, the IP addresses cannot be translated, this will invalidate the hash value. The transport and application layers are always secured by hash, so they cannot be modified in any way.Transport mode is used.

3. Network Management: A network management system comprises:

- Network elements - Sometimes called managed devices, network elements are hardware devices such as computers, routers, and terminal servers that are connected to networks element.
- Managed object – A managed is a characteristic of something that can be managed. For example, a list of currently active TCP circuits in a particular host computer is a managed object. Managed objects differ from variables, which are particular object instances. Using our example, an object instance is a single active TCP circuit in a particular host computer. Managed objects can be scalar (defining a single object instance) or tabular (defining multiple, related instances)
- Management Information Base (MIB) – A MIB is a collection of managed objects residing in a virtual information store. Collections of related managed objects are defined in specific MIB modules.

- Syntax notation - A Syntax notation is a language used to describe a MIB's managed objects in a machine-independent format. Consistent use of a syntax notation allows different types of computers to share information. Internet management systems use a subset of the International Organization for Standardization's (ISO's) Open system Interconnection (OSI) Abstract Syntax Notation (ASN.1) to define both the packets exchanged by the management protocol and the objects that are to be managed.
- Structure of Management Information (SMI) – The SMI defines the rules for describing management information. The SMI is defined using ASN.1
- Network management stations (NMSs) – Sometimes called consoles, these devices execute management applications that monitor and control network elements. Physically, NMSs are usually engineering workstation-caliber computers with fast CPUs, megapixel color displays, substantial memory, and abundant disk space. Atleast one NMS must be present in each managed environment.
- Parties – Newly defined in SNMPv2, a party is a logical SNMPv2 entity that can initiate or receive SNMPv2 communication. Each SNMPv2 party comprises a single, unique party identity, a logical network location, a single authentication protocol, and a single privacy protocol. SNMPv2 messages are communicated between two parties.

This is an extremely simple example using numbers you can workout on a pocket calculator (those of you over the age of 35 can probably even do it by hand)

1. Select primes p=11, q=3.
2. n = pq = 11.3 = 33
   phi = (p-1)(q-1) = 10.2 = 20

3. Choose e=3
   Check gcd(e, p-1) = gcd(3, 10) = 1 (i.e. 3 and 10 have no common factors except 1),

   Check gcd(e, q-1) = gcd(3, 2) = 1

   Therefore gcd(e, phi) = gcd(e, (p-1)(q-1)) = gcd(3, 20) = 1

4. Compute d such that ed = 1 (mod phi)
   i.e. compute d = e^{-1} mod phi = 3^{-1} mod 20

   i.e. find a value for d such that phi divides (ed-1)

   i.e. find d such that 20 divides 3d-1.

   Simple testing (d = 1, 2,..) gives d = 7

   Check: ed-1 = 3.7 – 1 = 20, which is divisible by phi

5. Public key = (n, e) = (33,3)
   Private key = (n, d) = (33,7)

Q41) Distinguish between SSL and SET.
Ans:

Sr .no	Issue	SSL(Secure Socket Layer)	SET (Secure Electronic Transaction)
1.	Main Aim	Exchange of data in an encrypted form.	E-commerce related payment mechanism.
2.	Certification	Two Parties exchange certificates.	All the involved parties must be certified by a trusted third party.
3.	Authentication	Mechanism in place, but not very strong	Strong mechanisms for authenticating all the parties involved.
4.	Risk of merchant fraud	Possible, since customer gives financial data to merchant.	Unlikely, since customer gives financial data to payment gateway.
5.	Risk of customer Fraud	Possible, no mechanisms exist if a customer refuses to pay later.	Customer has to digitally sign payment instructions.
6.	Action in case of customer fraud	Merchant is liable.	Payment gateway is liable.
7.	Practical Usage	High.	Not Much.

Q 42) Discuss SET used in e-commerce.

Ans: SET stands for Secure Electronic Transaction. SET is an open encryption and security specification designed to protect credit card transactions on the Internet. The need for SET came from the fact that MasterCard and Visa realized that for e-commerce payment processing, software vendors were coming up with new and conflicting standards. To avoid all sorts of future incompatibilities, MasterCard and Visa decided to come up with a standard, involving all the major software companies.

SET is not itself a payment system. Rather it is a set of security protocols and formats that enable users to employ their existing credit card payment infrastructure on an open network, such as the Internet, in a secure fashion.

SET services can be summarized as follows:

- It provides a secure communication channel among all the parties involved in an e-commerce transaction.
- It provides authentication by the use of digital certificates
- It ensures confidentiality, because the information is only available to the parties involved in a transaction and that too only when and where necessary.

