Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI) is a collection of servers used to create and manage public keys and digital certificates. It creates digital certificates which bind public keys to entities, stores them securely and revokes them when required.

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

So the public key infrastructure provides a secure environment for online transactions, confidential email and e-commerce by:

- Authenticating the identity of the entities (the sender and the receiver)
- Maintaining the data integrity.

authentication ko kura vairaxa so data integrity esai hunxa lekhdine

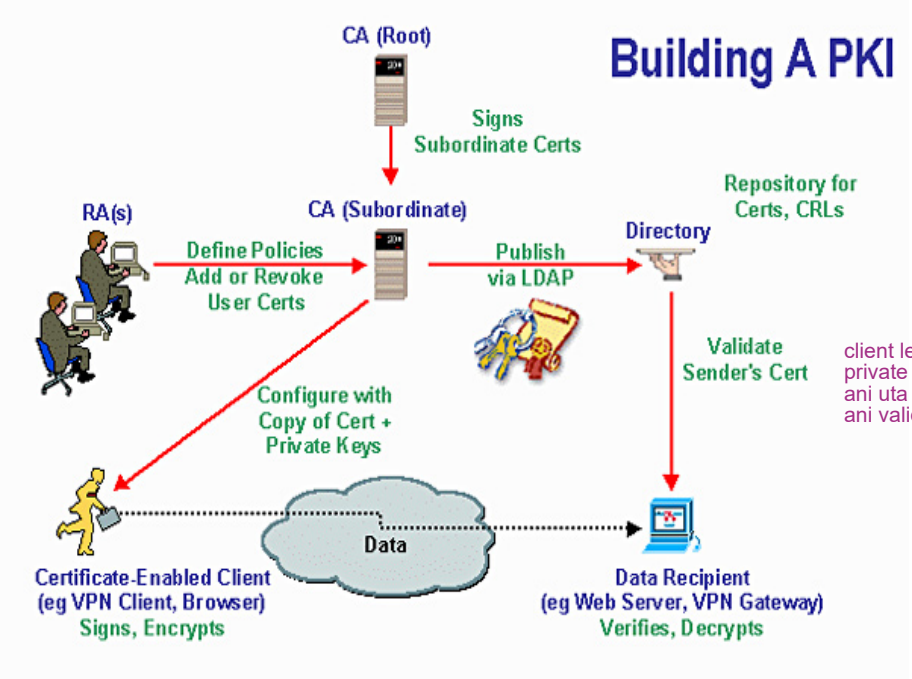
Consider an example where asymmetric key encryption is used for communication between two entities say Alice and Bob. Alice uses Bob's public key to encrypt the message and sends it to Bob. Bob decrypts the received message using his private key and reads the message. The main drawback of this method is that there is no way to guarantee that the public key which Alice used for the encryption of the message actually belongs to Bob or some other malicious user claiming to be Bob.

sender le chai receivre ko public key use gaerew message encrypt
problem vneko tyo key receiver ko nahunani skxa ni

To address this problem PKI has evolved. To ensure the identity of the entities in a communication, PKI uses a trusted third party to distribute keys and to authenticate the identities. This is done by integrating digital certificates.

Components of public key infrastructure are:

- Certificate Authorities
- Registration Authorities
- Certificate Repositories
- Digital Certificates



yo diagram ma client chai client verify vairaxa

client le cerficate rw
private key pairaxa
ani uta sever lai tyo certificate dekhauxa
ani validate grxa server le chai

Figure 1: PKI Components

1 Certificate Authority

Certificate Authority (CA) is a trusted third party that authenticates the identities of servers, individuals and other entities. A CA confirms the identity of the entity by issuing a digital certificate that binds the identity with the pubic key of that entity.

Functions of the CA are:

list tayar parne (kun kun expire huna lako xa tyo ani expire vako xa xaena xanew kun kun maintain garirakhne
publish ni esai le grne

- Issuing certificates
- Maintain and issue Certificate Revocation Lists (CRLs)
- Publish its certificates and CRls
- Maintain status information of certificate expiration dates

These tasks may be delegated by the CA.

One of the main function of a certificate authority is issuing certificates (i.e. creating the certificates and signing them). Consider a server that has requested for a digital certificate for itself. After its identity has been verified by the registration authority, the request is then forwarded to the certificate authority. The certificate authority generates a certificate in a standard format (X.509 certificate standard). The

certificate contains the identity of the server and its public key. This certificate is then signed by the certificate authority with its own private key and the certificate is issued to the requesting server. The CA's signature on the certificate verifies the integrity of the certificate. A copy of the certificate is locally saved and it may also be published in public repositories.

esko thau ma tyo hierarchy wala diagram banaune

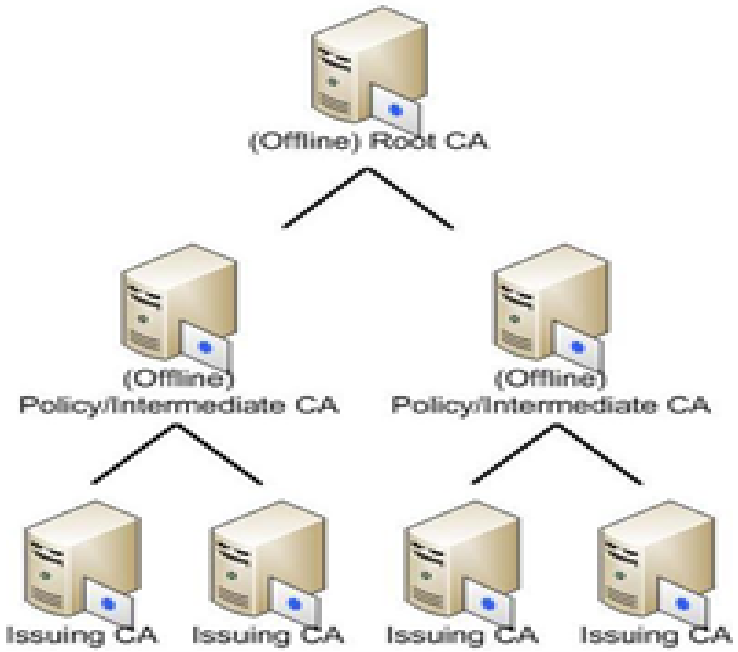


Figure 2: Certificate Hierarchy

The certificate authority is the root of trust in a public key infrastructure. When a hierarchical architecture of CA is followed, there is a root CA which has its own digital certificate. Such a certificate is self-signed. The root CA creates a chain of trust by signing certificates of the subordinate certificate authorities. This means that the certificates issued by subordinate CA's are trusted by the root CA. So a web browser or a user can trust a certificate issued by the subordinate CA if it trusts the root CA. Most web browsers and operating systems have the certificate of the root CA embedded in them. For example in Internet Explorer, if we go to the Content tab in Internet Options, we can see the certificates. It has all certificates of the CA's that the browser trusts.

