U-IV DIGITAL SIGNATURES, CERTIFICATES & STANDARDS

- ✓ Due to problems associated with MAC, Digital signature standard (DSS) was developed for digitally signing the document or certificates
- ✓ NIST (National Institute of standard Technology) published DSS standard as FIPS
- ✓ FIPS revised in 1993 and 1996. DSS makes use of SHA-1 algorithm for calculating the message digest of an original message and uses message digest to perform digital signature
- ✓ DSS make the use of algorithm called digital signature algorithm (DSA)
- ✓ Similar to RSA, DSA is also based on asymmetric key cryptography. However their objectives are totally different
- ✓ As we know RSA is primarily used for encrypting the message but we can use RSA to produce digital signature
- ✓ DSA can only be used to perform digital signature , it cannot be used for encryption

RSA and Digital Signature:

Lets assume sender **A** wants to send a message **M** to receiver **B** along with digital signature **S** calculated over message **M** following steps occur for preparation of message

Step I:

Sender A uses SHA-1 Message digest algorithm for calculating the MD1 of original message M as shown below

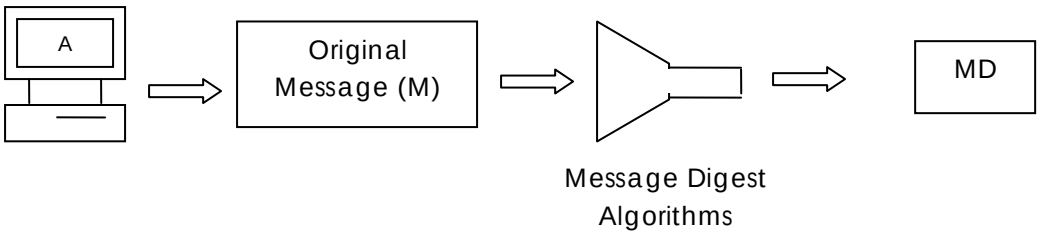


Fig. Message Digest Calculation

Step II:

Sender A now encrypts the MD with his private key and the output of this process is called digital signature (DS)

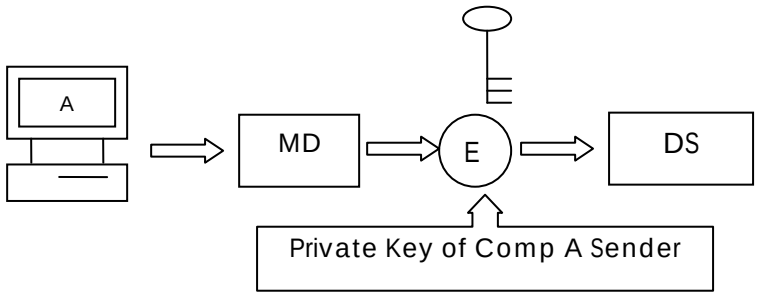


Fig. Digital Signature Creation

Step III:

Now sender A sends original message M along with digital signature DS to receiver B as shown below

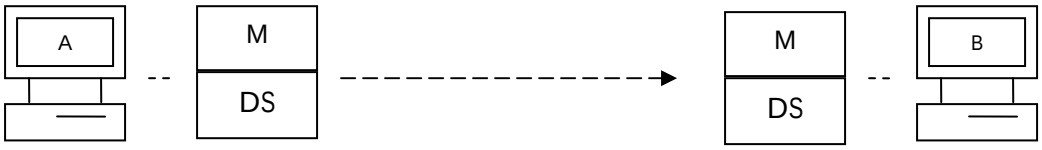


Fig. Transmission of original message & digital signature

Step IV:

- ✓ **B** receives original message (M) from **A** and digital signature DS
- ✓ **B** uses same message digest algorithm used by **A** and calculates MD2 of received message as shown below

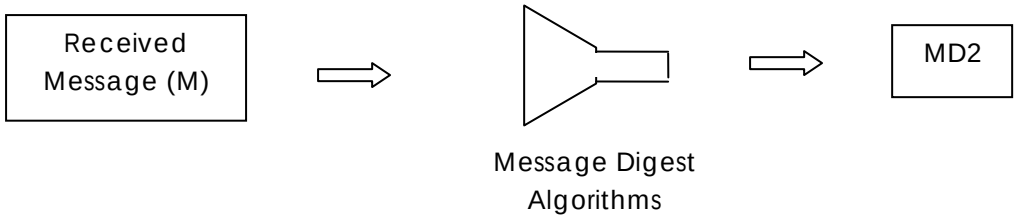


Fig. Message Digest Calculation at receiver

Step V:

- ✓ Now receiver uses A's public key to decipher(decrypt) the digital signature
- ✓ Output of above step is the original message digest (MD1) calculated by A.

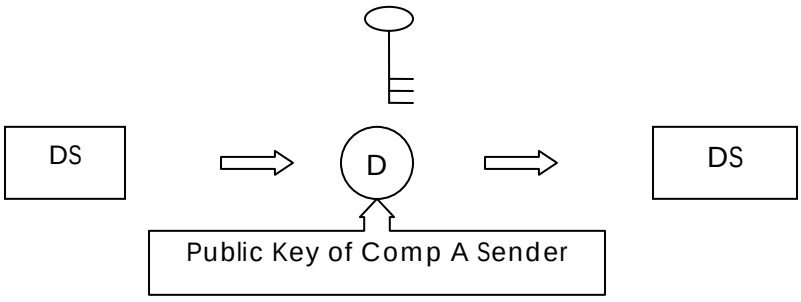


Fig. Receiver retrieves senders MD

Step V:

- ✓ B now compares two message digest i.e. MD2 –calculated in step-4 and MD1- retrieved from A's digital signature in step5
- ✓ If MD1=MD2 then – B accepts original message (M) as the correct unaltered message from A
- ✓ i.e. B is assured/confirmed that message came from A not from someone else posing as A

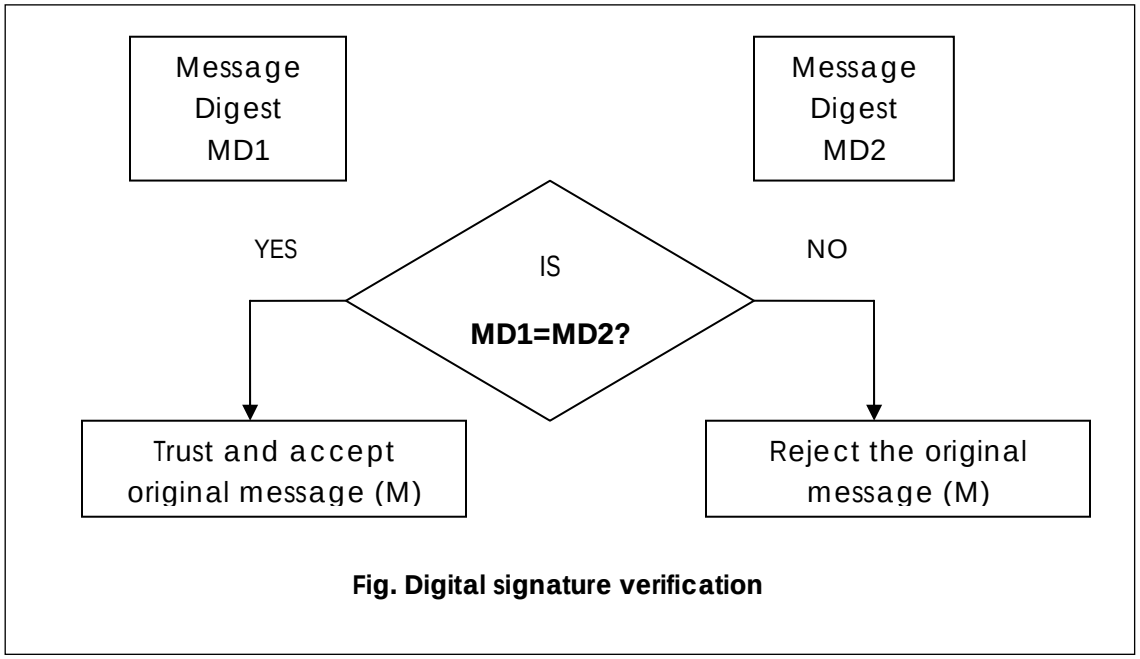


Fig. Digital signature verification

Q. Why attacker doesn't alter message and recalculate MD and sign it again?

ANS: Attacker can perform 2 steps very well (i.e. alter the message and recalculate the MD) but can not sign it again because for that attacker needs A's private key)

Since only A knows his private key , attacker cannot use A's private key to encrypt message digest (i.e. sign the message) again

Thus principle of digital signature is quiet strong secure and reliable

Digital signature Algorithm (DSA):

Description of DSA is mathematical and complicated

DSA algorithm makes uses of following variables

p = Prime no. of length l - bits

Where l – is multiple of 64 between 512 and 1024(i.e. l=512 or 576 or 640 ……..1024)

q=160 bit prime factor of (p-1)

g= $h^{(p-1)/q}$ mod p where h is a no. less than (p-1) such that $h^{(p-1)/q}$ mod p > 1

X= a number less than Q

Y= g^x mod p

H= message digest algorithm (SHA-1)

First 3-variables p, q and g are public in nature and can be sending across insecure network

Private Key is X where as public key is Y

Let's assume sender want to sign message M and sends signed message to receiver then following steps takes place

1. Sender generates a random no. k which is less than q
2. Sender calculates
   - (a) r=(g^k mod p) mod q
   - (b) s=(k^l (H(M)+Xr)) mod q

Values of r and s are signatures of sender the sender sends these values to receiver to verify signature the receiver calculates

- W= S^{-1} mod q
- U1= (H(m) * W) mod q
- U2= (r w) mod q
- V= ((g^{u1} * y^{u2}) mod p) mod q

If v=r sign said to be verified otherwise it is rejected

Digital certificates:

We have seen diffie and hellman algorithm for key exchange but its also having problem regarding man in middle attack

Solution for above problem is digital certificates

Conceptually we can compare digital certificate to the document such as our passport or driving license which proves our identity by specifying

- Name
- Nationality
- Date and place of birth
- Photograph and signature

Concept of digital certificate:

Digital certificate would be actually a computer file with name abc.cer. So similar to passport digital signature signifies the association between my public key and me

Digital certificates are issued by trusted parties, government authorities in which all the concerned parties have great amount of trust and belief

Imagine situation if our passport is issued by ordinary shopkeeper then no one trust that passport

As we mentioned digital signature establishes relation between user and his public key, therefore a digital certificate must contain user's public key and his name

Digital Certificate

Subject Name: XXXXXXX

Public Key: XXXXXXX

Serial No. : XXXXXXXX

Other Data: XXX

Valid for:

Valid to:

Issuer Name:

Certificate Authority (CA):

CA is trusted agency that can issue certificates. Government finalizes that who can acts as CA and who cannot. Usually CA is reputed organization such as post offices, financial institution and software companies etc

Two of worlds famous CA are Entrust and Verisign

Technical Details of digital certificate:

A standard called as X.509 defines the structure of digital certificate

The international telecommunication union (ITU) came up with this standard in 1988 at that time it was the part of another standard called as X.509

