IEEE 802.1X PORT-BASED NETWORK ACCESS CONTROL:

IEE 802.1X port-based network access control was designed to provide access control for functions for LANs.

Until an AS authenticates a supplicant (using an authentication protocol), the authenticator only passes control and authentication messages between the supplicant and the authentication server (AS), the 802.1X control and channel is unblocked, but the 802.11 data channel is blocked.

Once a supplicant is authenticated and keys are provided, the authenticator can forward data from the supplicant, subject to predefined access control limitations for the supplicant to the network. Under these circumstances, the data channel is unblocked.

802.1X uses the concepts of controlled and uncontrolled ports.

Ports are logical entities defined within the authenticator and refer to physical network connections. Each logical port is mapped to one of these two types of physical ports.

An uncontrolled port allows the exchange of protocol data units (PDUs) between the supplicant and the authentication server, regardless of the authentication state to the supplicant.

A controlled port allows the exchange of PDU's between a supplicant authorizes such an exchange.

The essential element defined in 802.1X is a protocol known as EAPOL (EAP over LAN). EAPOL operates at the network layer and makes use of an IEEE 802 LAN, such as ethernet or Wi-Fi, at the link level.

EAPOL enables a supplicant to communicate with an authentication and supports

the exchange of EAP packets for authentication.

When the supplicant first connects to the LAN, it does not know the MAC addres of the authentication. Actually it does not know whether there is an authenticator present at all.

By sending an EAPOL-start packet to a special group-multicast address reserved for IEEE 802.1X authenticators, a supplicant can determine whether an authenticator is present and let it know that the supplicant is ready. In many cases, the authenticator will already be notified that a new device has connected from some hardware notification.

For example, a hub knows that a cable is plugged in before the device sends any data. In this case the authenticator many preempt the start message with its own message.

In either case the authenticator sends an EAP Request Identity message encapsulated in an EAPOL - EAP Packet.

The EAPOL-EAP is the EAPOL frame type used for transporting EAP packets.

The authenticator uses the EAP-key packet to send cryptographic keys to the supplicant. Once it has decided to admit it to the Network. The EAP-Logoff packet types indicates that the supplicant wishes to be disconnected from the network.

The EAPOL packet format includes the following fields:

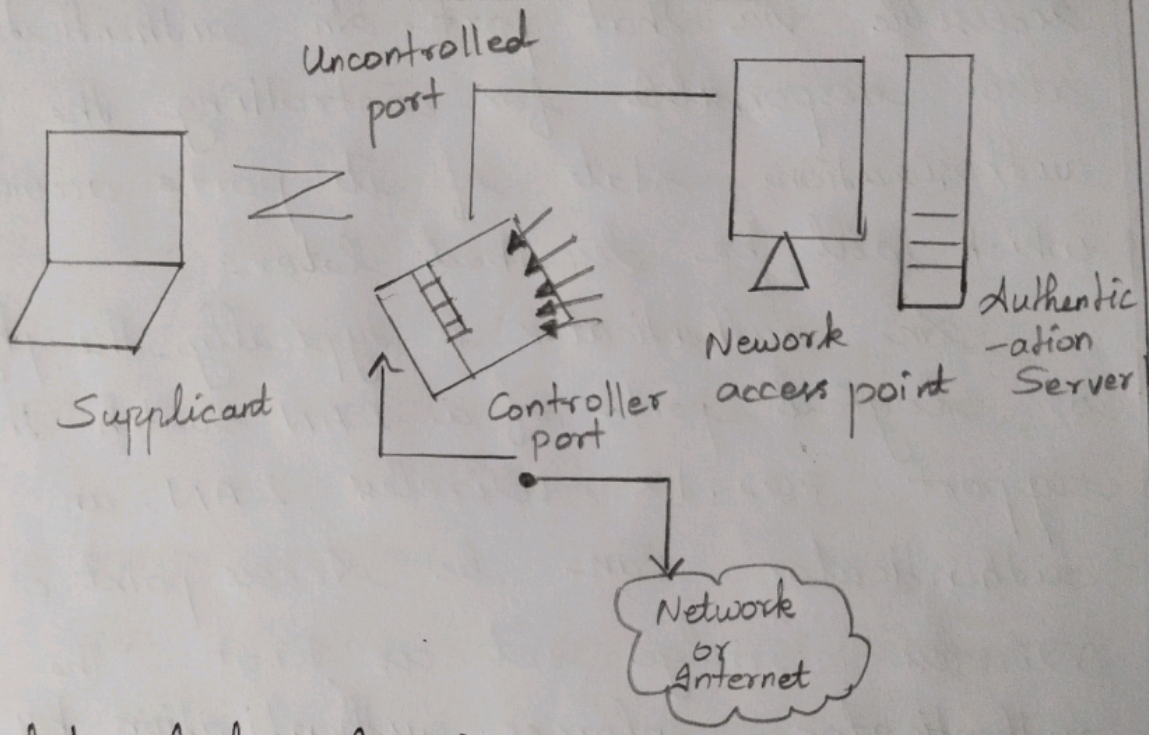
Protocol version : version of EAPOL

packet type : Indicates start, EAP, key, log off, etc...