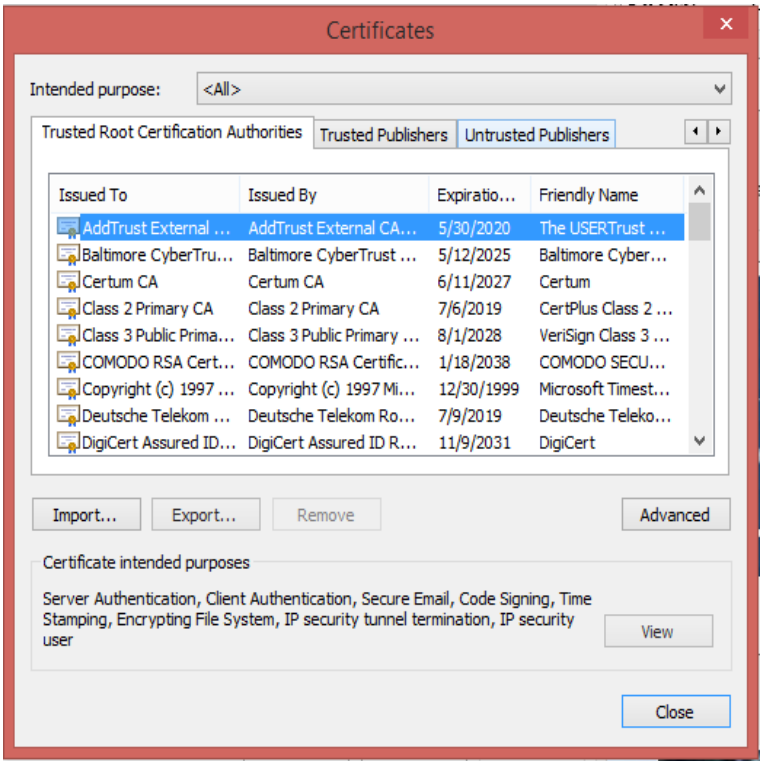


Figure 3: Internet Explorer's installed CA certificates

Another function of certificate authority is to maintain and issue Certificate Revocation List's (CRLs). The certificate authority has the right to suspend , revoke or renew a certificate. During the life of any certificate it can be suspended by the CA. At this stage its validity is temporarily suspended. The CA can revoke a certificate any time before its normal expiration. At this stage the certificate is not valid. This may happen when a private key is lost or when an unauthorized user gains knowledge of the private key. In such cases the CA updates the CRLs and its internal records with the required certificate information and time stamp. The CRLs are signed by the CA and are placed in a public repository.

Certificate Authority issues the certificates with an expiration date. Once the certificate has expired, it can no longer be used for authentication. The owner of a certificate is informed about an upcoming expiration of the certificate so that the user can follow a renewal process.

2 Registration Authority

Registration authority is a part of the public key infrastructure. It verifies the requests for digital certificates by validating the identity of the entity that places the request. For example if a company requests for a digital certificate, then the RA verifies the identity of the owner by checking various identity documents such as divers license or a pay stub etc. After verifying the identity, the RA then forwards the valid request to the CA. Then a digital certificate is issued by the CA. A CA can have more than one RA's. Each RA has a name and public key by which the CA can recognize it.Each RA is certified by its corresponding CA. Any message that the CA receives with the RA's signature is a trusted message.

3 Certificate Repositories

Certificate repositories are mainly used to store and distribute certificates. All the issued certificates are stored in the repository so that the applications can retrieve them easily. A directory system is best used for this process. Lightweight Directory Access Protocol (LDAP) is one of the best technology at present for certificate repositories. These directories store the certificates and make it easier for applications to retrieve these certificates for a user. This directory system supports a large number of certificates. It stores the certificates and the public keys for those certificates. The main advantage of these directories is that they can be used in highly distributed networks and they are made publicly accessible. It also makes the search easier by storing the certificates in a hierarchical structure. The certificate repository also contains certificate status information and revocation information. Apart from storing and distributing it also updates the certificates and their status.

4 Digital Certificates

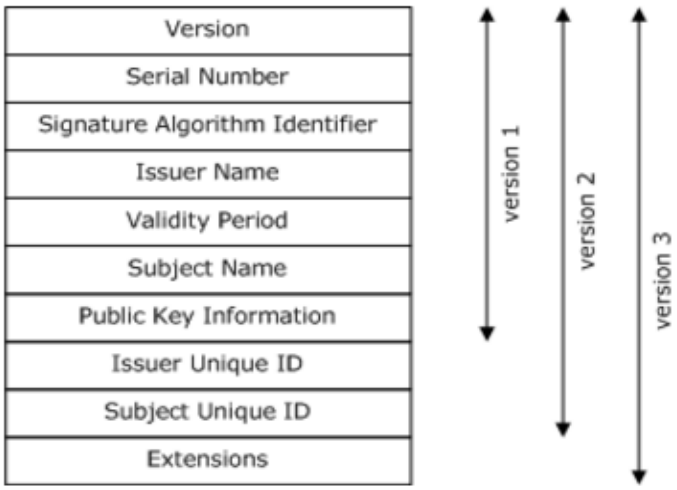
Figure 4: Digital Certificate

A Digital Certificate is an electronic document which provides information to prove the identity of an entity. It binds the identity of an entity to its public key. Digital certificates are generated in a standard format.

Consider a user who wants to shop online through an online shopping web site such as Amazon. The user types the link to the Amazon web site and the web browser connects to the web site. The main concern here is whether the web site truly belongs to Amazon company or is it a malicious party posing to be Amazon. To solve this trust issue, digital certificates are used in a public key infrastructure, and a trusted third party is used which can establish the identity of the entity and integrity of the public key.

4.1 Certificate Structure

The X.509 certificate standard is widely used to structure digital certificates. There have been three versions of this standard and at present version 3 of this standard is being used.



@veser, signa ixya vye vandihala ani subject ni deu ani public board ma afno naam id subject ani extensions lekhdeu

Figure 5: Structure a of X.509 Digital Certificate

There are ten basic fields in a digital certificate. Six of which are mandatory and four are optional fields.

The mandatory fields are:

- Serial number
- Signature algorithm
- Issuer name
- Validity period
- Subject name
- Public key information

The optional fields are:

- Version
- Issuer unique ID
- Subject unique ID
- Extensions

These optional fields are used in version 2 and version 3.

Version: This field specifies the version number of the certificate. This can be either version 1 or 2 or 3. When extensions are included

in a certificate, this field indicates version 3. If the it includes unique identifiers without extensions, then it is version 2. If it does not include extensions and unique identifiers, then it is version 1.

Serial number: It is a unique positive number assigned for each certificate. It is assigned by the issuer to identify the certificate.

Signature Algorithm: This field indicates the algorithm used by the issuer to sign the certificate. Some examples are: RSA encryption algorithm with SHA-1 hashing algorithm, RSA with MD5 or DSA with SHA-1 algorithm.

Issuer: This field indicates the X.500 Distinguished Name of the trusted third party which signed and issued the certificate.

Validity: Validity indicates the date from when the certificate is valid (i.e. valid from) and the date until when the certificate is valid (i.e. valid to).

Subject: Subject is the distinguished name of the entity that owns the certificate. The owner is the entity associated with the public key in the certificate. Owner can be a CA, RA, person, company, or application.

Public key information: This field contains the public key of the subject and the algorithm identifier. jasko certificate ho tskae public key append garine tw honi

Issuer unique ID: This is a unique identifier to facilitate the reuse of issuer's name over time.