Since then X.509 as it was revised three times current version is called X.509V3

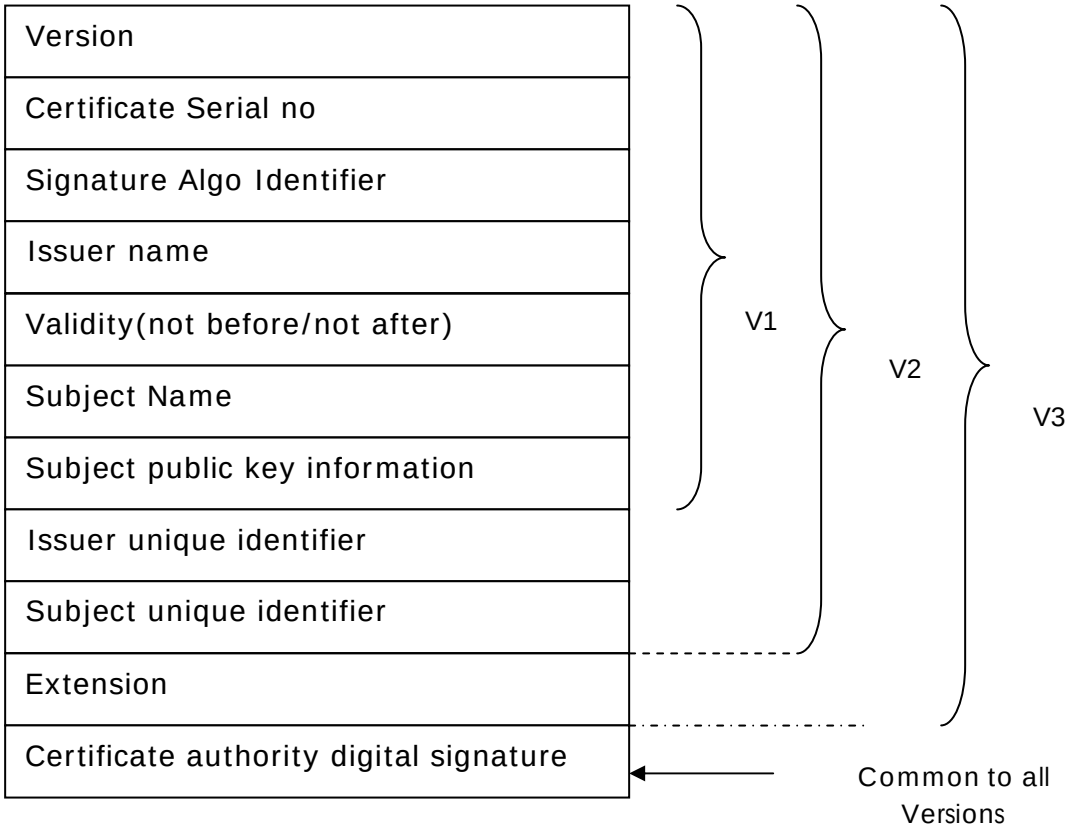


Fig. Contents of a digital certificate

Digital certificate creation:

Parties involved: - As we know mainly three parties involved in the process of digital certificate creation namely subject (end user) and the issuer – certification authority (CA). A third party is also optionally involved in certificate creation and management

Since CA can be overloaded with variety of task such as issuing new certificate maintaining the old one so CA can delegate some of its task the third party called as registration authority (RA) as shown

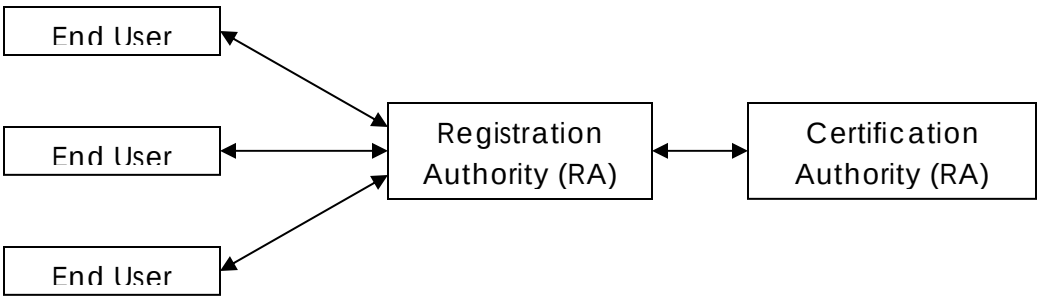


Fig. Registration Authority (RA)

RA acts as mediator between end user and CA it performs following task

1. Accept and verify registration information about new user
2. Generate keys on behalf of end user
3. Accept and authorizes request for key backup and recovery
4. Accept and authorize request for certificate revocation

Due to RA CA becomes isolated entity which makes it less susceptible to security attacks

Digital certificate creation Steps:

Certificate creation contains following steps outlines in fig

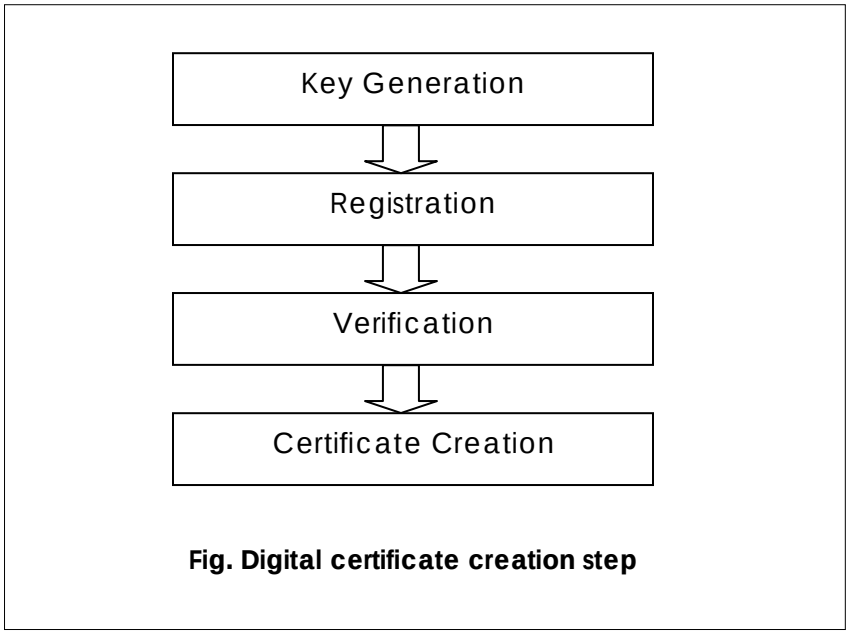
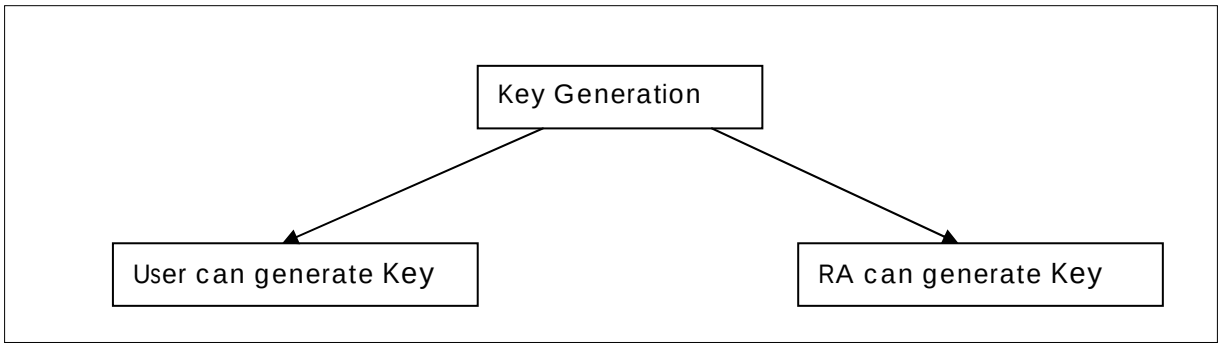


Fig. Digital certificate creation step

Step I: Key generation



As shown below user generates private and public key by interacting with software. After creating keys user keeps private key and diverts public key towards RA/CA

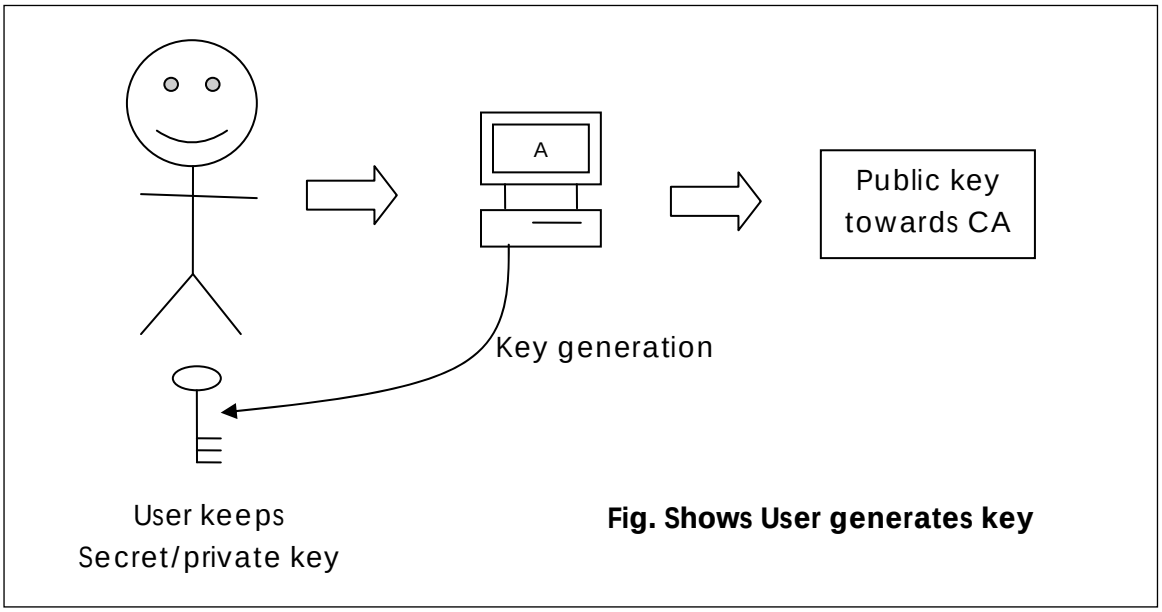


Fig. Shows User generates key

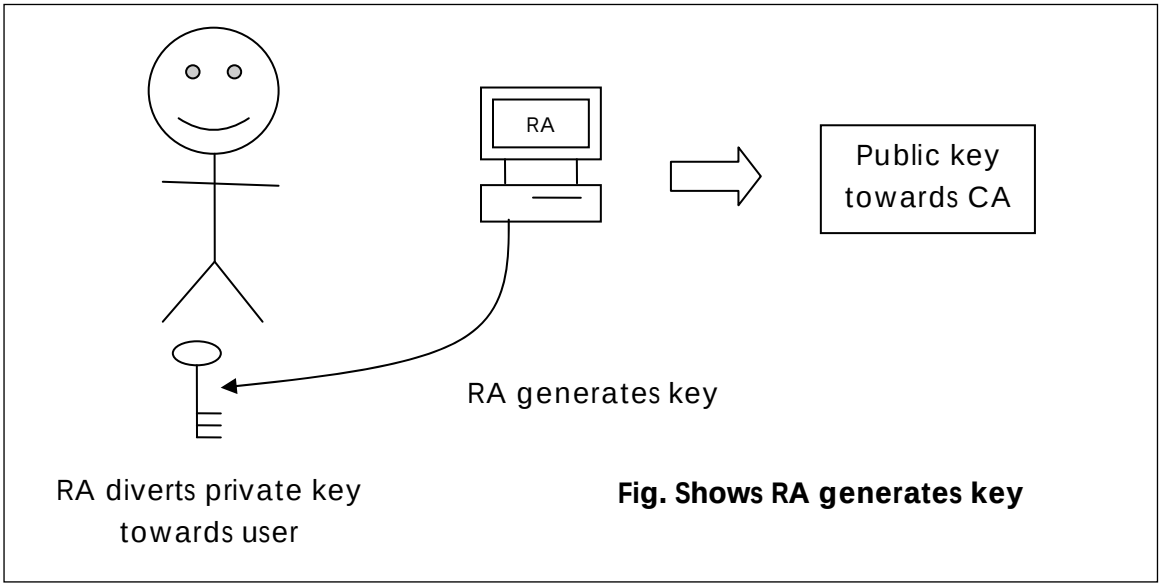


Fig. Shows RA generates key

Alternative is RA can generate key pair on subject behalf and transmits private key to concerned subject.

