Name of Teacher: Miss Radhika M. Patil

Class: B.Sc. Computer Science (Entire)- III **Semester : 6**

Course Title: Computer Networks

UNIT III – Session Layer and Presentation Layer

- **Session layer:**

- The **session layer** is also known as a network dialog controller, it creates, maintains, synchronizes the interaction between communicating applications.
- The **session layer** tracks the dialogs between systems, which are also called sessions.
- This layer manages a session by initiating the opening and closing of sessions between end-user application processes.
- It also controls single or multiple connections for each end-user application and directly communicates with both the presentation and the transport layers.
- The services provided by the **session layer** are generally implemented in the application environment using remote procedure calls (RPCs).
- In the **Session layer**, streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.
- A protocol such as Zone Information Protocol, AppleTalk Protocol, and Session Control Protocol are used to implement sessions on Web browsers.
- Through check pointing and recovery session management and restoration are possible using these protocols.
- For example, in live television programs sessions are implemented, in which the audio and video streams emerging from two different sources are merged.

- **Session Layer Services:**

Session layer provides following services:

1. Dialog management
2. Synchronization
3. Activity Management
4. Exception Handling

1. Dialog Management:

- The session layer behaves as a dialog controller.
- It allows two communication machines to enter into a dialog.
- it allows the communication between two processes which can be either half-duplex or full-duplex.

2. Synchronization:

- Synchronization refers to the idea that multiple processes are to join up or handshake at a certain point, so as to reach an agreement or commit to a certain sequence of action.
- Data synchronization refers to the idea of keeping multiple copies of a dataset in coherence with one another, or to maintain data integrity.
- This layer allows a process to add checkpoints which are considered as synchronization points into stream of data.
- If some error occurs in the middle of the transmission of data, then the transmission will take place again from the checkpoint.
- This process is known as Synchronization and recovery.

- **Example:**

  If a system is sending a file of 2500 pages, It is advisable to add checkpoints after every 100 to ensures that a 100-page unit is successfully received and acknowledged independently.

- In this case, if a crash happens during transmission of page number 824; then retransmission begins on page 801. There is no need to retransmit pages 1 to 800 pages.

3. Activity Management:

- Allow the user to delimit data into logical units called *activities*.
- Each activity is independent of activities that come before and after it, and an activity can be processed on its own.
- Activities might be used to delimit files of a multi-file transfer.
- Activities are also used for isolating, collecting all the messages of a multi-message exchange together before processing them.
- The receiving application would begin processing messages only after all the messages had arrived.
- This provides a way of helping insure that all or none of a set of operations are performed.

For example,

1. a bank transaction may consist of locking a record
2. updating a value, and then
3. unlocking the record.

If an application processed the first operation, but never received the remaining operations (due to client or network failures), the record would remain locked forever.

4. Exception handling:

A General purpose mechanism for reporting errors.