Subject unique ID: This field contains a unique identifier to facilitate the reuse of subject's name over time.

Extensions: This field is present in version 3 certificates. The extensions are used to give more information about the certificate which is not given by the basic fields. Extensions have three basic elements: an extension identifier, criticality flag and extension value. Extension identifier gives the format of the extension value, criticality flag indicates that the extension is important. Some of the extensions are: key usage, subject name alternative, basic constraints, policy constraints, name constraints etc.

4.2 Types of Certificates

Based on the usage, digital certificates can be of different types. Such as:

- **Personal:** Certificates which are used by individuals for secure email.

- **Organisation:** Certificates used by corporate companies for internal use, to identify employees of the company for secure email.
- **Server:** Certificates used by web site owners to establish a secure connection with a user by proving the ownership of the domain name.
- **Developer:** Certificates used by developers to prove the identity of the applications and the software programs.
- **Government:** Certificates used for government security.

Based on the different classes of certificates, CA performs different levels of verification to check the identity of the owner. If a certificate belongs to a higher class, such as certificates used for online transactions, then a higher level of verification is performed. For certificates which are used for personal email, a lower amount of verification may be needed. Depending on the usage of the certificates different levels of verification are performed by the CA.

4.3 Working of Digital Certificates

Digital Certificates in a Public Key Infrastructure work in the following way:

1. Consider an online shopping web site such as Amazon. The server of the Amazon company requests for a digital certificate from a certificate authority.
2. The certificate authority verifies the identity of the company and generates a digital certificate. It hashes the contents of the certificate and signs (encrypts) the hash value using its private key. It includes this signature in the certificate and issues the certificate to the company.

   tyo public key pauxa ni tw ca le ani tyo purae message content bata hash value nikalxa ani tesali encrypt grxa using its private key
3. A user who wants to connect to the Amazon web site enters the HTTPS web address in his browser. The browser tries to connect to the web site.
4. A digital certificate is sent from the web server of the Amazon company to the browser.
5. When the browser receives a certificate from the web server it performs the following tasks:
   - It checks whether the CA who signed the certificate is trusted by the browser. The browser already has the trusted CA

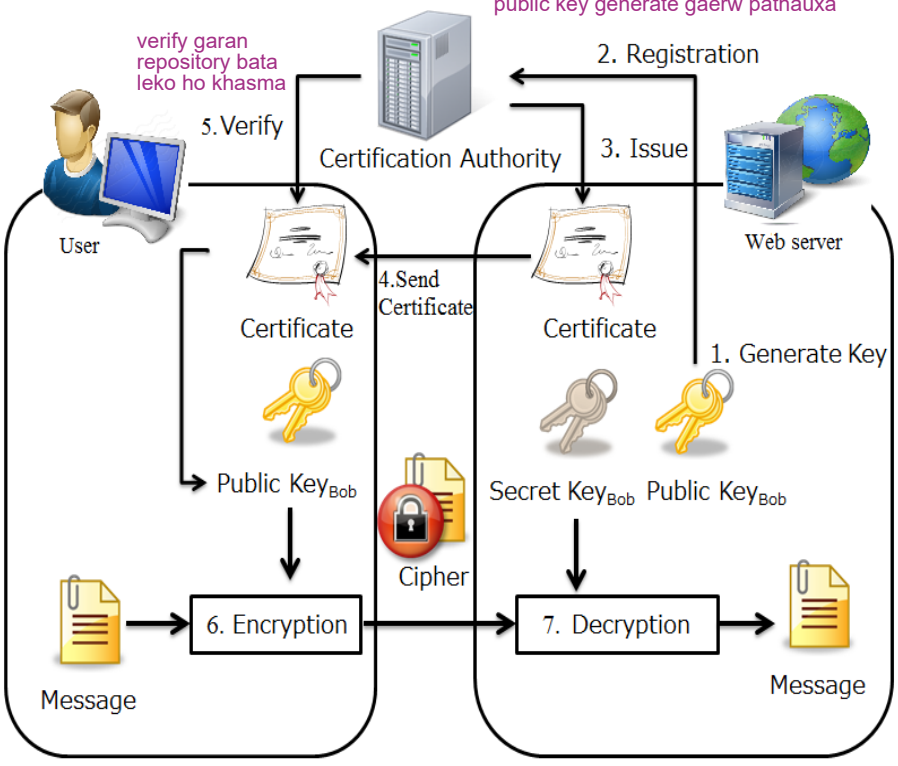
detailed explanation: suruma public key generate gryo server le ani tyo pathauxa CA lai ani usle tyoni include garerew digital certificate issue grxa ani teha content of digital certificate bata hashfunction nikalxa teslai encrypt(signs) grxa using its own private key ani yo jun digital certificate xa tyo server lai issue grxa aba server le ma trusted ho vnerw browser lai certificate pathauxa ani broswer le verify grna khojxa so uhh sanga tyo CA ko tw public key as CA trusted party ho browser ko so tyo use garerw server le pathako certificate ma vako sign decrypto grxa ani teha bata obtains hash ani content bata ni hash compute grxa if both matches tyo verify vyo natra vye vyenw

certificates installed, so it has the public key information of the CA.

- With the public key of the CA, the browser decrypts the signature in the company's certificate and obtains a hash.
- It also computes a new hash of the content in the certificate..
- If both the hashes match, then the signature in the certificate is verified to be signed by the trusted CA and the public key in the certificate is valid.
- Now the name in the certificate is checked against the web site's name. If it matches then a secure connection is established for the online transactions.
- The browser also checks whether the certificate is within its expiry period.

certificate ma public key hunxa ni tw so suruma public key generate gaerw pathauxa

verify garan repository bata leko ho khasma



eha overall ma CA le private keyle encrypt grirxa uta user le public key le decrypt so yo authentication honi tw

ani tespxi message encrypt receiver public key message decrypt receivers private key so yo confidentiality

message patahuna lai receiver kae public key le encrypt receiver le afno private key le decrypt

Figure 6: Working of Digital Certificates

All this process is transparent to the user and it is carried out in milliseconds. The integrity of the certificate is guaranteed, as long as the CA's signature can be verified. It also makes sure that the public key in the certificate is valid and has not been tampered with. It guarantees that the public key belongs to the owner of the certificate and it can be used for secure communication. Checking the name on the certificate against the web site's name helps in preventing man-in-the-middle attacks, where a malicious user modifies the certificate and claims to be the site that the user wants to establish communication with.

Transport Layer Security (TLS)
Secure Socket Layer (SSL)

SSL
(Secure Socket Layer)

Type text here

SSL History

- Netscape developed The Secure Sockets Layer Protocol (SSL) in 1994, as a response to the growing concern over security on the Internet.