Here there is possibility of exposing private key during transfer from RA to end user (Simply this approach is less secure)

Step II: Registration

- This step requires only when user or subject generate key pair in first step
- User sends public key and registration information , all evidences about himself to RA by using software wizard or certificate signing request (CSR)
- Evidences however not in electronic usually it consist of POP based document e.g. PAN, passport etc

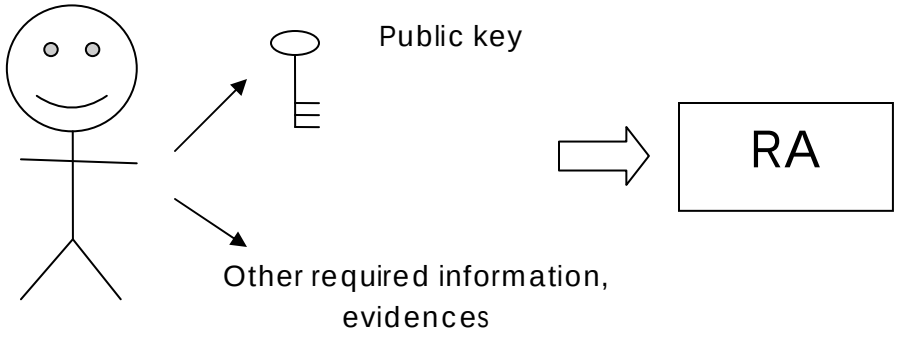


Fig. Subject sends public key and evidence

Step III: Verification

Now RA verifies the users credentials this is having two aspects

1. Verifies users credentials such as organization , business record , History
2. To ensure user who is requesting indeed posses the private key corresponding to public key or not

Above check is called proof of possession (POP) of private key for this RA can do following

1. Demand user digitally sign her certificate from that RA verify genuine /authentic user
2. RA can create random number challenge encrypt it with users public key and send encrypted challenge to the user. If user successfully decrypt the challenge RA assume user posses the right private key

3. RA can generate dummy certificate for user and encrypt it using users public key and send it to the user. User can decrypt it only if he is having correct corresponding private key and he can obtain plaintext certificate

Step IV: Certificate creation

RA passes on all the details of user to CA

CA does its own verification if required

CA create certificate by using program in X.509 standard format

CA send certificate to user as well maintain the copy of certificate in a certificate directory LDAP(light weight directory access protocol)this is central storage location maintained by the CA (Certification authority) it allows user and applications to access X.500 directories depending on their privileges CA send certificate through email attachment or sends email

Why to trust on Digital Certificate?

- As digital certificate is simple computer file in a specific standard format so any one can produce it
- Can we trust on a file of specified format which is only having important information regarding users public key signed by any authority
- Obviously not we cannot trust digital certificate on above ground
- We can trust the certificate if its signed by trusted authority or trusted party (CA) who always signs a digital certificate with his own private key
- Trusted party gives assurance that I've signed this certificate to guarantee that this user posses the specified public key
- So simply trust on the digital certificate as CA gives you guarantee

How Does CA sign a digital Certificate? (Digital certificate signing process)

- CA signs certificate with his own private key

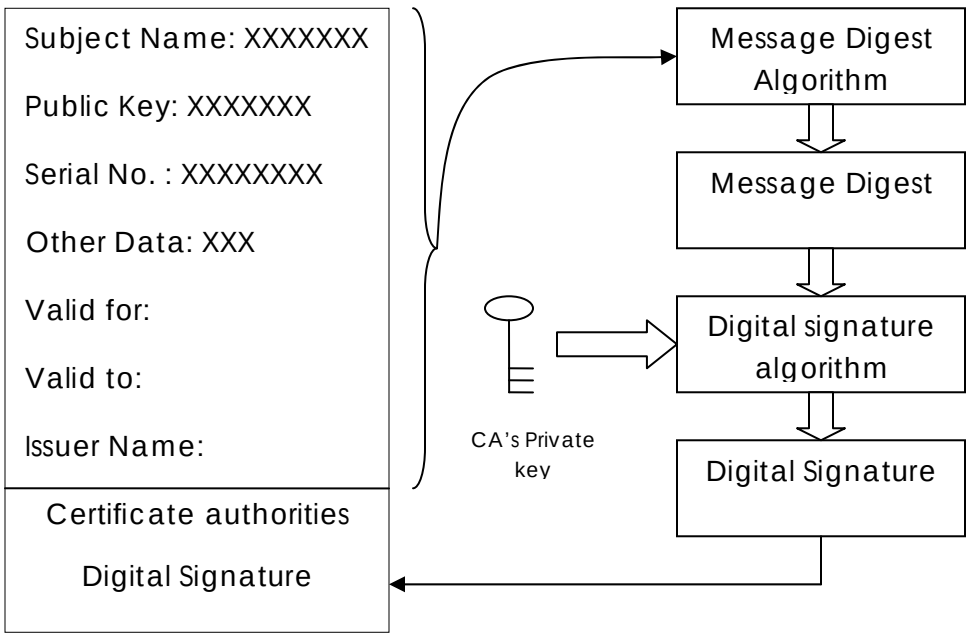


Fig. Certificate signing process

- As shown content of certificate is treated as message and passed to MD algorithm like SHA-1
- Output of MD algorithm is Message Digest(MD)
- MD is encrypted by using certificate authorities private key to produce digital signature
- At the end digital signature of certificate authority is stored as last field of digital certificate

Digital Certificate / signature Verification:

Consider we have received digital certificate of user and interested to verify it. What should we do for this?

Clearly we need to verify digital signature of CA

For verification of signature we have to follow the steps shown in the following block diagram

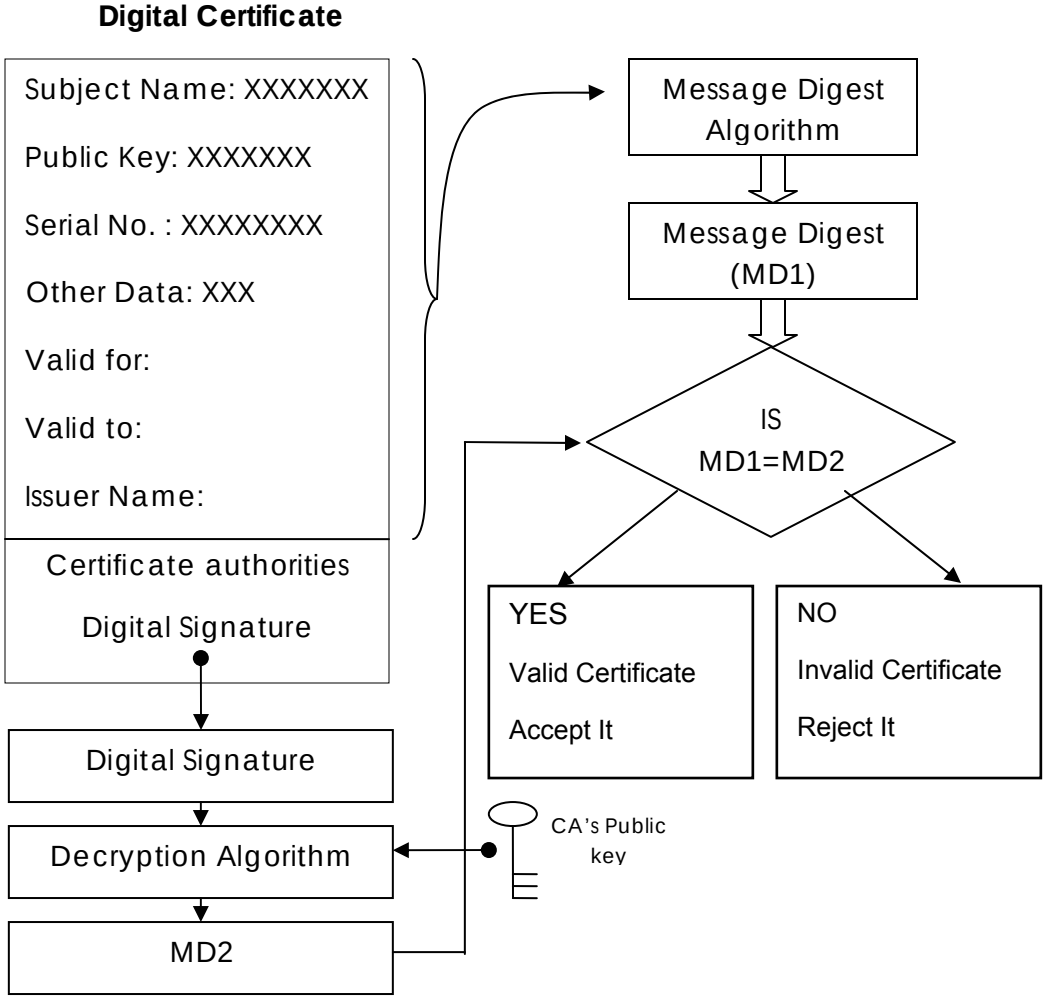
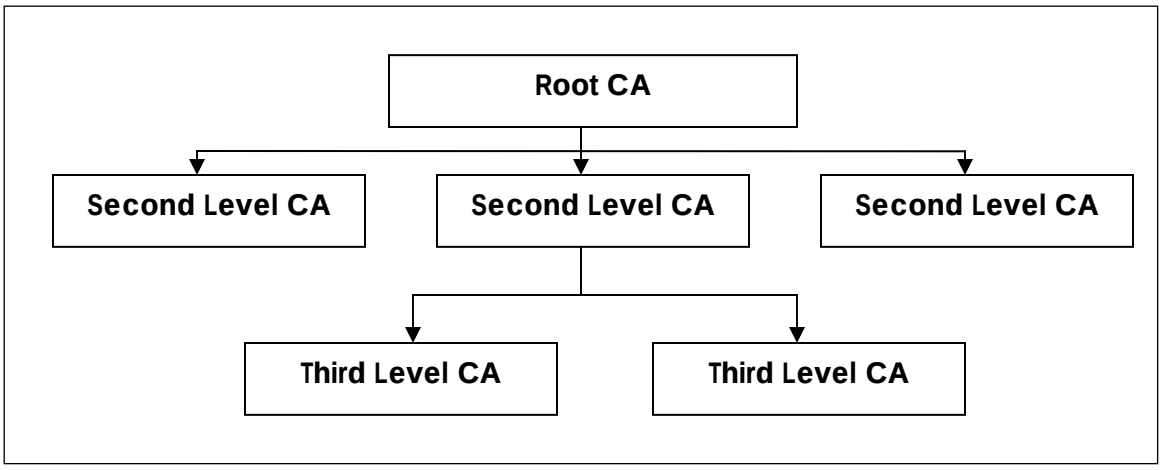


Fig. Certificate Verification

- All the fields except last one(digital signature) of received digital certificate passed to Message Digest algorithm
- MD algorithm calculates MD1
- Now user extracts digital signature of CA from certificate
- User Deciphers CA's signature by using CA's public key
- This produces another message digest call it as MD2
- Now user compares MD1 with MD2 if found match MD1=MD2 then user get convinced that certificate is signed by CA otherwise user will not trust the certificate and rejects it

Certificate Hierarchy:



- Security of certificate can be increased by increasing the level of hierarchy of CA's
- As shown root CA will act as MD i.e. the highest authority of certification
- Then at second level there are many managers reporting to root CA
- Mat peoples are there at third level reporting to the managers at second level and so on
- Purpose of creating hierarchy is just to relieve MD or CEO 's to perform all types of task in all departments

Kerberos :(Network authentication protocol)

- Kerberos is network authenticator protol used in many real-time system
- Kerberos is based on another protocol called as Nedham-Shroeder
- Desigend at MIT in 1980
- Available as open source or supported by commercial software's
- Kerberos signify a multithreaded dog as per Greek mythology

Why Kerberos?