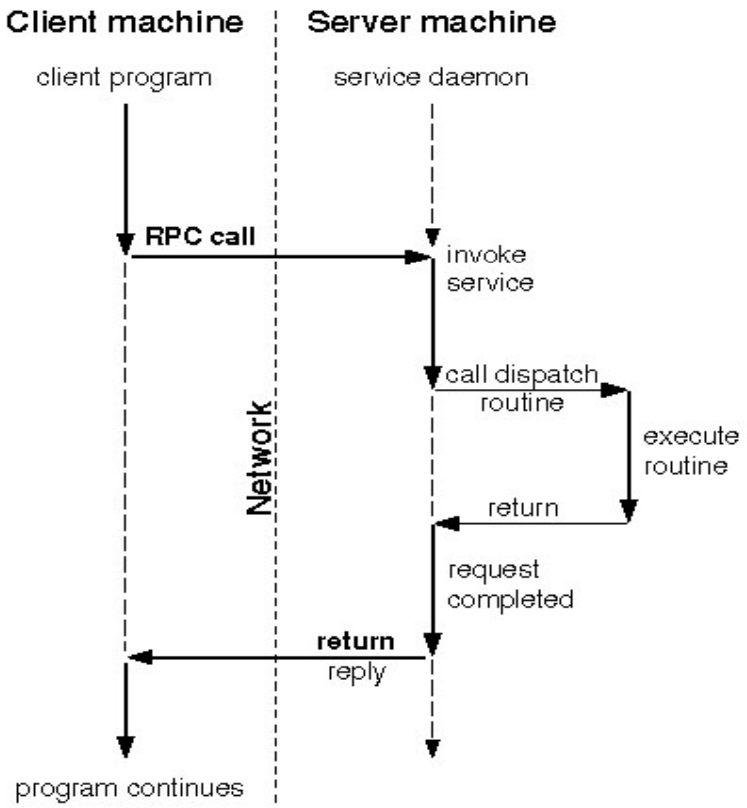
- **Remote Procedure Call (RPC):**

- A remote procedure call is an inter process communication technique that is used for client-server based applications.
- It is also known as a subroutine call or a function call.
- A client has a request message that the RPC translates and sends to the server.
- This request may be a procedure or a function call to a remote server.
- When the server receives the request, it sends the required response back to the client.
- The client is blocked while the server is processing the call and only resumed execution after the server is finished.

The following steps take place during an RPC:

1. A client invokes a *client stub* procedure, passing parameters in the usual way. The client stub resides within the client's own address space.
2. The client stub *marshalls* the parameters into a message. Marshalling includes converting the representation of the parameters into a standard format, and copying each parameter into the message.
3. The client stub passes the message to the transport layer, which sends it to the remote server machine.
4. On the server, the transport layer passes the message to a server stub, which demarshalls the parameters and calls the desired server routine using the regular procedure call mechanism.

4. When the server procedure completes, it returns to the server stub (e.g., via a normal procedure call return), which marshalls the return values into a message. The server stub then hands the message to the transport layer
5. The transport layer sends the result message back to the client transport layer, which hands the message back to the client stub.
6. The client stub demarshalls the return parameters and execution returns to the caller.



- **RPC Issues:**

1. Marshalling:

- Parameters must be *marshalled* into a standard representation.
- Parameters consist of simple types (e.g., integers) and compound types (e.g., C structures or Pascal records).
- Moreover, because each type has its own representation, the types of the various parameters must be known to the modules that actually do the conversion.
- The client stub packages up the parameters into a network message. This is called marshalling.
- The server stub unmarshals the arguments from the network message and server executes procedure.
- The procedure is completed then the server stub marshals the result into a network message and are sent back to the client.
- Client stub reads the message, unmarshals it and then store it into stack.

2. Binding:

- How does the client know *who* to call, and *where* the service resides?
- The most flexible solution is to use *dynamic binding* and find the server at run time when the RPC is first made.
- The first time the client stub is invoked, it contacts a name server to determine the transport address at which the server resides.

3. Semantics:

- The client and server don't share an address space. That is, addresses referenced by the server correspond to data residing in the client's address space.
- One approach is to simulate call-by-reference using *copy-restore*.
- In copy-restore, call-by-reference parameters are handled by sending a copy of the referenced data structure to the server, and on return replacing the client's copy with that modified by the server.
- However, copy-restore doesn't work in all cases. For instance, if the same argument is passed twice, two copies will be made, and references through one parameter only changes one of the copies.

4. Transport protocol:

- What transport protocol should be used?
- In RPC, transport protocol that is TCP is used.

5. Exception Handling:

How are errors handled?

- **Advantages of Remote Procedure Call**

1. Remote procedure calls support process oriented and thread oriented models.
2. The internal message passing mechanism of RPC is hidden from the user.
3. The effort to re-write and re-develop the code is minimum in remote procedure calls.
4. Remote procedure calls can be used in distributed environment as well as the local environment.
5. Many of the protocol layers are omitted by RPC to improve performance.

- **Disadvantages of Remote Procedure Call**

1. The remote procedure call is a concept that can be implemented in different ways. It is not a standard.
2. There is no flexibility in RPC for hardware architecture. It is only interaction based.
3. There is an increase in costs because of remote procedure call.

- **Presentation Layer**

- A Presentation layer is mainly concerned with the syntax and semantics of the information exchanged between the two systems.
- It acts as a data translator for a network.
- This layer is a part of the operating system that converts the data from one presentation format to another format.
- The Presentation layer is also known as the syntax layer.