Common EAPOL frame types:

Frame Type	Definition
EAPOL-EAP	Contains an Encapsulated EAP packet.
EAPOL-Start	A supplicant can issue this packet instead of waiting for a challenge from the authenticator.

EAPOL - Log off	Used to return the stat of the port to unauthorized when the supplicant if finished using the network.
EAPOL - Key	Used to exchange cryptographic keying information.



Packet body length:

If the packet includes a body, this field indicates the body length.

Packet body:

The payload for this EAPOL packet. An example is an EAP packet. There are three principal roles in 802.1X operation

to make the mechanism work.

Authenticator:

An authenticator is responsible for enforcing the authentication of a device that attaches to its controlled port before allowing access to LAN services these are accessible via that port. An authenticator is also responsible for controlling the authorization state of its ports accordingly which will be described later.

An authenticator is typically the physical or logical ports of a LAN switch that support 802.1X. Wireless LAN, an authenticator can be Access point or a Wireless Bridge set as Root. The authenticator enforces authentication by forwarding access decision to the Authentication Server.

Supplicant:

A supplicant is responsible for communicating its credentials to the Authenti-cator in response from the authenticator. The supplicant may initiate authentication.

exchanges beside the Authenticator itself, depending on the setting.

A network adapter of a workstation, or refer as client, may play the role of the supplicant the network adapter can be a typical Ethernet card adapter running 802.1X-Compliant client software, or a wireless LAN client that support 802.1X.

A port of network element can also be a supplicant. The examples are, a switch's port that connect to another switches port, and a Non-root wireless Bridge that connect to a Root Wireless Bridge.

Authentication Server:

The Authentication server performs the authentication function to check the credentials of the supplicant on behalf of the Authenticator and indicated whether the supplicant is authorized to access the LAN Services.

The authentication server is typically a RADIUS (Remote Authentication Dial-In

User Service) Server with Extensible Authentication Protocol (EAP) Support.

Authentication types used with 802.1X:

There are several EAP authentication types:

- EAP-MD5:

* EAP - Message Digest 5 is an EAP type that uses an MD5 hash of a username and password to create challenges and responses from the client to the RADIUS Server.

* EAP-MD5 also does not provide mutual authentication. It only allows for the server to validate the client. That's why EAP-MD5 is considered to be the least secure EAP Authentication types among others.

- EAP-TLS:

EAP - Transport layer security is an EAP type defined in RFC 2716 that is used in certificate-based security environments. The EAP-TLS exchange of messages provides mutual authentication

with both client and Server mutually validating each other via certificates. In wireless network, EAP-TLS provides dynamic WEP key generation, thus strengthen wireless LAN Security.

- EAP-TTLS:

EAP Tunneled Transport Layer Security is an extension of EAP-TLS, which requires only server-side certificates, eliminating the need to configure certificates for each client.

EAP-TTLS still maintain mutual authentication as EAP-TLS because users are authenticated to the network using ordinary password-based credentials.

- EAP-Cisco Wireless or also called Light Weight EAP (LEAP):

This EAP authentication type is developed by cisco and used primarily in Cisco Wireless LAN devices. It is a proprietary authentication type from Cisco.

In the Wireless LAN, it encrypts data transmission using dynamically

generated WEP Keys, and Support mutual authentication.

LEAP is developed to overcome EAP-MD5 in wireless LAN where the WEP Keys are static and offers no mutual authentication. LEAP provides mutual authentic-ation because client will authentication because client will authenticate RADIUS server after it is authenticated.

- Protected EAP (PEAP):

PEAP authentication is designed to support one-Time password (OTP), windows NT or 2000 domain, and LDAP user databases over a wireless LAN.

It is based on EAP-TLS authentication but uses a password or PIN instead of a client certificate for authentication.

In wireless LAN, PEAP uses a dynamic session-based WEP key, which is derived from the client adapter and RADIUS server, to encrypt data.

Secure Shell

The secure shell (SSH) is an access credential that is used in the SSH protocol. In other words, it is a cryptographic network protocol that is used for transferring encrypted data over the network. The port number of SSH is 22. It allows you to connect a server, on multiple servers, without having to remember or enter your password for each system. That is to log remotely from one system to another. It always comes in key pairs.

Public key

Everyone can see it, no need to protect it (for encryption function).