Participants in the SET system can be summarized as follows:

- Cardholder: Using the Internet, consumers and corporate purchasers interact with the merchants for buying goods and services. A cardholder is an authorized holder of a payment card such as MasterCard or Visa that has been issued by an issuer.
- Merchant: A merchant is a person or an organization that wants to sell goods or services to cardholders. A merchant must have a relationship with an acquirer for accepting payments on the Internet.
- Issuer: The issuer is a financial institution that provides that provides a payment card to the cardholder. The most critical point is that issuer is the ultimately responsible for the payment of the cardholder's debt.
- Acquirer: This is a financial institution that has a relationship with merchants for processing payment card authorizations and payments.
- Payment gateway: The payment gateway processes the payment messages on behalf of the merchant. The payment gateway acts as an interface between SET and the existing credit card payment networks for payment authorizations. The merchant exchanges SET messages with the payment gateway over the Internet. The payment gateway, in turn, connects to acquirers systems using a dedicated network line in most cases.
- Certification Authority: This is an authority that provides public key certificates to cardholders, merchants and payment gateways.

SET Process:

1. Customer opens a credit card account with a bank that supports electronic payment mechanisms and the SET protocol.
2. After verification, customer receives a digital certificate from a CA which also contains details such as the customer's public key and expiration date.
3. Merchant that wants to accept a certain brand of credit cards must possess a digital certificate.
4. Customer places an order.
5. Merchant sends its digital certificate to the customer assuring the customer that he is dealing with a valid merchant.
6. Order and payment details are sent confirming the purchase transaction and giving credit card details. The payment information is so encrypted that the merchant cannot read it. The customer's digital certificate assures the merchant of the customer's identity.
7. Merchant forwards payment details of the customer to the payment gateway via acquirer for payment authorization.
8. Payment gateway verifies customer's credit card details with the help of the issuer and either authorizes or rejects the payment.
9. Assuming payment gateway authorizes the payment, merchant sends confirmation of the order to the customer.
10. Merchant ships the goods and provides ordered services.
11. Merchant sends a request for payment to the payment gateway which handles all of the payment processing.

Major transactions supported by SET:

- Purchase request – comprises of four messages: Initiate Request, Initiate Response, Purchase Request and Purchase Response.
- Payment authorization – comprises of two messages: Authorization Request and Authorization Response.
- Payment capture – comprises of two messages: Capture Request and Capture Response.

Q43] What is firewall ? How does a firewall ensure security of data? Explain different configurations in which firewall can be set up.

Ans-

A] Firewall

A firewall is a computer that sits between your internal network and the rest of the network and attempts to prevent bad things from happening ie it attempt to protect the systems inside from attack from outside.

Need of firewall-

-Most corporate networks, applications are not designed for security previously because they have not been attacked.

-But now users want connectivity to internet for exchanging mails , to share files , to communicate with other people or with publicly available services etc.

-But the internet is not a safe place. There are spies from unfriendly countries , users from competing companies , criminals anxious to steal information for profit etc.

-So firewall is needed because it manage access to services in ways that individual systems should but don't. They can enforce policies such as systems outside the firewall cant access file services on any systems inside the firewall.

B] Working

-Firewall protection works by blocking certain types of traffic between a source and a destination.

-Basically, a firewall, working closely with a router program, examines each network packet to determine whether to forward it toward its destination.

- A firewall also includes or works with a proxy server that makes network requests on behalf of workstation users.

-A firewall is often installed in a specially designated computer separate from the rest of the network so that no incoming request can get directly at private network resources.

-All network traffic has a source, a destination, and a protocol. This protocol is usually TCP, UDP, or ICMP.

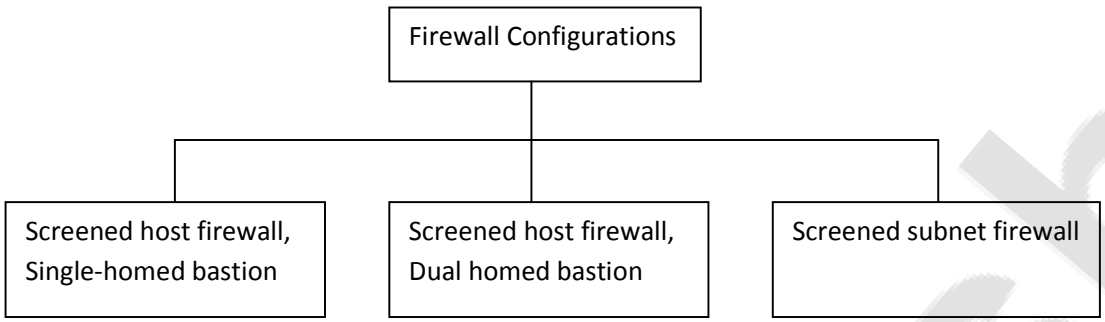
-Firewall protection works by allowing the network security administrator to choose which protocols and ports or message types to allow — and which ones to deny.