- **Presentation layer functions or Services:**

1. **Translation:**

- The processes in two systems exchange the information in the form of character strings, numbers and so on.
- Different computers use different encoding methods, the presentation layer handles the interoperability between the different encoding methods.
- It converts the data from sender-dependent format into a common format and changes the common format into receiver-dependent format at the receiving end.

2. **Encryption:**

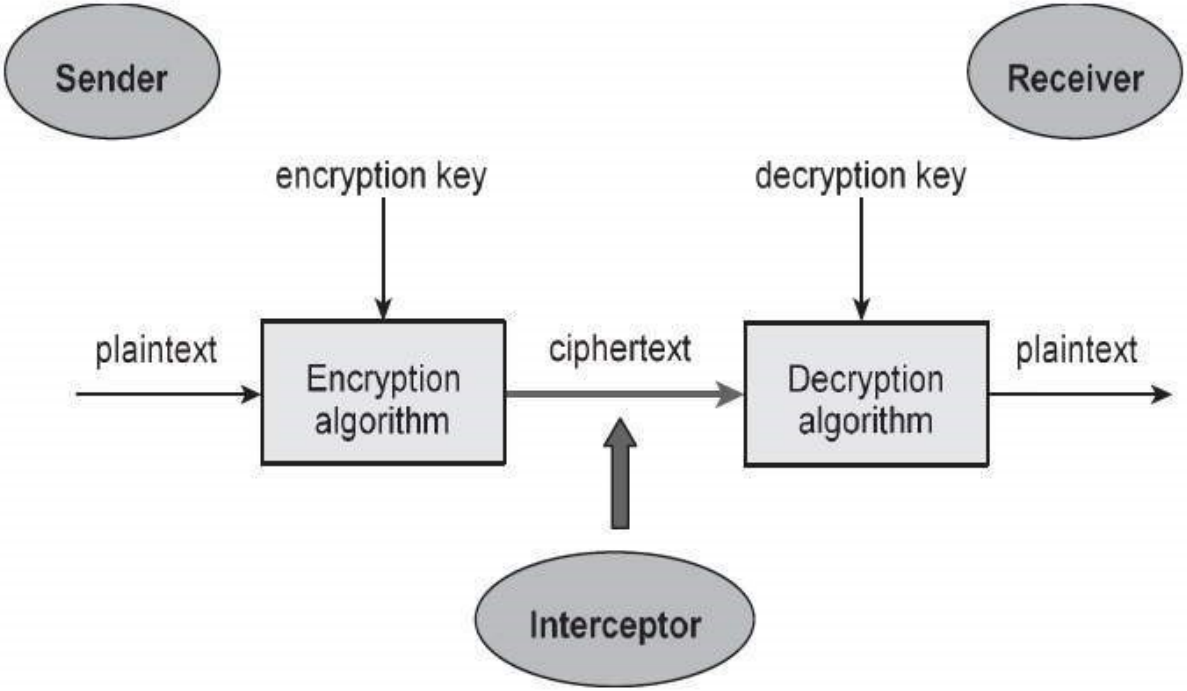
- Encryption is needed to maintain privacy. Encryption is a process of converting the sender-transmitted information into another form and sends the resulting message over the network.

3. **Compression:**

- Data compression is a process of compressing the data, i.e., it reduces the number of bits to be transmitted. Data compression is very important in multimedia such as text, audio, video.