Private key

Stays in computer, must be protected (for decryption function)

Key pairs can be of the following types:-

⇒ User key- If the public key and private key remain with the user.

⇒ Host key - If public key and private key are on a remote system.

⇒ Session key - Used when a large amount of data is to be transmitted.

Features of SSH:

Encryption:

Encrypted data is exchanged between the server and client, which ensures confidentiality and prevents unauthorized attacks on the system.

Authentication:

For authentication, SSH uses public and private key pairs which provide more security than traditional password authentication

Data Integrity:

SSH provides data integrity of the message exchanged during the communication.

Tunneling:

Through SSH we can create secure tunels for forwarding network connections over encrypted channels.

SSH functions:

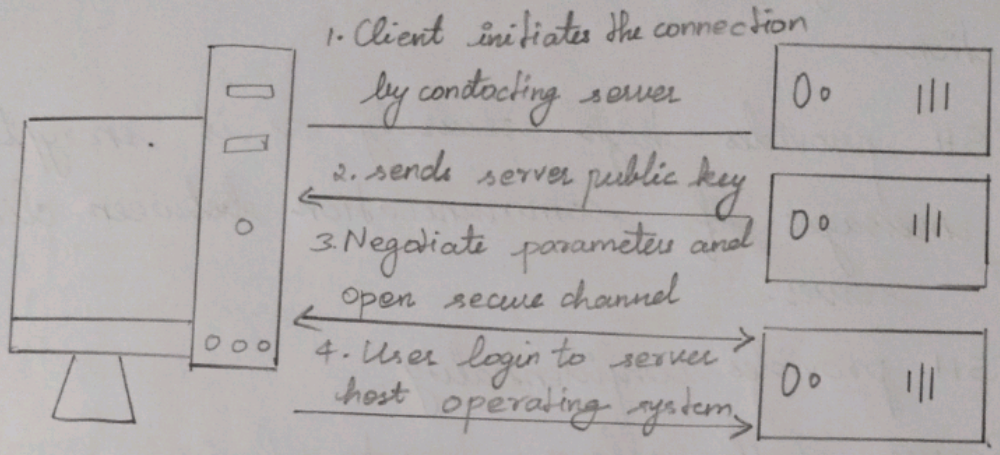
There are multiple functions performed

by SSH function, here below are some functions.

* SSH provides high security as it encrypts all messages of communication between client and server.
* SSH provides confidentiality.
* Secure shell allows remote login, hence is a better alternative to TELNET.
* Secure shell provides a secure file transfer protocol, which means we can transfer files over the Internet securely.
* Secure shell supports tunneling which provides more secure connection communication

SSH protocol:

To provide security between a client and a server the SSH protocol uses encryption. All user authentication and file transfers are encrypted to protect the network against attacks.



Secure Shell Client

Secure Shell Server

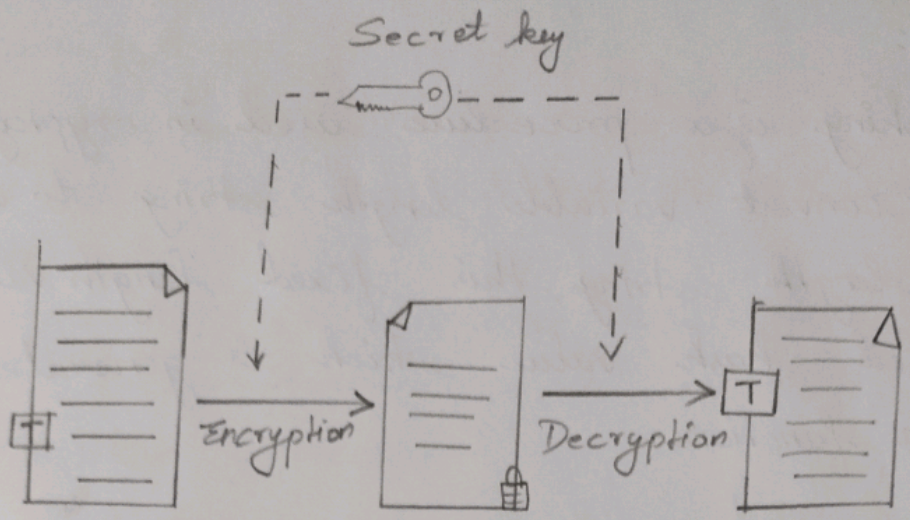
Techniques used in Secure Shell

There are majority three techniques used in secure shell, which are

Symmetric Cryptography:

In symmetric key cryptography the same key used for encrypting and decrypting the message, a unique single shared key is kept between the sender and receiver.