Different configurations of Firewalls

Companies such as Cisco and other major vendors have introduced a multitude of firewall products that are capable of monitoring traffic using different techniques. Some of today's firewalls can inspect data packets up to Layer 4 (TCP layer). Others can inspect all layers (including the higher layers) and are referred to as deep packet firewalls.

C] Firewall Configurations

-In practical implementation a firewall is usually a combination of packet filters and application gateways. Based on this there are three possible configurations of firewall ,



1. Screened host firewall, Single-homed bastion

-In this configuration , a firewall set up consist of two parts :

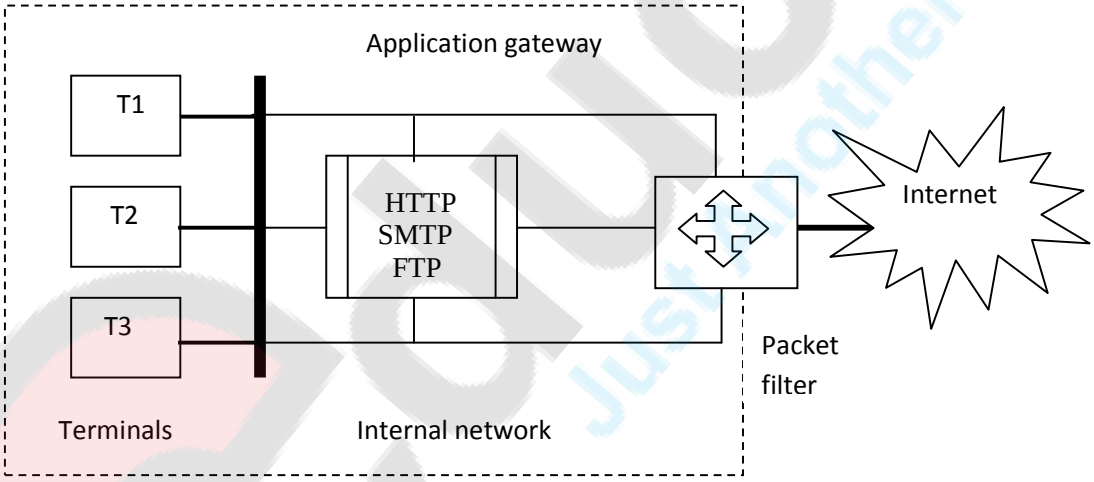
1)Packet filtering router

2)Application gateway

-Their purposes are :

The packet filter ensures that the incoming traffic is allowed only if it is destined for the application gateway , by examining the destination address field of every incoming IP packet. Similarly , it also ensures that the outgoing traffic allows only if it is originating from the application gateway , by examining the source address field of every outgoing IP packet.

The application gateway performs authentication and proxy functions.



Screened host firewall, Single-homed bastion

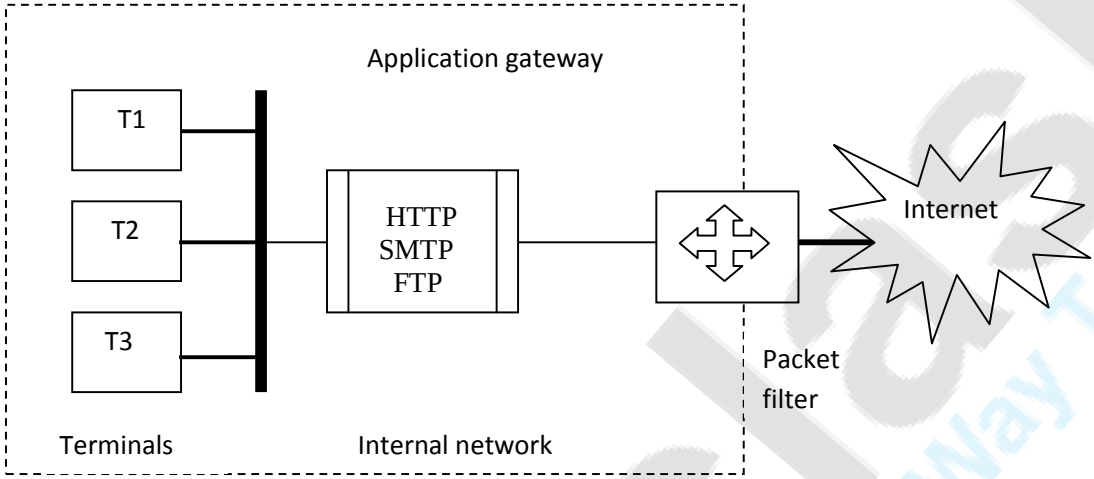
-This configuration increases the security of the network by performing checks at both packets and application levels. This also gives more flexibility to the network administrators to define more granular security policies.