- SSL was originally developed for securing web browser and server communications.
- SSL v3.0 was specified in an Internet Draft (1996)

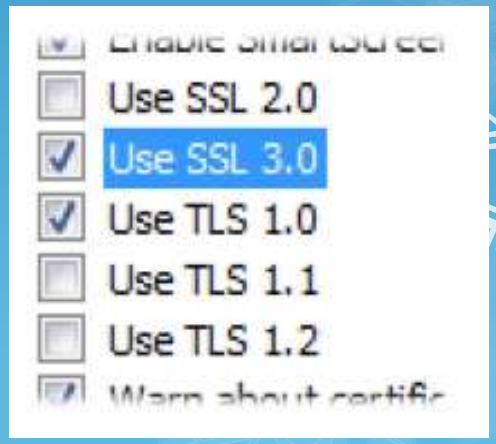
SSL (Secure Socket Layer)

- SSL is a Secure Sockets Layer
- SSL is the standard security technology for establishing an encrypted link between a web server and a browser.
- This link ensures that all data passed between the web server and browsers remain private and integral
- There are several versions of the SSL protocol defined. The latest version, the Transport Layer Security Protocol (TLS), is based on SSL 3.0

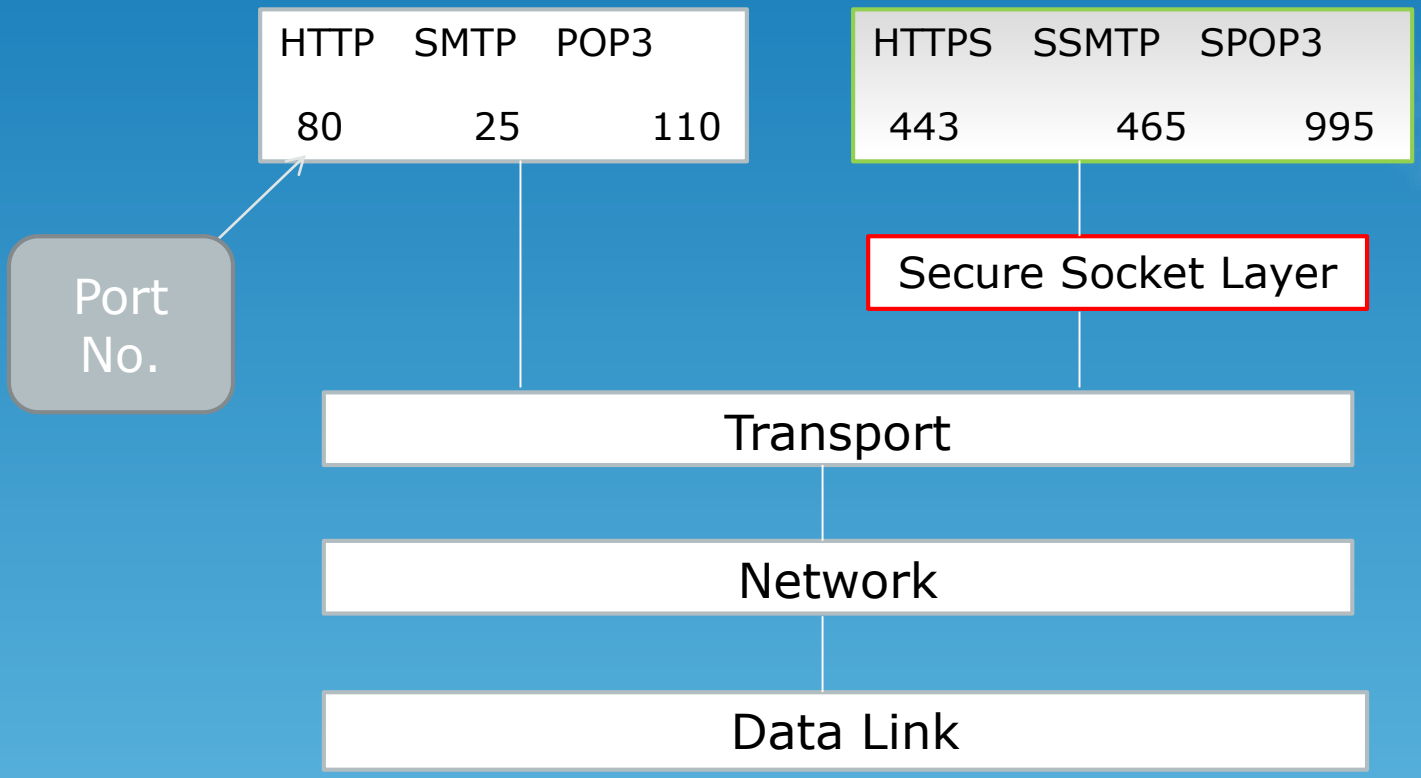
SSL Version 1.0

SSL Version 2.0

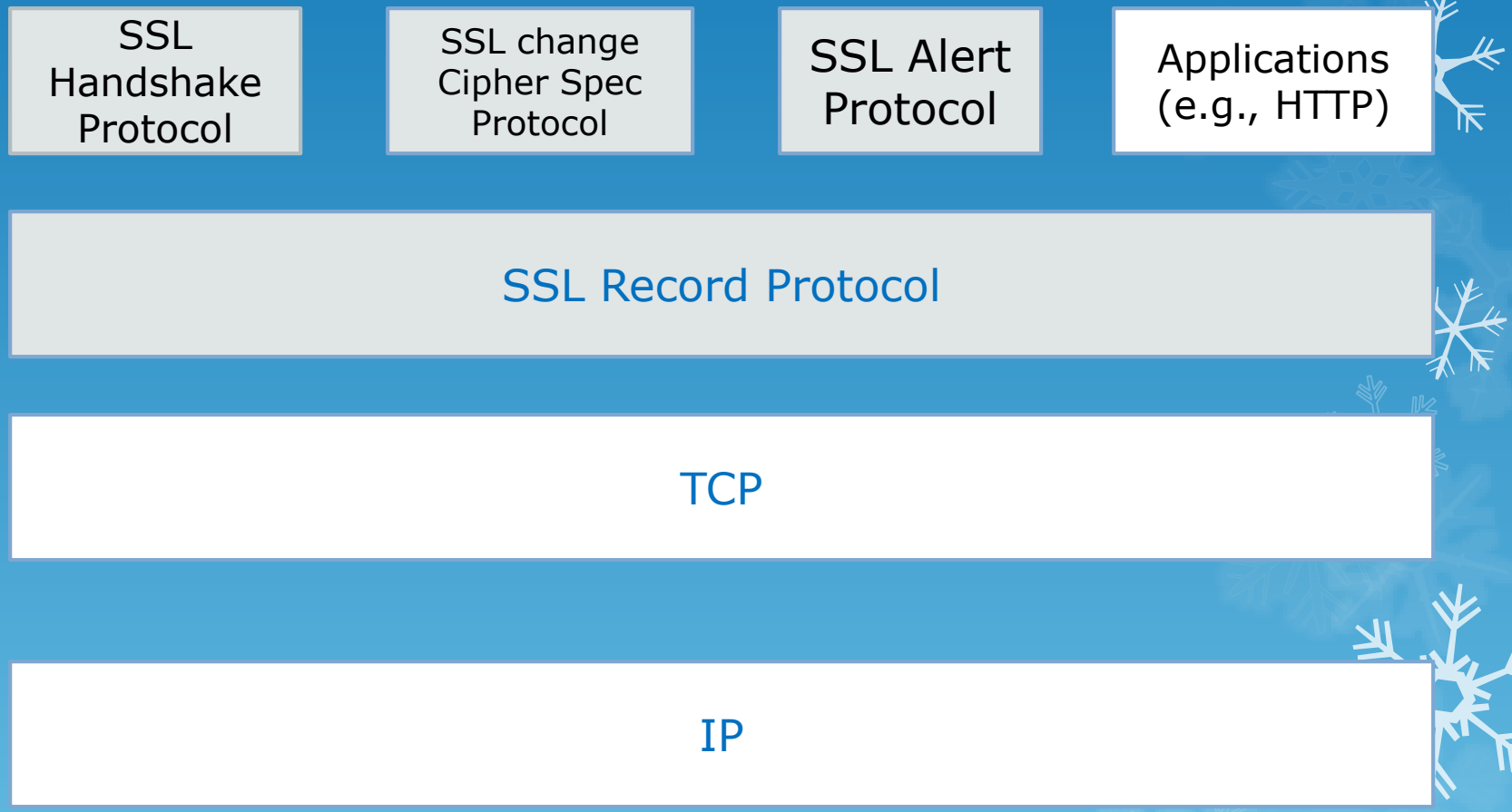
SSL Version 3.0



Where SSL fits?