- **Cryptography:**

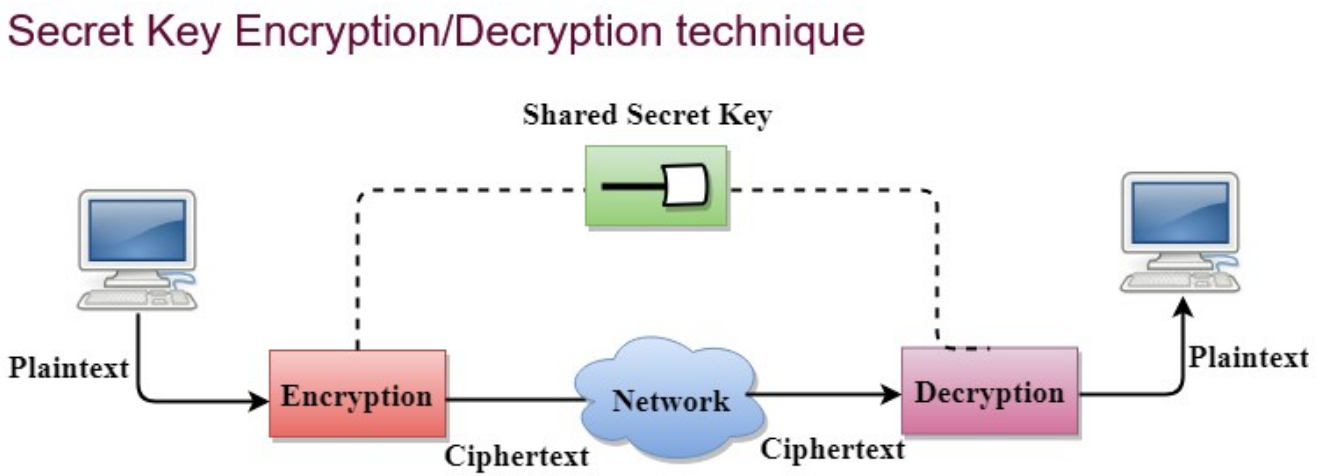
- Cryptography is a method of protecting information and communications through the use of codes, so that only those for whom the information is intended can read and process it.
- The "crypt-" means "hidden or secret" and the "graphy" means "writing.“ So the cryptography means “Secret Writing”.
- In computer science, cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher.



- **Types of Cryptography:**

1. Symmetric Or Secret Key cryptography
2. Asymmetric Or Public Key cryptography

1. Symmetric Or Secret Key cryptography:



- In Secret Key Encryption/Decryption technique, the same key is used by both the parties, i.e., the sender and receiver.
- The sender uses the secret key and encryption algorithm to encrypt the data; the receiver uses this key and decryption algorithm to decrypt the data.
- In Secret Key Encryption/Decryption technique, the algorithm used for encryption is the inverse of the algorithm used for decryption. It means that if the encryption algorithm uses a combination of addition and multiplication, then the decryption algorithm uses a combination of subtraction and division.
- The secret key encryption algorithm is also known as symmetric encryption algorithm because the same secret key is used in bidirectional communication.
- In secret key encryption/decryption algorithm, the secret code is used by the computer to encrypt the information before it is sent over the network to another computer.
- The secret key requires that we should know which computers are talking to each other so that we can install the key on each computer.

- **Data Encryption Standard (DES):**

- The Data Encryption Standard (DES) was designed by IBM and adopted by the U.S. government as the standard encryption method for non military and non classified use.
- The Data Encryption Standard is a standard used for encryption, and it is a form of Secret **Key Cryptography**.

- **Advantage:**

1. Efficient:

- The secret key algorithms are more efficient as it takes less time to encrypt the message than to encrypt the message by using a public key encryption algorithm.
- The reason for this is that the size of the key is small. Due to this reason, Secret Key Algorithms are mainly used for encryption and decryption.

- **Disadvantages of Secret Key Encryption:**

1. Each pair of users must have a secret key. If the number of people wants to use this method in the world is N, then there are N(N-1)/2 secret keys. For example, for one million people, then there are half billion secret keys.
2. The distribution of keys among different parties can be very difficult. This problem can be resolved by combining the Secret Key Encryption/Decryption with the Public Key Encryption/Decryption algorithm.

2. Asymmetric or Public Key Encryption/Decryption technique

- There are two keys in public key encryption: a private key and a public key.
- The private key is given to the receiver while the public key is provided to the public.