- Disadvantage :

Internal users are connected to the application gateways as well as to the packet filter. Therefore if packet filter is somehow successfully attacked ad its security compromised then the whole internal network is exposed to the attacker.

2. Screened host firewall , dual homed bastion

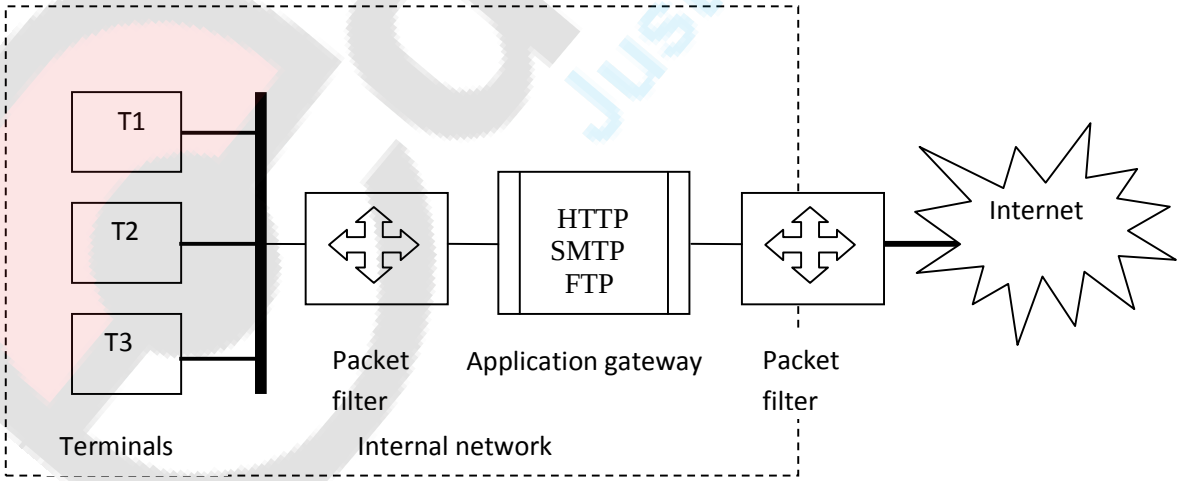
- To overcome drawback of Screened host firewall, Single-homed bastion configuration , this firewall configuration exists.
- Here direct connections between the internal hosts and packet filter are avoided. Instead , the packet filter connects only to the application gateway , which in turn , has a separate connection with the internal hosts.
- Therefore even if the packet filter is successfully attacked , only the application gateway is visible to the attacker.



Screened host firewall , dual homed bastion

3. Screened subnet firewall

- It offers the highest security among the possible firewall configurations.
- Here two packet filters are used , one between the internet and the application gateway and another one between the application gateway and the internal network.
-Now there are three levels of security for an attacker to break into. Attacker doesn't come to know about the internal network , unless he breaks into both the packet filters and the single application gateway standing between them.



Screened subnet firewall

Q 44) What is Firewall? Explain different types of Firewalls.

Ans: The Internet is a vital and growing network that is changing the way many organizations and individuals communicate and do business. Using the internet we can get connected to any other computer, no matter how far the two are located from each otheron the network. However, the Internet suffers from significant and widespread security problems. Many agencies and organizations have been attacked or probed by intruders, with resultant high losses to productivity and reputation. In some cases, organizations have had to disconnect from the Internet temporarily, and have invested significant resources in correcting problems with system and network configuration. Sites that are unaware of or ignorant of these problems face a significant risk that network intruders will attack them. Even sites that do observe good security practices face problems with new vulnerabilities in networking software and the persistence of some intruders. But this facility usually may be a nightmare for network support staff, which is left with a very difficult job of trying to protect the corporate networks from a variety of attacks.