- Sending username and password in clear text may cause problem to security
- If each time password is send in clear there is chance for interception
- So to resolve and sort out above problem Kerberos is needed

Firewall VS Kerberos:

- Firewalls make a risky assumption: those attackers are coming from the outside. In reality, attacks frequently come from within.
- Kerberos assumes that network connections (rather than servers and work stations) are the weak link in network security

Design requirements for Kerberos:

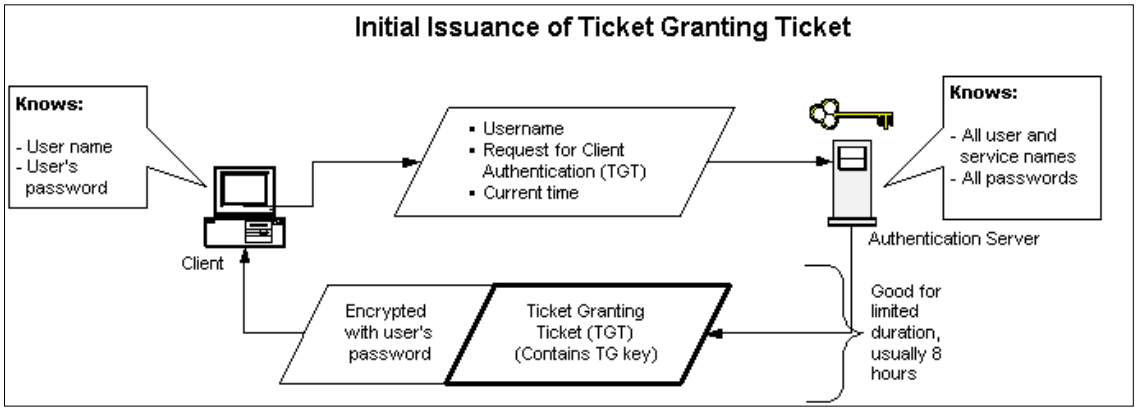
- Interactions between hosts and clients should be encrypted.
- Must be convenient for users (or they won't use it).
- Protect against intercepted credentials.
- Private Key: Each party uses the same secret key to encode and decode messages.
- Uses a trusted third party which can vouch for the identity of both parties in a transaction. Security of third party is imperative.

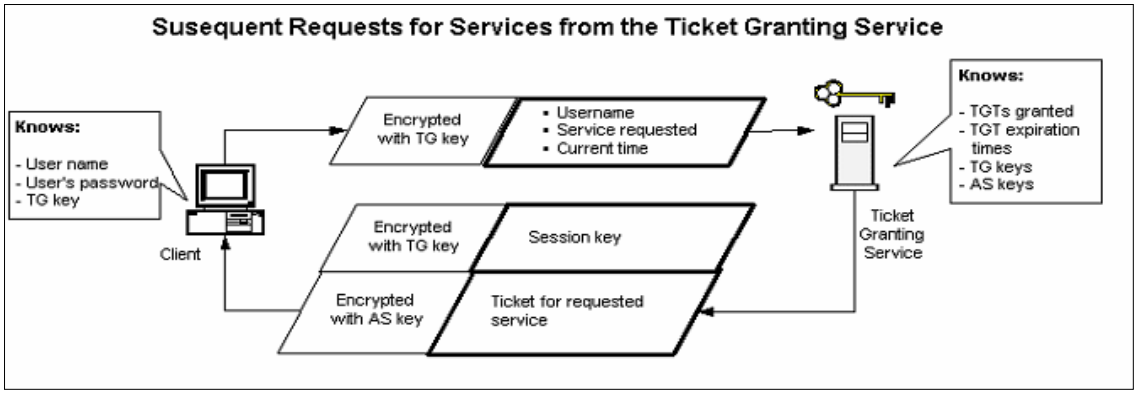
Working of Kerberos:

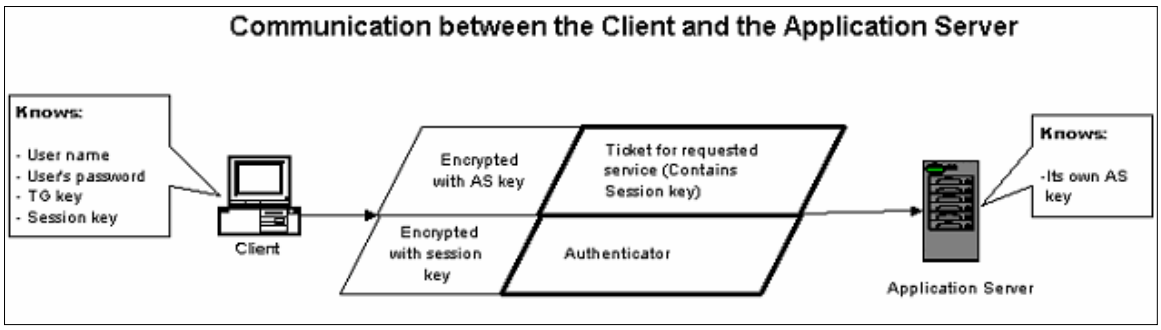
- Instead of client sending password to application server:
  - Request **Ticket** from authentication server
  - Ticket and encrypted request sent to application server

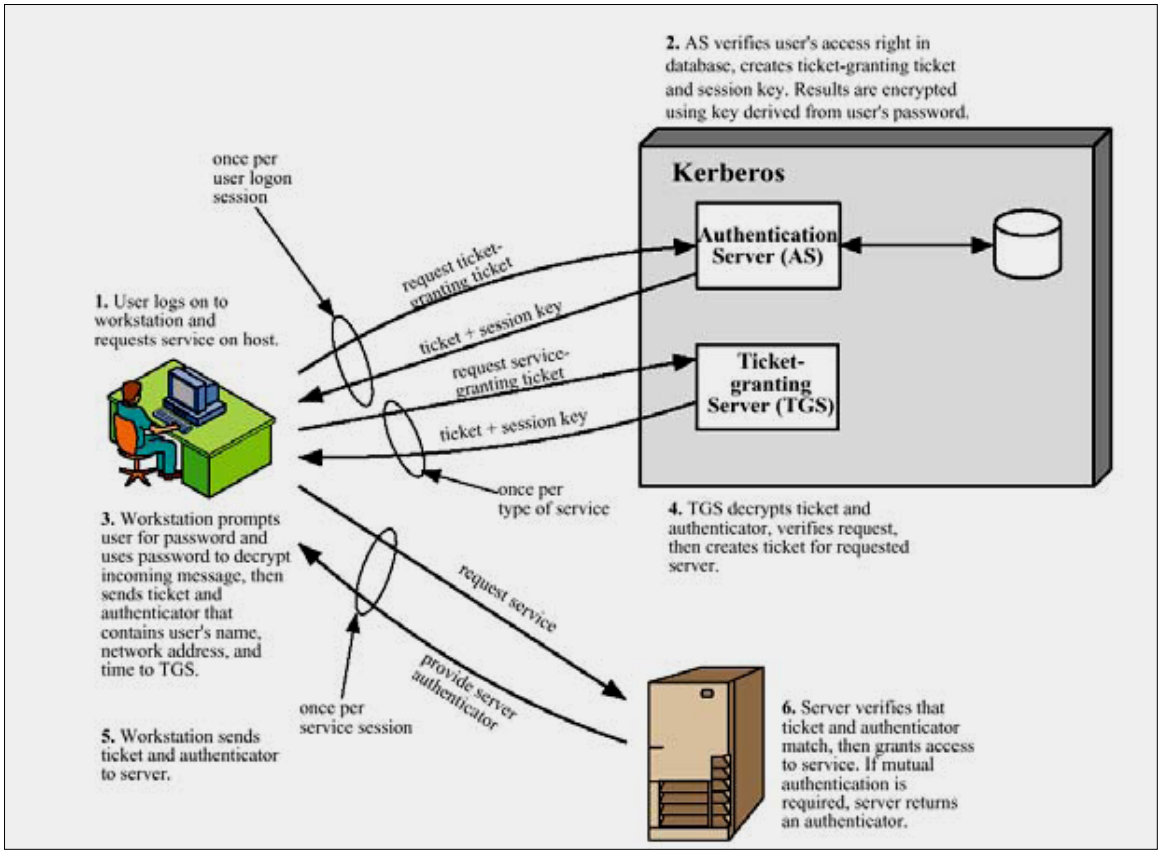
Applications

1. Authentication
2. Authorization
3. Confidentiality
4. Within Network and small set of Networks









Version 4 problem

Problems:
– Lifetime associated with the ticket-granting ticket:
– If too short → the user is repeatedly asked for the password
– If too long → a greater opportunity to replay exists.
The threat is that an opponent will steal the ticket and use it before it expires.
- Inter realm authentication is not possible in V4

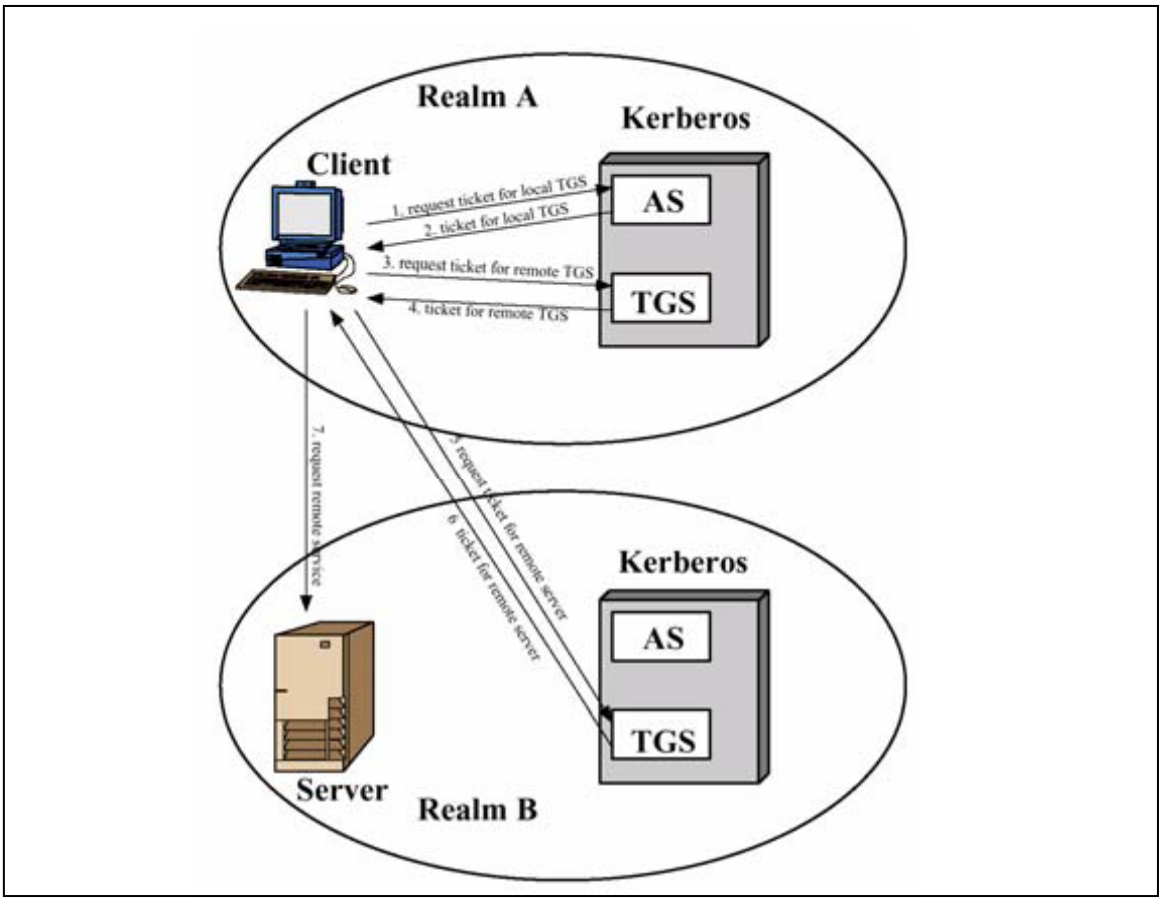


Fig. Shows Request for service in another realm (Kingdom)