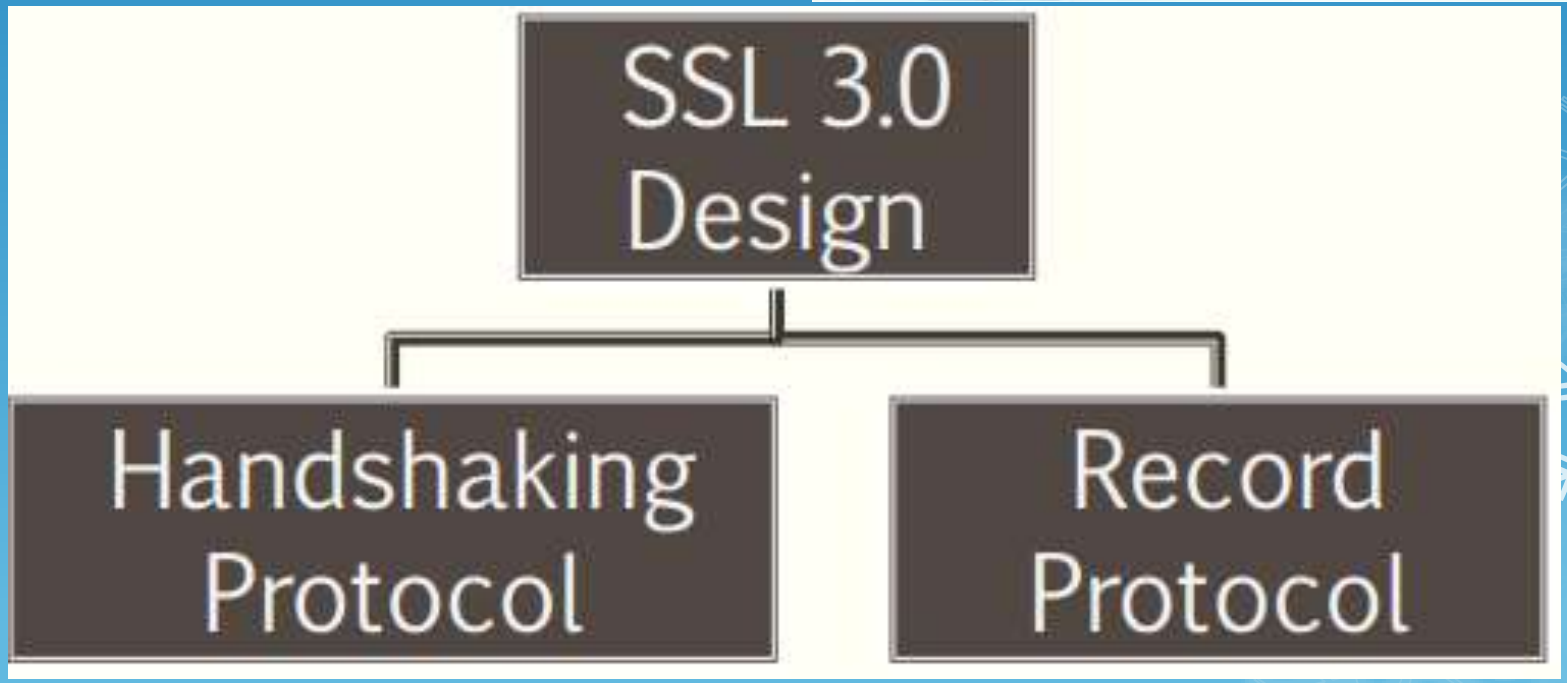


SSL architecture



SSL

- It is the most widely known as the protocol that, coupled with HTTP, secures the Web and uses the “https” URI scheme



SSL components

- **SSL Handshake Protocol**
  - Negotiation of security algorithms and parameters
  - Key exchange
  - Server authentication and optionally client authentication
- **SSL Record Protocol**
  - Fragmentation
  - Compression
  - Message authentication and integrity protection
  - Encryption
- **SSL Alert Protocol**
  - Error messages (fatal alerts and warnings)
- **SSL Change Cipher Spec Protocol**
  - A single message that indicates the end of the SSL handshake

SSL Goals

Confidentiality

The data being transmitted over the Internet or network needs confidentiality. In other words, people do not want their credit card number, account login, passwords or personal information to be exposed over the Internet.

Integrity Protection

The data needs to remain integral, which means that once credit card details and the amount to be charged to the credit card have been sent, a hacker sitting in the middle cannot change the amount to be cha rged and where the funds should go.

Authentication

Your organization needs identity assurance to authenticate itself to customers / extranet users and ensure them they are dealing with the right organization.

Your organization needs to comply with regional, national or international regulations on data privacy, security and integrity

Transport Layer Security (TLS)

Two protocols are dominant today for providing security at the transport layer

- Secure Sockets Layer (SSL) protocol
- **Transport Layer Security (TLS) protocol**

Definition:

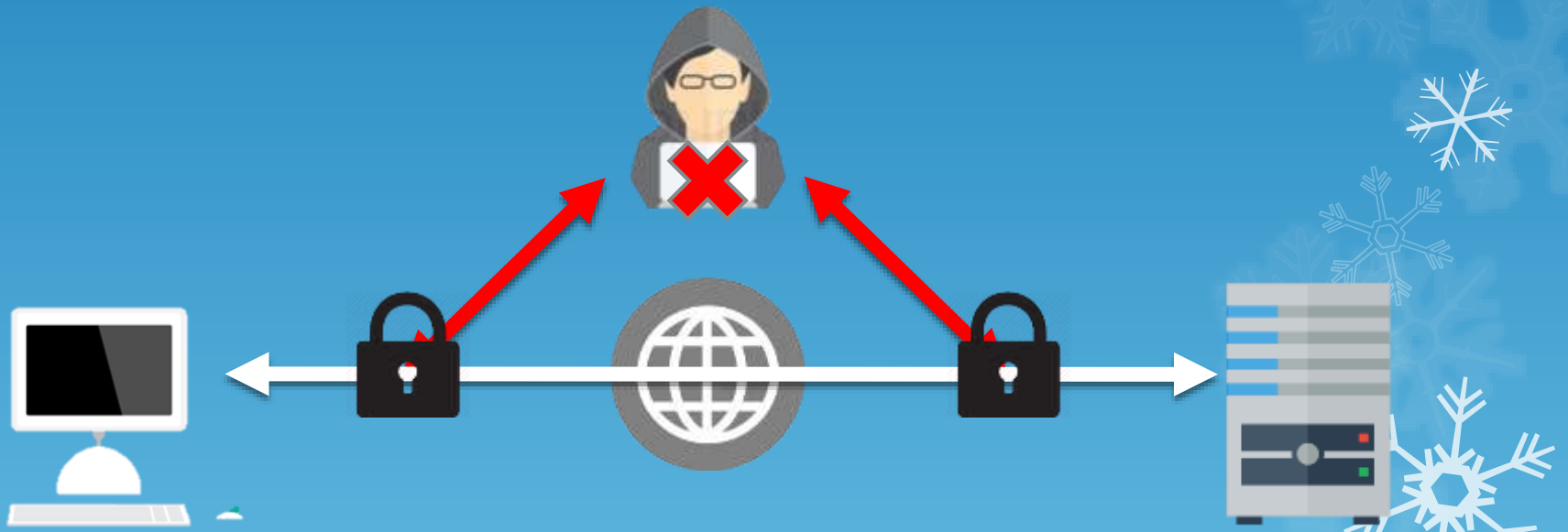
- Transport Layer Security (TLS) was designed to provide security at the transport layer.
- TLS was derived from a security protocol called Secure Sockets Layer (SSL).