At a broad level, there are two kind of attaks :

1. Most Corporations have large amounts of valuable and confidential data in their networks. Leaking of this critical information to competitors can be a great setback.
2. Apart from the danger of the insider information leaking out, there is a great danger of the outside elements (such as viruses and Worms) entering a corporate network to create havoc.

- Some of the problems with network security are result of inherent vulnerabilities in the services (and the protocols that the services implement) , while others are a result of host configuration and access controls that are poorly implemented or overly complex to administer.
- Additionally, the role and importance of system management is often short changed in job descriptions, resulting in many administrators being, at best, part time and poorly prepared.

Types of Firewalls:

There are several classifications of firewalls depending on where the communication is taking place, where the communication is intercepted and the state that is being traced.

1. Network Layer and Packet Filters:
   Network layer firewalls, also called packet filters, operate at a relatively low level of the TCP/IP protocol stack, not allowing packets to pass through the firewall unless they match the established rule set. The firewall administrator may define the rules; or default rules may apply. The term "packet filter" originated in the context of BSD operating systems.
   Network layer firewalls generally fall into two sub-categories, stateful and stateless. Stateful firewalls maintain context about active sessions, and use that "state information" to speed packet processing. Any existing network connection can be described by several properties, including source and destination IP address, UDP or TCP ports, and the

current stage of the connection's lifetime (including session initiation, handshaking, data transfer, or completion connection). If a packet does not match an existing connection, it will be evaluated according to the ruleset for new connections. If a packet matches an existing connection based on comparison with the firewall's state table, it will be allowed to pass without further processing.

Stateless firewalls require less memory, and can be faster for simple filters that require less time to filter than to look up a session. They may also be necessary for filtering stateless network protocols that have no concept of a session. However, they cannot make more complex decisions based on what stage communications between hosts have reached.

Modern firewalls can filter traffic based on many packet attributes like source IP address, source port, destination IP address or port, destination service like WWW or FTP. They can filter based on protocols, TTL values, netblock of originator, of the source, and many other attributes.

Commonly used packet filters on various versions of Unix are ipf (various), ipfw (FreeBSD/Mac OS X), pf (OpenBSD, and all other BSDs), iptables/ipchains (Linux).

2. Application Layer:

   Application-layer firewalls work on the application level of the TCP/IP stack (i.e., all browser traffic, or all telnet or ftp traffic), and may intercept all packets traveling to or from an application. They block other packets (usually dropping them without acknowledgment to the sender). In principle, application firewalls can prevent all unwanted outside traffic from reaching protected machines.

   On inspecting all packets for improper content, firewalls can restrict or prevent outright the spread of networked computer worms and trojans. The additional inspection criteria can add extra latency to the forwarding of packets to their destination.

Q 45) Explain: a) Viruses, b) Intrusion.

Ans:

a) Viruses: -

- A virus is a piece of software that can infect other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

- A virus can do anything that other programs do. The only difference is that it attaches itself to another program and executes secretly when the host program is run.

- Once a virus is executing, it can perform any function, such as erasing files and programs.

Virus has following four phases:

i) **Dormant Phase:** The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or the capacity of the disk exceeding some limit. Not all viruses have this stage.

ii) **Propagation Phase:** The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

iii) **Triggering Phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events.

iv) **Executing Phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Types of Viruses

- There has been a continuous arms face between virus writers of antivirus software since viruses first appeared.
- As effective countermeasures have been developed for existing types of viruses, new types have been developed.

The various types of virus are shown below.

. Parasitic Virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.

. Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.

. Boot sector virus: Infects a master boot record and spreads when a system is booted from the disk containing the virus.

. Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.

. Polymorphic virus: A virus that mutates with every infection, making detection by the signature of the virus impossible.

. Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection.

b) Intrusion: -

One of the most publicized threats to security is the intruder.

Following are the classes of intruders.

. Masquerader: An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account.

. Misfeasor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges.

. Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

-The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

-Intruder attacks range from the benign to the serious. Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users.

Intrusion Techniques

- The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system.
- Generally, this requires the intruder to acquire information that should have been protected.
- With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

The password file can be protected in one of two ways:

i) **One-way function:** The system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value.

ii) **Access Control:** Access to the password file is limited to or a very few accounts.

Intrusion Detection

A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years. This interest is motivated by a number of considerations, including the following:

i) If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.

ii) An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

iii) Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

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

The following approaches to intrusion detection:

1. Statistical anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time.

a. Threshold detection: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

b. Profile based: A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

2. Rule-based detection: Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

a. Anomaly detection: Rules are developed to detect deviation from previous usage patterns.

b. Penetration identification: An expert system approach that searches for suspicious behavior.

eduClash
Just Another Way To Learn