Difference between Version 4 and 5

- Encryption system dependence (v.4 DES with non standard PCBC, v.5 you can choose the encryption algorithm and use CBC)
- Internet protocol dependence (v.4 only IP; v.5 any type)
- Message byte ordering (v.4 arbitrary; v.5 defined by ASN1 Standard)
- Ticket lifetime (v.4 21h max; v.5 arbitrary)
- Authentication forwarding to other hosts (v.4 no; v.5 yes) (A client issues a request to a print server that then accesses the client's file from a file server, using the client's credentials for access.)
- Inter-realm authentication: v.4 N2 (!) realm to realm relationships (v5. simpler)

Kerberos V. 5

- V5 : allows inter-realm authentication with less overhead than v. 4
- Kerberos v5 is an Internet standard
- specified in RFC1510, and used by many utilities To use Kerberos:
- you need a KDC on your network
- you need to have "Kerberised" applications running on all participating systems

X.509 Authentication Service:

- Part of CCITT X.500 directory service standard
- Distributed servers maintain user information in database
- Defines framework for authentication service
- Directory may store public key certificate with public key of user
- Signed with private key by certification authority
- Defines authentication protocol
- Uses public key cryptography and digital signature
- Used in a variety of context like
    1. S/MIME
    2. IP Security
    3. SSL/TLS, SET protocols

Authentication procedure:

X.509 include three alternative authentication procedures

1. One way authentication
2. Two way authentication
3. Three way authentication

All above authentication procedure uses public key signatures

One-way authentication:

- 1-message (A->B) used to establish
- Message includes identity of A and that message is from A
- Message was intended for B
- Integrity and originality of message i.e. message must include timestamp, nonce, B's identity and is signed by A

Two way authentication:

- Two messages (A->B and B->A) which also establishes in addition
- Identity of B and reply is from B
- Reply is intended for A
- Integrity and originality of reply
- Reply includes original nonce from A , also timestamp and nonce from B

Three way authentication:

3 messages (A->B, B->A, A->B) which enables above authentication possible without Synchronized clock possible

i.e. timestamp need not be checked or relied upon

E-mail Security:

- E-mail is most widely used application on Internet
- Using e-mail user can send messages(pictures, sound
- Due to wide uses of e-mail security of it is major important issue
- RFC822 defines the format of text e-mail message. An e-mail message is considered to be made up of two portion its content (body) and header i.e. too similar to our normal postal system

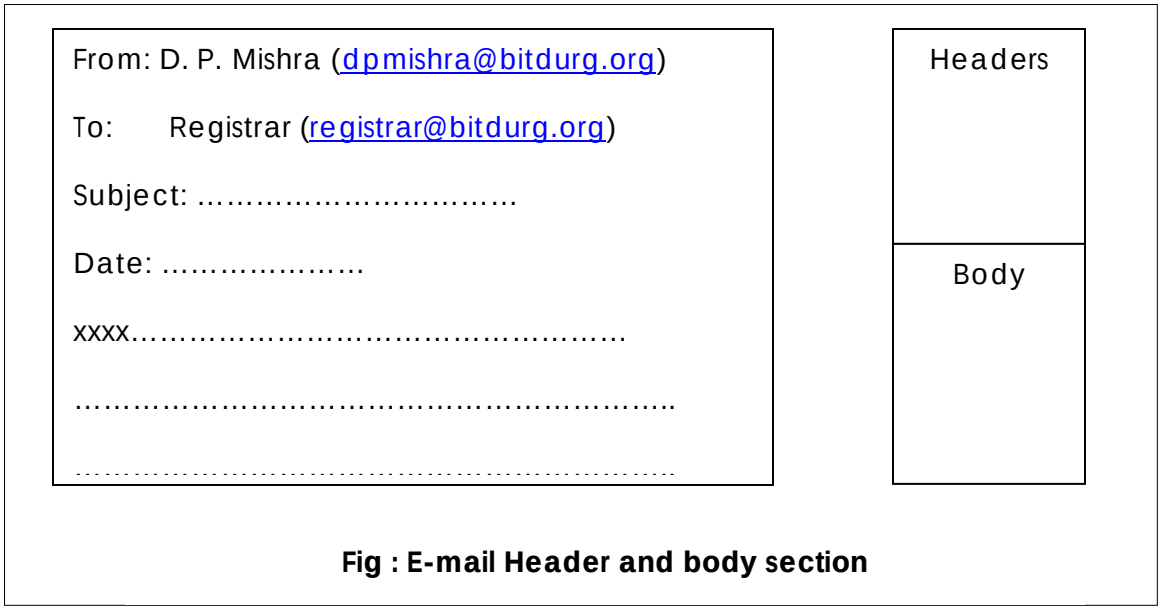


Fig : E-mail Header and body section

SMTP (Simple mail transfer protocol) is used for e-mail communication. The e-mail client software at sender end gives message to local SMTP server and this local SMTP server transfer's message to receivers SMTP. Main job is to carry mail messages between sender and receiver

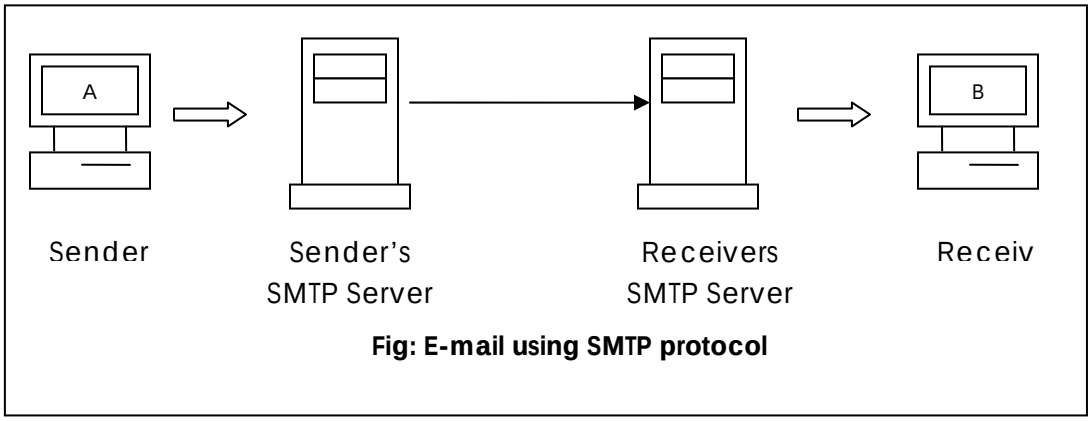


Fig: E-mail using SMTP protocol

It uses TCP/IP protocol underneath i.e. SMTP runs on the top of TCP/IP (in application layer)

Following are the main three e-mail security protocols

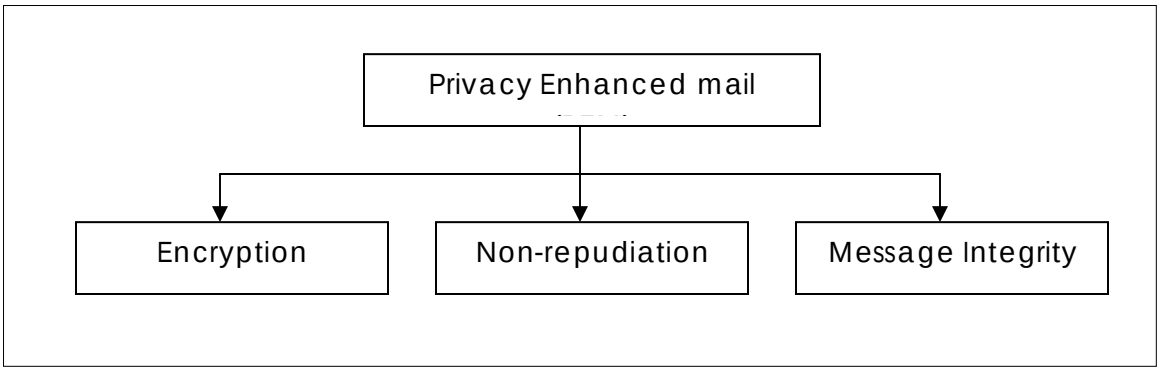
1. Privacy enhanced mail (PEM)
2. Pretty good privacy(PGP)
3. Secure MIME (S/MIME) multipurpose internet mail extensions

Privacy enhanced mail (PEM):

Internet e-mail security standard adopted by internet architecture board (IAB) to provide secure electronic mail communication over the Internet

PEM was initially developed by Internet Research task force (IRTF) and privacy security research group (PSRG) they then handed over PEM to Internet Engineering task force (IETF)

PEM supports main three cryptographic functions



Working of PEM (Privacy enhanced mail):

- Broad level steps of PEM are shown in following fig

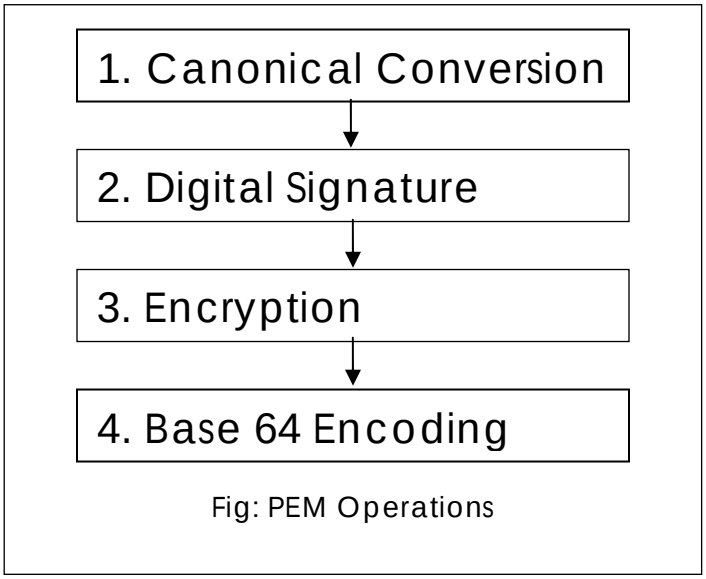


Fig: PEM Operations

Above steps are performed at sender end and at receiver end above steps are performed in reverse order i.e. 4,3,2,1

Step-I Canonical conversion:

- Since there is possibility that computers used by sender and receiver are not of same architecture and operating system
- So there is possibility that some content or message would be represented differently on different computers
- E.g. in MS-DOS enter key is represented by two characters while in UNIX enter key is represented by 1-character
- In order to maintain appearance symmetry of message on different machines PEM transform each e-mail message to abstract canonical representation i.e. message is converted to uniform and architecture independent format

Step-II Digital Signature:

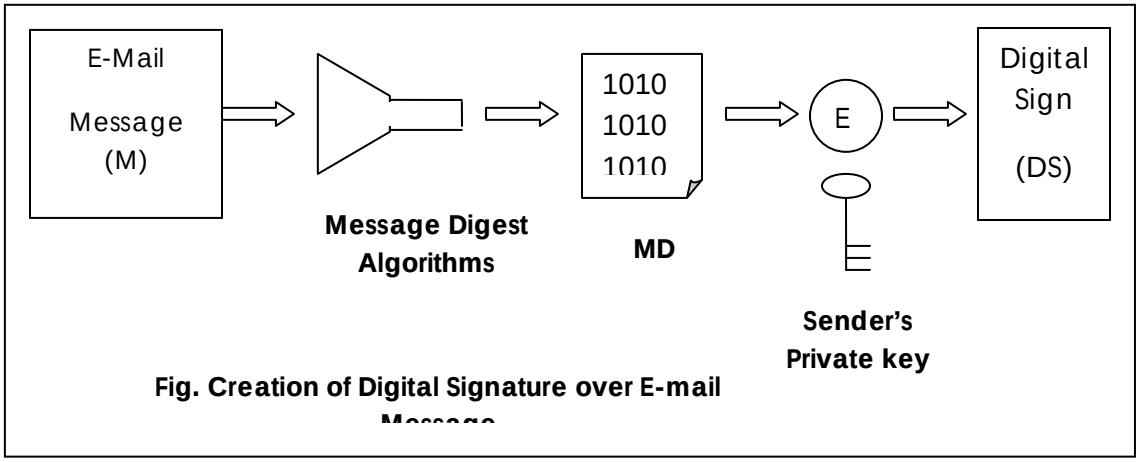


Fig. Creation of Digital Signature over E-mail Message

- As shown in above block diagram e-mail message is passed through message digest algorithm to generate MD
- MD is encrypted by using senders private key to generate digital signature equivalent to e-mail message
- Digital Signature and message is combined and encrypted by using symmetric key as shown in next block diagram