Transport Layer Security (TLS)

- TLS is the successor to the Secure Sockets Layer (SSL).
- Transport Layer Security (**TLS**) is a protocol that ensures privacy between communicating applications and their users on the Internet.
- Is a widely deployed protocol for securing client-server communications over the internet.
- TLS is designed to prevent eavesdropping, tampering, and message forgery

Why do we need it?

- TLS ensures that no third party may eavesdrop or tamper with any message.

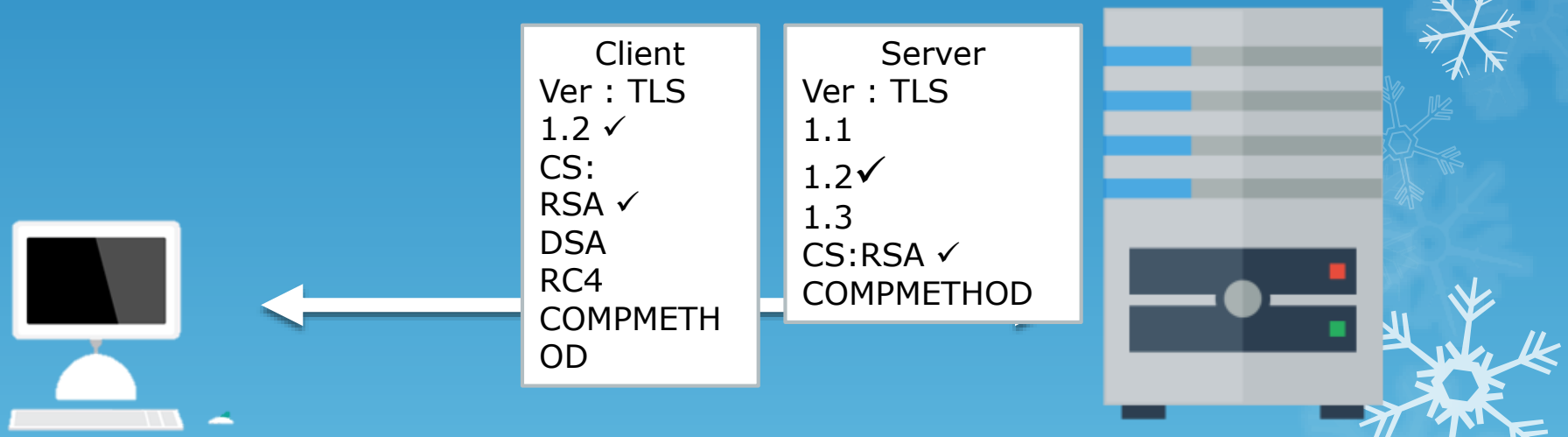


Working of Transport Layer Security

- The Client connect to server (using TCP). The client can be anything.
- The Client sends a number of specifications :
  - Version of SSL/TLS
  - Which cipher suites, compression method it wants to use.

Working of Transport Layer Security

- The server checks what the highest SSL/TLS version is that is supported by them both, picks a cipher suite from one of the client's options (if it supports one), and optionally picks a compression method.



Working of Transport Layer Security

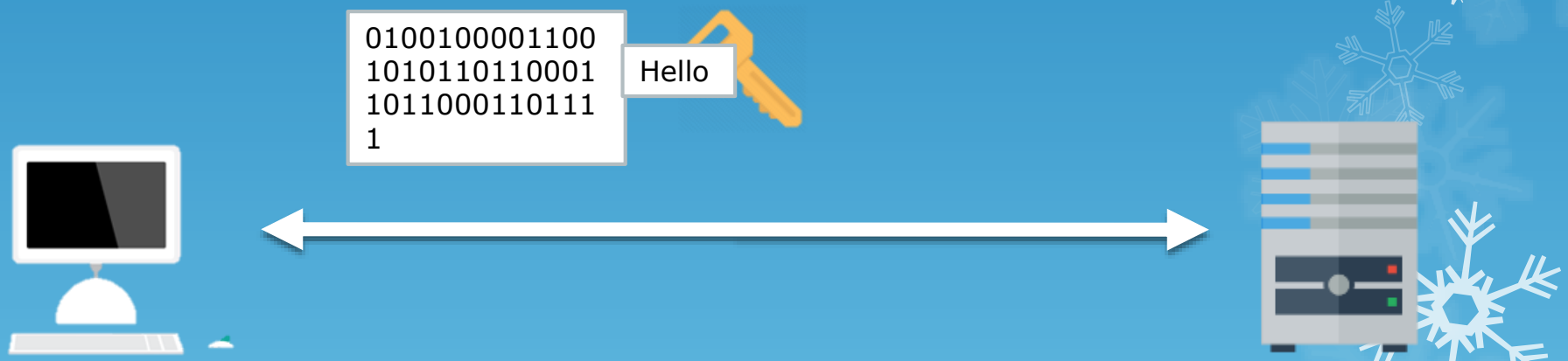
- After this the basic setup is done, the server sends its certificate.
- This certificate must be trusted by either the client itself or a party that the client trusts.
- For example if the client trusts GeoTrust, then the client can trust the certificate from Google.com, because GeoTrust cryptographically signed Google's certificate.

Working of Transport Layer Security

- Having verified the certificate and being certain this server really is who he claims to be (and not a man in the middle), a key is exchanged.
- This can be a public key, a "PreMasterSecret" or simply nothing, depending on the chosen ciphersuite.

Working of Transport Layer Security

- Both the server and the client can now compute the key for the symmetric encryption.



Working of Transport Layer Security

- The handshake is now finished, and the two hosts can communicate securely.