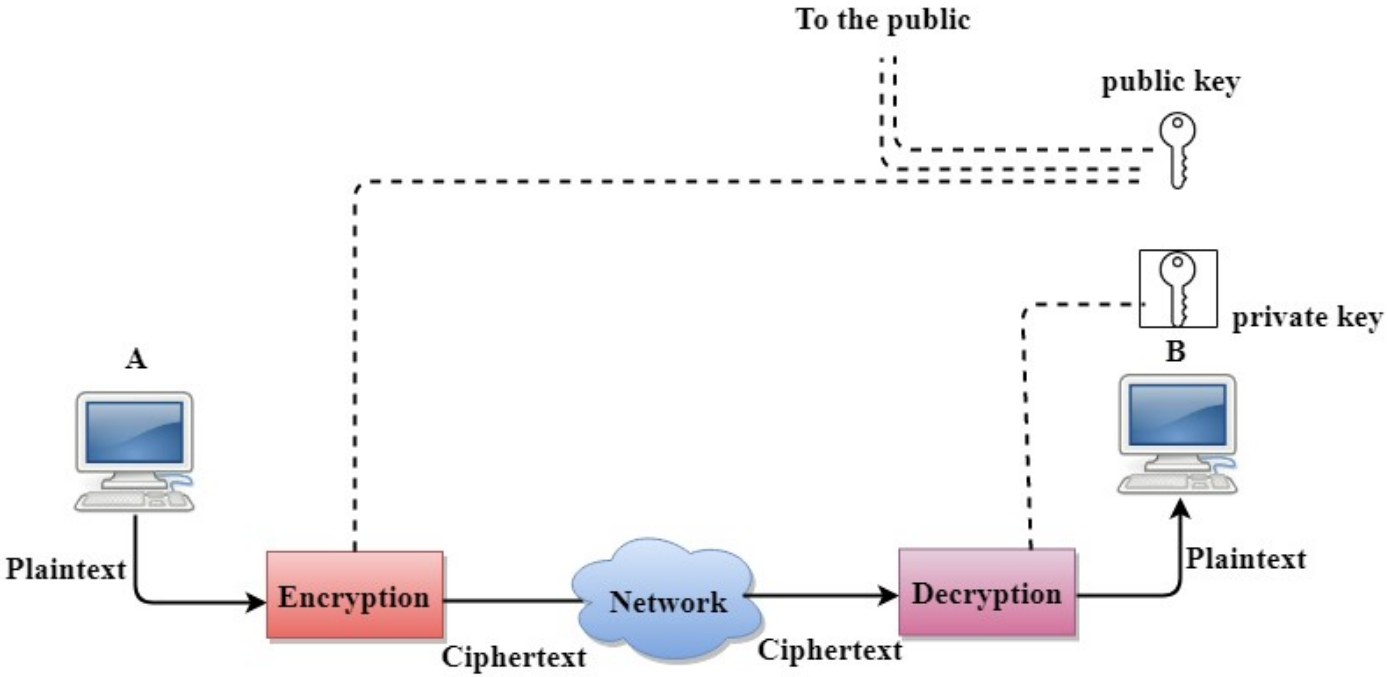


Fig. Asymmetric or Public Key Cryptography

- In the above figure, we see that A is sending the message to user B. 'A' uses the public key to encrypt the data while 'B' uses the private key to decrypt the data.
- In public key Encryption/Decryption, the public key used by the sender is different from the private key used by the receiver.
- The public key is available to the public while the private key is kept by each individual.
- The most commonly used public key algorithm is known as RSA.

- **Advantages of Public Key Encryption:**

1. The main restriction of private key encryption is the sharing of a secret key. A third party cannot use this key. In public key encryption, each entity creates a pair of keys, and they keep the private one and distribute the public key.
2. The number of keys in public key encryption is reduced tremendously. For example, for one million users to communicate, only two million keys are required, not a half-billion keys as in the case of secret key encryption.

- **Disadvantages of Public Key Cryptography:**

1. **Speed:** One of the major disadvantage of the public-key encryption is that it is slower than secret-key encryption. In secret key encryption, a single shared key is used to encrypt and decrypt the message which speeds up the process while in public key encryption, different two keys are used, both related to each other by a complex mathematical process. Therefore, we can say that encryption and decryption take more time in public key encryption.
2. **Authentication:** A public key encryption does not have a built-in authentication. Without authentication, the message can be interpreted or intercepted without the user's knowledge.
3. **Inefficient:** The main disadvantage of the public key is its complexity. If we want the method to be effective, large numbers are needed. But in public key encryption, converting the plaintext into cipher text using long keys takes a lot of time. Therefore, the public key encryption algorithms are efficient for short messages not for long messages.

THANK YOU…