Step-III Encryption:

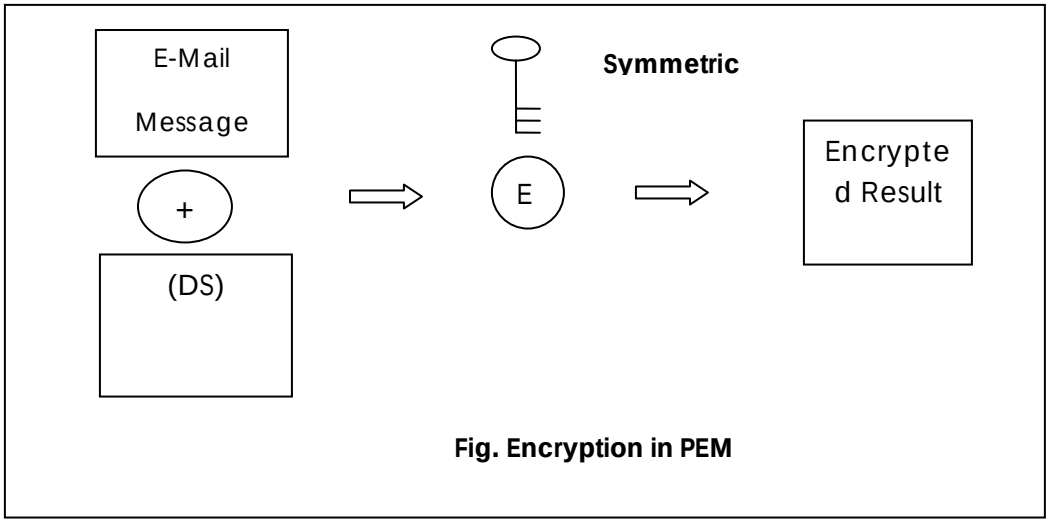


Fig. Encryption in PEM

As shown e-mail message along with DS is encrypted by using symmetric key

Step-IV Base-64 Encoding:

This is the last step in PEM, base64 encoding is (also called as radix-64 encoding or ASCII armor) process transform binary input into portable character

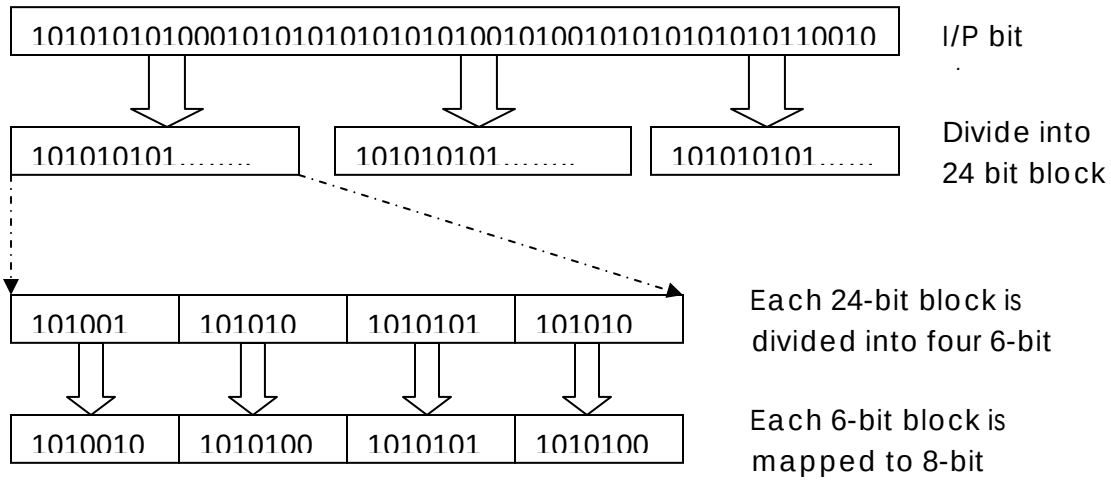


Fig. Base 64 Encoding Concept

- As shown in above block diagram output of step –III i.e. CT is treated as input stream
- Input stream is divided into 24 bit block
- Each 24-bit block is divided into four 6-bit block
- Each 6-bit block is further mapped to 8-bit block to produce the final result of BASE-64 Encoding

Pretty Good Privacy (PGP):

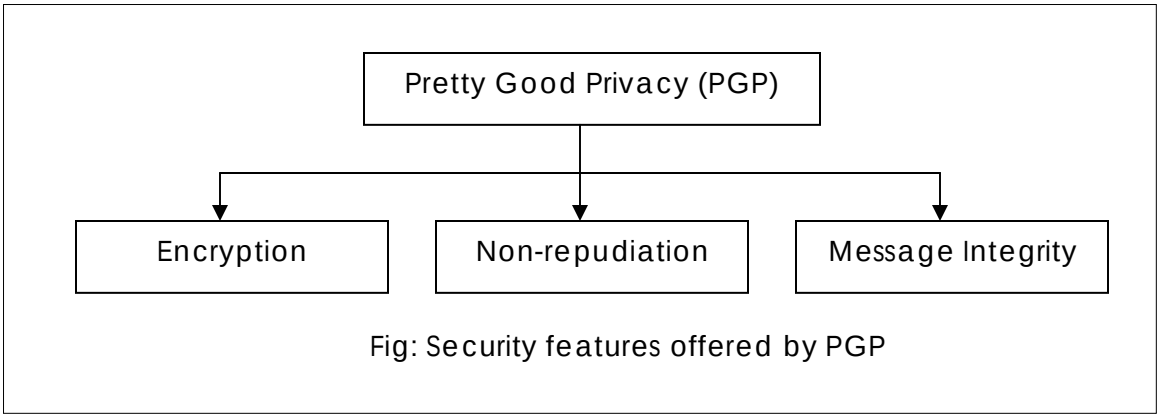


Fig: Security features offered by PGP

- Developed by Phil Zimmerman
- Supports the basic requirement of cryptography
- Simple to use and completely free including its source code
- Algorithm is supported by PGP are RSA, DSS, CAT, IDEA and DES-III
- PGP is more popular and widely used as compared to PEM
- Broad level steps of PGP are shown in following fig

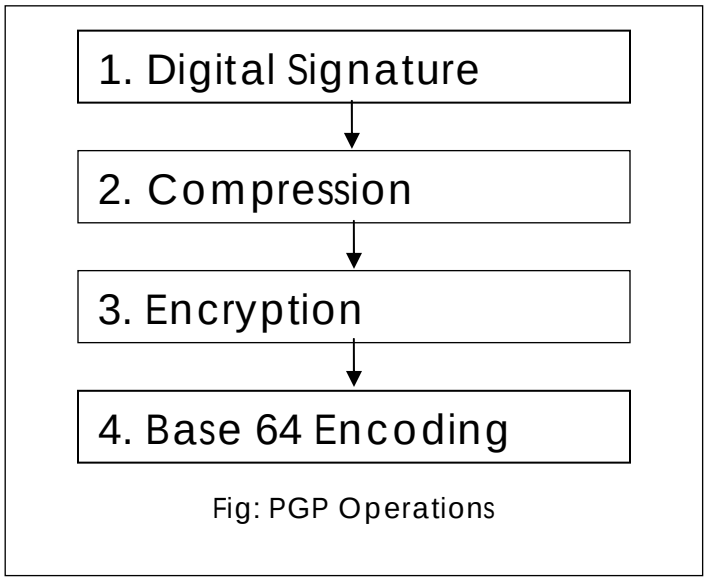


Fig: PGP Operations

Step-I Digital Signature:

Consist of creation of message digest of e-mail message by using SHA-I algorithm resulting message digest is then encrypted with senders private key and result is senders digital signature

Step-II Compression:

Input message and digital signature are compressed together to reduce the size of the final message that will be transmitted

For compression famous ZIP program is used. ZIP is based on the Lempel Ziv algorithm

Lempel ziv algorithm looks for repeated strings or words and stores them invariables and then replaces then occurrence of word by variables

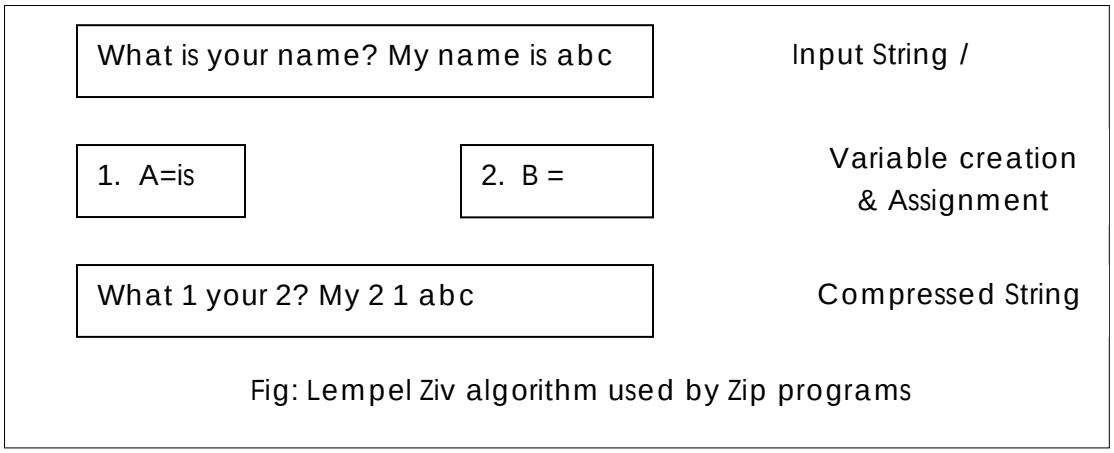


Fig: Lempel Ziv algorithm used by Zip programs

Step-III Encryption:

In this step compressed output of step-II is encrypted by symmetric key for this IDA algorithm in CFB mode is performed

Step-IV Digital Enveloping:

- Here symmetric key of step-III is encrypted with receiver's public key.
- The output of step-II and step-III together form digital envelope

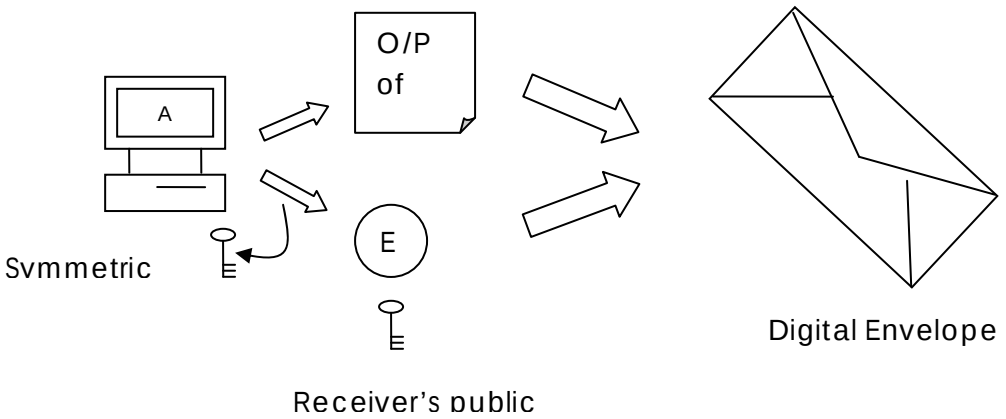


Fig. Formation of digital envelope

Step-V Base-64 Encoding:

- As shown in following block diagram output of step –IV i.e. digital envelope is treated as input stream
- Input stream is divided into 24 bit block
- Each 24-bit block is divided into four 6-bit block
- Each 6-bit block is further mapped to 8-bit block to produce the final **result of BASE-64 Encoding**

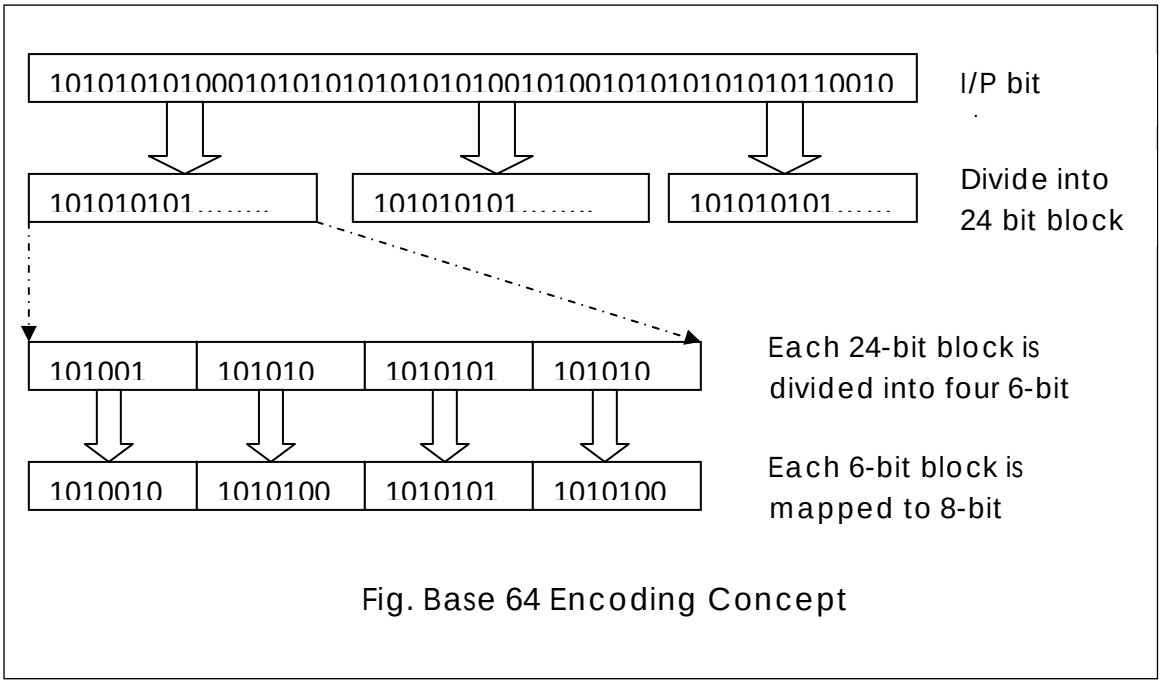


Fig. Base 64 Encoding Concept

Secure multipurpose Internet mail extension(S/MIME)

- Traditional mail was text based now users want to transfer text along with data file in various binary formats
- To cater the need of user MIME system extends the basic email system
- A mime email system contain normal message along with some special header and formatted sections of text
- Each section can hold ASCII encoded portion of data

- Each section starts with explanation that how the data follows should be interpreted or decoded at recipients end
- Recipient e-mail system uses the explanation to decode the data

From : dpmishra < dpmishra@bitdurg.org>

To : Ashwini < ashwini@gmail.com >

Subject : Regarding SMIME

MIME Version 1.0

Content type image/gif

<Actual image data in binary form>

As shown in above s-mime message format the content type is image or gif so based on it recipient mail system will recognize that this is .gif file and it invokes appropriate program that can read interpret and display the content of .gif file

MIME Headers:

Email system provides headers like from, to , date, subject etc where as MIME specification adds 5-new headers to the e-mail system which describes the information about the body of message

1. **MIME Version:** must have value of 1.0 this field indicates that message confirms to RFC 2045 and 2046
2. **Content type:** Describes data contained in the body of message so that receiver e-mail system can deal with received e-mail message
3. **Content transfer encoding:** Specifies the type of transformation that has been used to represent the body of the message
4. **Content ID:** Identifies MIME entities uniquely with reference to multiple context
5. **Content description:** Used when body is not readable

S-MIMIE functionality: Too much similar to PGP

Note: when we enhance basic MIME system to provide security features, it is called as secure multipurpose Internet mail extension

X.400:

Is messaging (notably e-mail) standard specified by ITU-TS (International telecommunication union –Telecommunication standard)

- It's an alternative to SMTP protocol
- X.400 is common in Europe and Canada
- Its actually a set of standard , each in the range of number 400
- X.400 is an official standard where as SMTP is defacto standard
- As x.400 is official standard products with it are more rigorously tested than the products with SMTP implementations
- X.400 offers more capabilities than SMTP

IP and Web Security Protocols:

IPSEC (IP Security):

- IP packet contains data in plain text format
- The data of packet can be watched by anyone through whom the packets are passing
- We have seen some higher level securities like PGP,PEM, S-MIME to prevent problem related plaintext data of packet
- However there was general feeling from long time that why not to secure the IP packet itself rather than relying on higher layer protocols
- If we are able to achieve the IPSEC then there is no need to rely on higher level protocol
- Thus we have two levels of security mechanism that can serve as additional security mechanism or scheme
- First offers security at IP packet level itself
- Continue implementing higher level security mechanism depending on the requirement

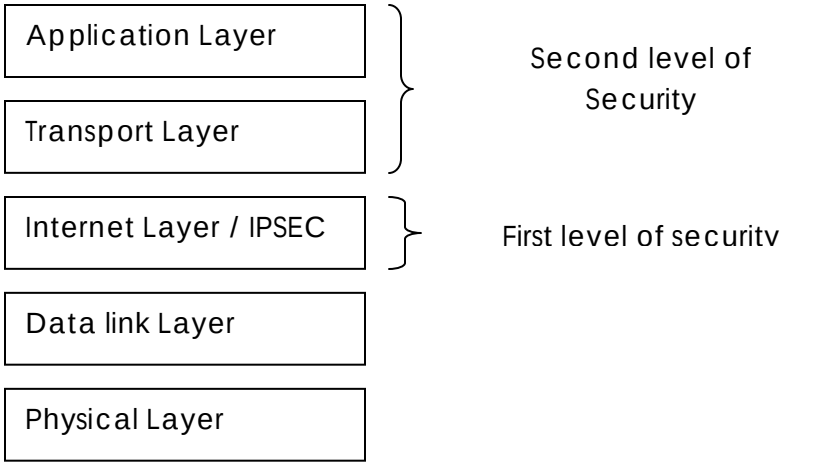


Fig. Security at Internet layer as ell as on above layers too

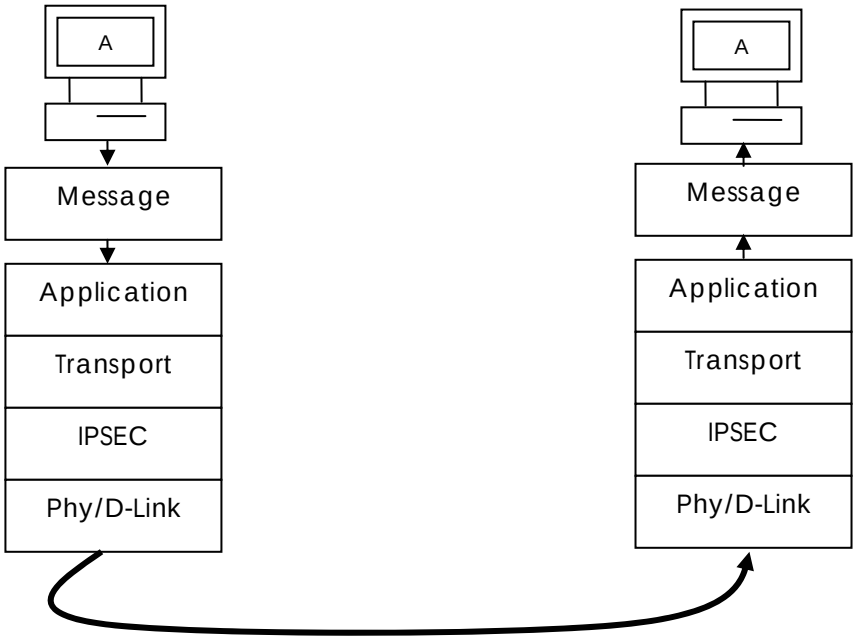


Fig: Conceptual IPSEC positioning in TCP/IP Protocol

- IP Packet consist of two portion i.e. IP header and Actual data
- IPSEC features are implemented in the form of Additional IP headers (called extension headers)

- IPSEC offers two main services

  1. Authentication
  2. Confidentiality

  Each of above services will require its own header

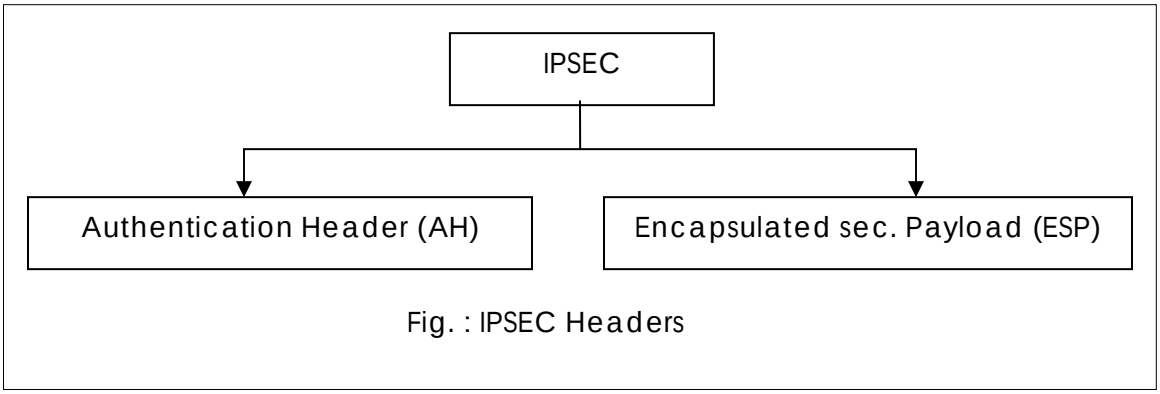
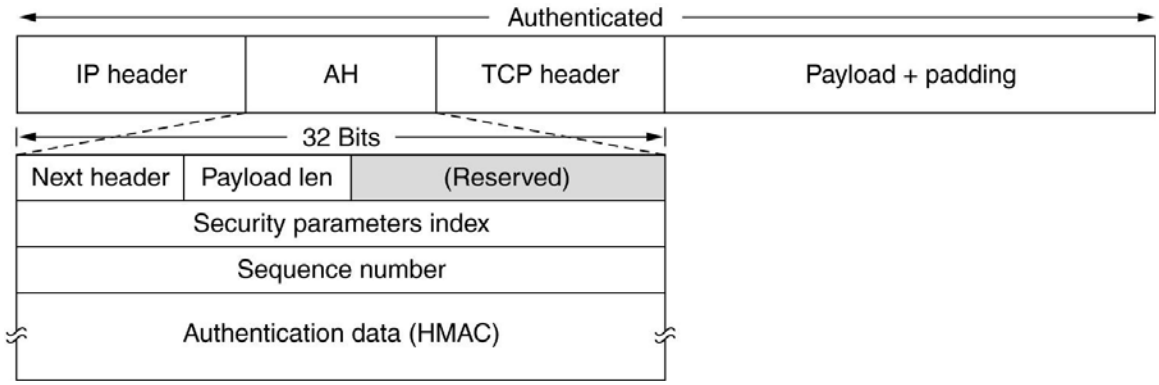


Fig. : IPSEC Headers

Authentication header (AH):