Working of Transport Layer Security

- To close the connection, a close notify 'alert' is used. If an attacker tries to terminate the connection by finishing the TCP connection (injecting a FIN packet), both sides will know the connection was improperly terminated. The connection cannot be compromised by this though, merely interrupted

Benefits of TLS\SSL

- Encryption
  - TLS can help to secure transmitted data using encryption.
- Interoperability
  - TLS works with most Web browsers, including Microsoft Internet Explorer and Netscape Navigator, and on most operating systems and Web servers.
- Algorithm flexibility
  - TLS provides options for the authentication mechanisms, encryption algorithms, and hashing algorithms that are used during the secure session.
- Ease of deployment
  - Many applications use TLS transparently on a Windows Server 2003 operating systems.
- Ease of use
  - Because you implement TLS beneath the application layer, most of its operations are completely invisible to the client.

IP security (IPSec)

The **IP security (IPSec)** is an Internet Engineering Task Force (IETF) standard suite of protocols between 2 communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted and authenticated packets. The protocols needed for secure key exchange and key management are defined in it.

Uses of IP Security –

IPsec can be used to do the following things:

- To encrypt application layer data.
- To provide security for routers sending routing data across the public internet.
- To provide authentication without encryption, like to authenticate that the data originates from a known sender.
- To protect network data by setting up circuits using IPsec tunneling in which all data is being sent between the two endpoints is encrypted, as with a Virtual Private Network(VPN) connection.

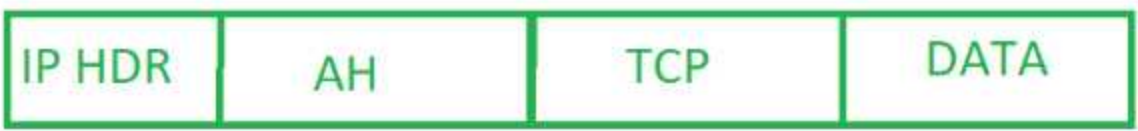
IPSec Components

1. **Encapsulating Security Payload (ESP) –**
   It provides data integrity, encryption, authentication and anti replay. It also provides authentication for payload.

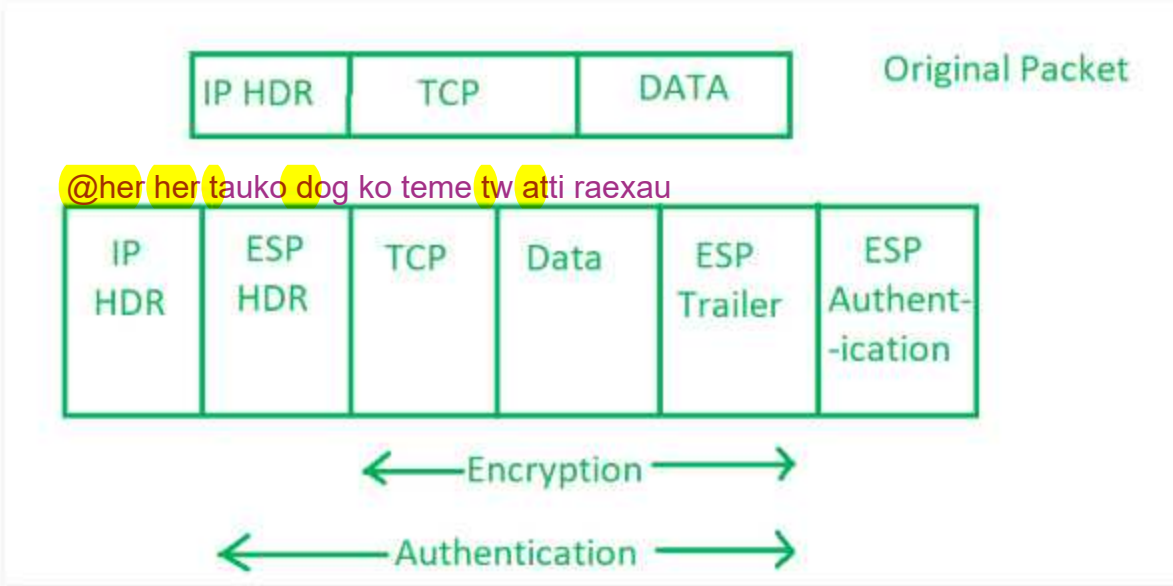
yo euta protocol ho ani yo protocol le chai message lai encrypt garne ani data integrity,confidentiality authentication provide garxa

2. **Authentication Header (AH) –**
   It also provides data integrity, authentication and anti replay and it does not provide encryption. The anti replay protection, protects against unauthorized transmission of packets. It does not protect data's confidentiality.

authenticate garyo vne jo paye tei le ma sender ho vnerw pathaunae mildenw so yo protocol ko kaam nae tei ho message replay huna nadine



3. **Internet Key Exchange (IKE) –**
   It is a network security protocol designed to dynamically exchange encryption keys and find a way over Security Association (SA) between 2 devices. The Security Association (SA) establishes shared security attributes between 2 network entities to support secure communication. The Key Management Protocol (ISAKMP) and Internet Security Association which provides a framework for authentication and key exchange. ISAKMP tells how the set up of the Security Associations (SAs) and how direct connections between two hosts that are using IPsec.



Working of IP Security –

suruma check grxa if packet haru IPSEC use gaerw safely transmit
garna milxa ki mildenw ani IKEphase 1 ma euta mode choose grxa tunel rw transport ma
(copy ko herne) ani cryptographic algorithms rw key choose grxa use grne wala
ani aba secure communication huxna duiwota bich ma

1. The host checks if the packet should be transmitted using IPsec or not. These packet traffic triggers the security policy for themselves. This is done when the system sending the packet apply an appropriate encryption. The incoming packets are also checked by the host that they are encrypted properly or not.

2. Then the **IKE Phase 1** starts in which the 2 hosts(using IPsec) authenticate themselves to each other to start a secure channel. It has 2 modes. The **Main mode** which provides the greater security and the **Aggressive mode** which enables the host to establish an IPsec circuit more quickly.
3. The channel created in the last step is then used to securely negotiate the way the IP circuit will encrypt data accross the IP circuit.
4. Now, the **IKE Phase 2** is conducted over the secure channel in which the two hosts negotiate the type of cryptographic algorithms to use on the session and agreeing on secret keying material to be used with those algorithms.
5. Then the data is exchanged across the newly created IPsec encrypted tunnel. These packets are encrypted and decrypted by the hosts using IPsec SAs.
6. When the communication between the hosts is completed or the session times out then the IPsec tunnel is terminated by discarding the keys by both the hosts.

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is an email security protocol is which was created by Phil Zimmerman in 1991. It was made available free of charge and became quite popular for personal use. The initial PGP protocol was proprietary and used some encryption algorithms with intellectual property restrictions. Subsequently, OpenPGP was developed as a new standard protocol based on PGP version 5.x.