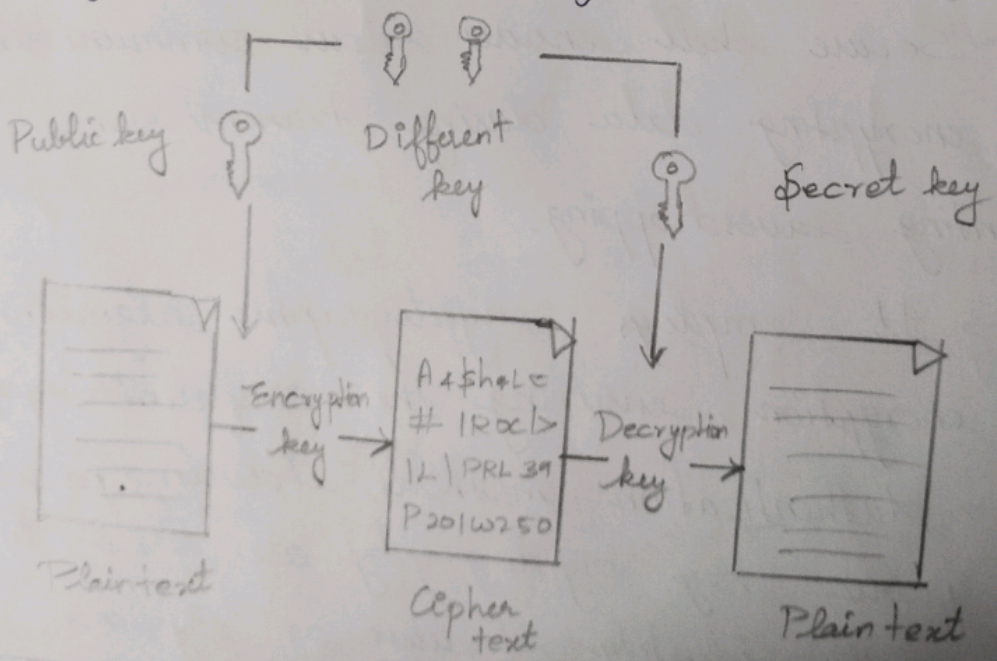
For example: DES (Data Encryption Standard) and AES (Advanced Encryption Standard).



Asymmetric cryptography:

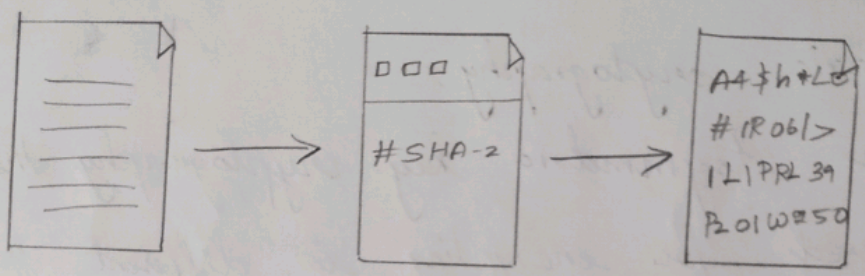
In Asymmetric key cryptography the key used for encrypting is different from the key used for decrypting the message.

For example: RSA (Rivest-Shamir-Adleman) and Digital Signature Algorithm.



Hashing:

Hashing is a procedure used in cryptography which convert variable length string to a fixed length string, this fixed length value is called hash value which is generated by hash function.



Secure Shell application:

1. Encryption and authentication:

→ Secure shell ensures secure communication by encrypting data during transmission, preventing eavesdropping.

→ It employs cryptographic algorithms for encryption, ensuring the confidentiality of data.

→ Authentication methods, including passwords and public key cryptography are used to verify the identity of users or systems.

2. Remote Access:

→ Secure shell is widely used for secure remote access to servers and network devices.

→ It replaces insecure protocols like Telnet, which transmit data in plaintext, making them susceptible to interception.

3. Secure file transfer:

→ Secure shell facilitates secure file transfer through tools like SCP (Secure Copy protocol) and SFTP (Secure file transfer protocol).

→ SCP allows copying files between hosts in a secure manner, while SFTP provides a more advanced file transfer capability with features like directory listing and remote file management.

4. Port forwarding:

→ Secure shell supports port forwarding, allowing the creation of secure tunnels for various applications.

→ This feature enhances security by encrypting communication between the client and the server for applications that may not inherently support encryption.

5. Key Management:

→ Public-key cryptography is a fundamental component of secure shell providing a secure and convenient way to authenticate users.

→ The Users can generate key pairs, consisting of a public key (shared) and a private key (kept secret). The private key is used for authentication.

6. Confidentiality:

→ Secure shell is highly configurable, allowing administrators to define security policies, access controls and other parameters.

→ Configuration files enables customization based on the specific security requirements of a network.

7. Security Best practices:

Regularly updating secure shell software is crucial to address security vulnerabilities.

Monitoring and analyzing secure shell logs contribute to the detection of suspicious activities.