- *Users generate their own OpenPGP public and private keys and then solicit signatures for their public keys from individuals or organizations to which they are known.*
- *An OpenPGP public key is trusted if it is signed by another OpenPGP public key that is trusted by the recipient. This is called the Web-of-Trust.*
- *OpenPGP does not include the sender's public key with each message, so it is necessary for recipients of OpenPGP messages to separately obtain the sender's public key in order to verify the message.*

Operational Description

esto protocol raexa jsle various algorithms use grne raexa

The actual operation of PGP, as opposed to the management of keys, consists of four services: authentication, confidentiality, compression, and e-mail compatibility.

tyo wordfile ko diagramni banaidina milxani pratek ma authentication ,encryptionma

authentication

Confidentiality

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed for storage or transmission using ZIP.
E-mail compatibility	Radix-64 conversion	To provide transparency for e-mail applications, an encrypted message may be converted to an ASCII string using radix-64 conversion.

this is usually done before encryption

Authentication

1. The sender creates a message.
2. SHA-1 is used to generate a 160-bit hash code of the message.
3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
4. The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
5. The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

CONFIDENTIALITY

Another basic service provided by PGP is confidentiality, which is provided by encrypting messages to be transmitted or to be stored locally as files.

- The sender generates a message and a random 128-bit number to be used as a session key for this message only.
- The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.
- The session key is encrypted with RSA using the recipient's public key and is prepended to the message.
- The receiver uses RSA with its private key to decrypt and recover the session key.
- The session key is used to decrypt the message.

Confidentiality and authentication (optional): First, a signature is generated for the plaintext message and prepended to the message. Then the plaintext message plus signature is encrypted using CAST-128 (or IDEA or 3DES), and the session key is encrypted using RSA (or ElGamal). This sequence is preferable to the opposite: encrypting the message and then generating a signature for the encrypted message.

COMPRESSION SSL mani encryption grnu vnda agadi compression garinxa

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space both for e-mail transmission and for file storage.

EMAIL COMPATIBILITY

When PGP is used; at least part of the block to be transmitted is encrypted. If only the signature service is used, then the message digest is encrypted (with the sender's private key). If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time symmetric key).

Thus, part or the entire resulting block consists of a stream of arbitrary 8-bit octets. However, many electronic mail systems only permit the use of blocks consisting of ASCII text. To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters.

The scheme used for this purpose is radix-64 conversion. Each group of three octets of binary data is mapped into four ASCII characters. This format also appends a CRC to detect transmission errors.

General Operational Overview of PGP

Notations:

K_s = session key used in symmetric encryption scheme
PR_a = private key of user A, used in public-key encryption scheme
PU_a = public key of user A, used in public-key encryption scheme
EP = public-key encryption
DP = public-key decryption
EC = symmetric encryption
DC = symmetric decryption

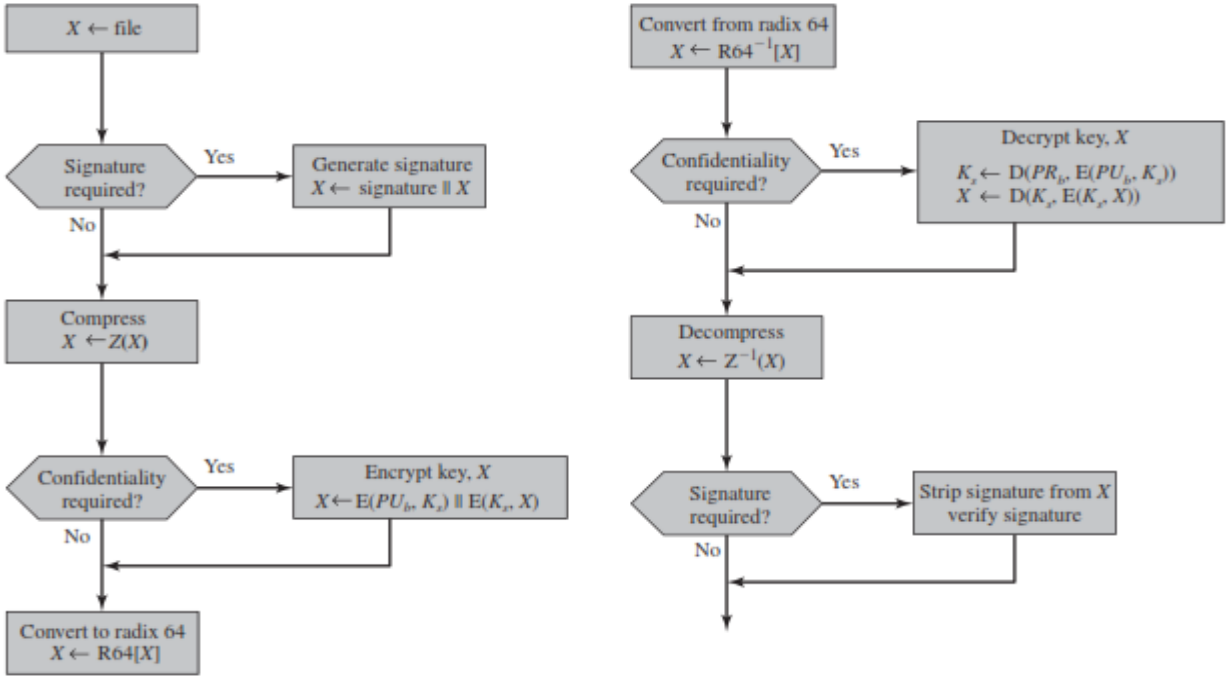


Figure: Transmission and reception of PGP messages

Firewall

A firewall is a system that prevents un-authorized access to or from a private network. It examines each message entering and leaving the network, and allows only those authorized messages to pass through. It can be implemented in hardware, software or both. It provides a line of defence against people or programs (including viruses and worms) that try to connect to your computer without invitation.

Why Firewalls?

It allows only authorized access to inside network. It prevents insider attacks on critical systems. A firewall as a barrier, checks information coming from the Internet or a network and allows it to pass through to your computer, depending on your firewall settings. It provides the means for implementing and enforcing the network access policy. In effect, firewall provides access control to users and services. It provides the ability to control access to site system. It can greatly improve network security and reduce risks to hosts on the subnet by filtering inherently insecure services.

How does it work?

When someone on the Internet or a network tries to connect to your computer, we call that attempt an "unsolicited request." When your computer gets an unsolicited request, Windows Firewall blocks the connection. If you run a program such as an instant messaging program or a multiplayer network game that needs to receive information from the Internet or a network, the firewall asks if you want to block or unblock (allow) the connection. If you choose to unblock the connection, Windows Firewall creates an exception so that the firewall won't bother you when that program needs to receive information in the future.

For example, if you are exchanging instant messages with someone who wants to send you a file (a photo, for example), Windows Firewall will ask you if you want to unblock the connection and allow the photo to reach your computer. Or, if you want to play a multiplayer network game with friends over the Internet, you can add the game as an exception so that the firewall will allow the game information to reach your computer.

Types of Firewall:

1. **Application Gateways**: The first firewalls were application gateways, and are sometimes known as proxy gateways. These are run with special software to act as a proxy server. This software runs at the application layer of OSI Model. Clients behind the firewall must be proxitized in order to use internet services.

2. **Packet Filtering:** Packet filtering is a technique whereby routers have ACLs (Access Control Lists) turned on. By default, a router will pass all traffic sent it, and will do so without any sort of restrictions. There is fewer overloads in packet filtering than with an application gateway, because the feature of access control is performed at a lower OSI layer.

3. **Hybrid Systems**: In an attempt to marry the security of the application layer gateways with the flexibility and speed of packet filtering, some vendors have created systems that use the

principles of both. In some of these systems, new connections must be authenticated and approved at the application layer. Other possibilities include using both packet filtering and application layer proxies.