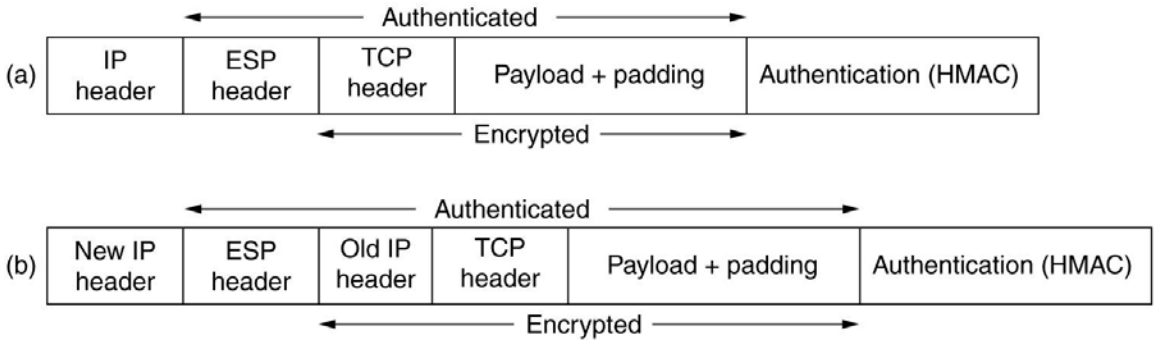


- Provides authentication integrity and an optional anti replay service
- IPSEC AH is header in IP Packet which contains a cryptographic checksum (Similar to message digest or hash)
- AH is simply inserted between IP header and any subsequent packet content no changes are required to the data contents of the packet. Thus security resides completely in the content of AH

i

Encapsulated Security Payload (ESP):

- This protocol provides data confidentiality
- ESP also defines new header to be inserted into IP packet
- ESP processing also includes transformation of the processed data into an unreadable encrypted format
- On recipient end AH is processed and checked by IPSEC if its correct then decryption of payload is carried out



Both AH and ESP can be used in one of two modes

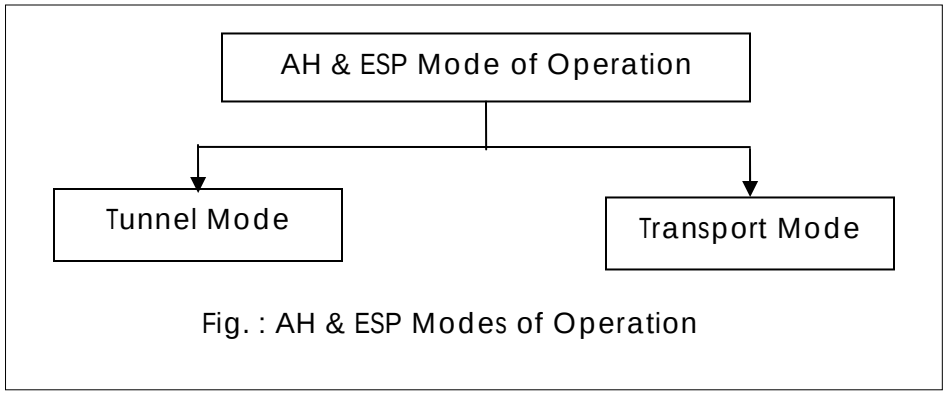
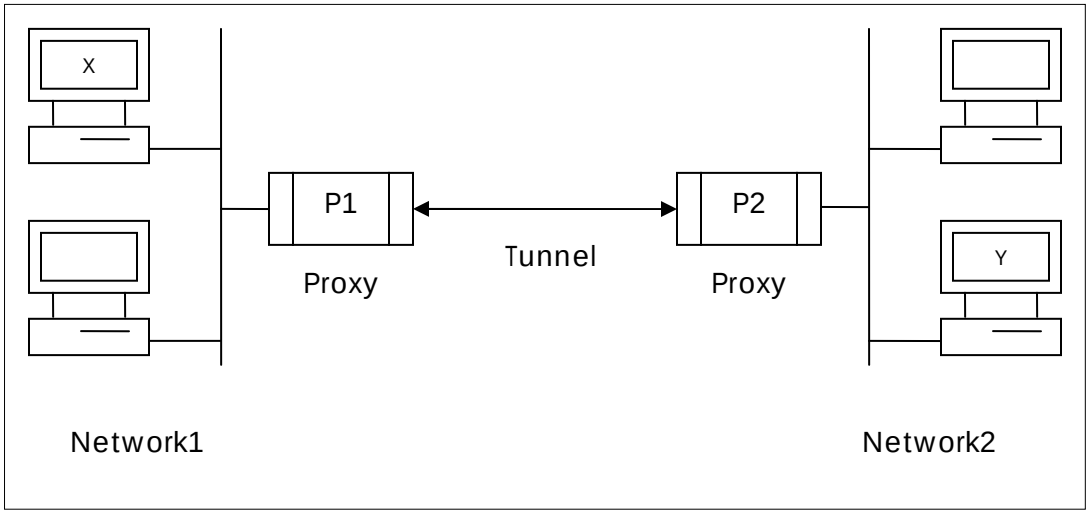


Fig. : AH & ESP Modes of Operation

Tunnel Mode:

In tunnel mode an encrypted tunnel is established between two hosts as shown below



- As shown in above block diagram X and Y are the two hosts wants to communicate with each other using IPSEC tunnel
- Both X and Y would identify their respective proxy servers say P1 and P2
- Logical encrypted tunnel is established between P1 and P2
- X sends information to P1 then tunnel carries information from P1 to P2 and P2 forwards it to Y

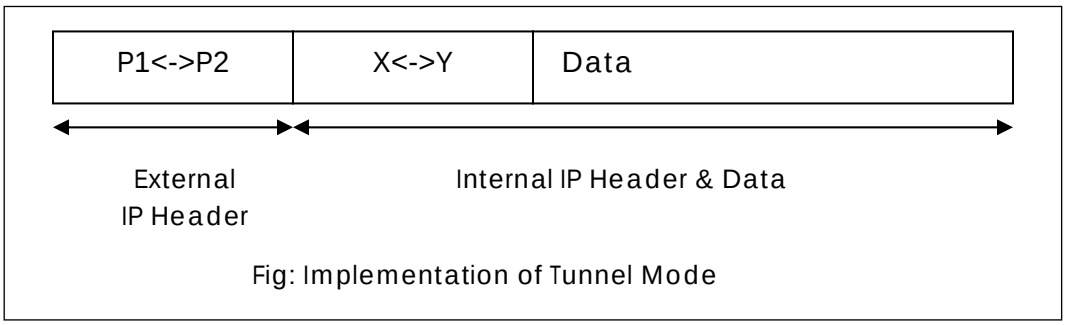


Fig: Implementation of Tunnel Mode

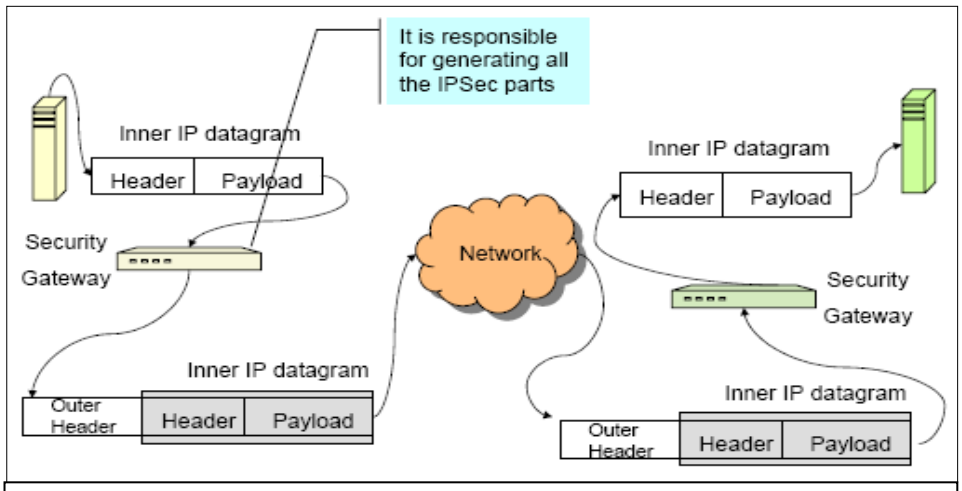


Fig: Shows IPSEC in Tunnel Mode

Transport Mode:

Doesn't hide the actual source and destination address the are visible in plain text while transfer as shown in following block diagram

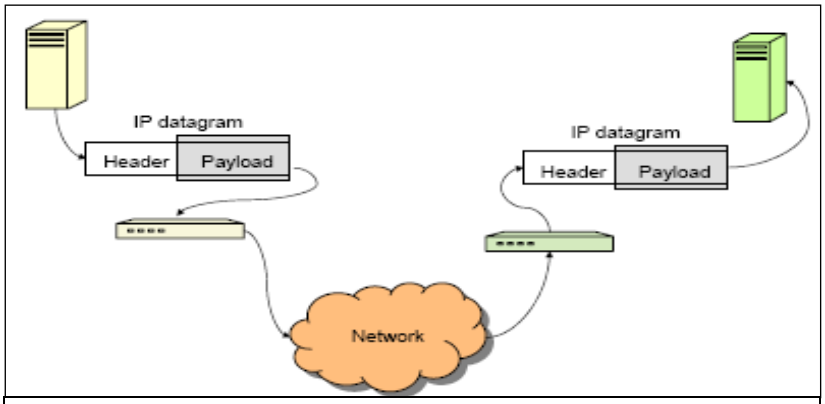


Fig: Shows IPSEC in Transport Mode

- Protection covers IP datagram payload (and selected header fields).
- Could be TCP packet, UDP, ICMP message
- Host-to-host (end-to-end) security:
- IPSec processing performed at the endpoints of the secure channel.
- So the endpoint hosts must be IPSec-aware, i.e. they must be able to do all the authentications and integrity checks plus all the deciphering.

The Internet Key Exchange (IKE) protocol:

- IKE is supporting protocol used in IPSEC this protocol is used in user key management
- IKE is used to negotiate the cryptographic algorithm to be later used by AH and ESP in actual cryptographic operation
- Output of IKE is SA(Security association)

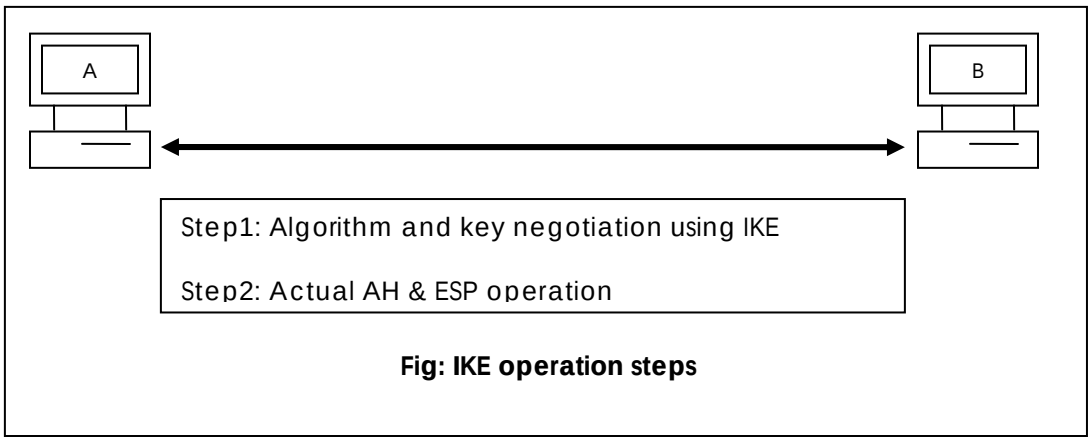


Fig: IKE operation steps

Security Association (SA):

SA is agreement between communicating parties about factors such as

1. IPSEC protocol version in use
2. Mode of operation (Transport or Tunnel mode)
3. Cryptographic algorithm
4. Cryptographic keys and lifetime of keys etc

Once SA is established both major protocols IPSEC (i.e. AH and ESP) make use of it for actual operation

Note: If both AH and ESP are used in that case communicating parties require two set of SA one for AH and other for ESP

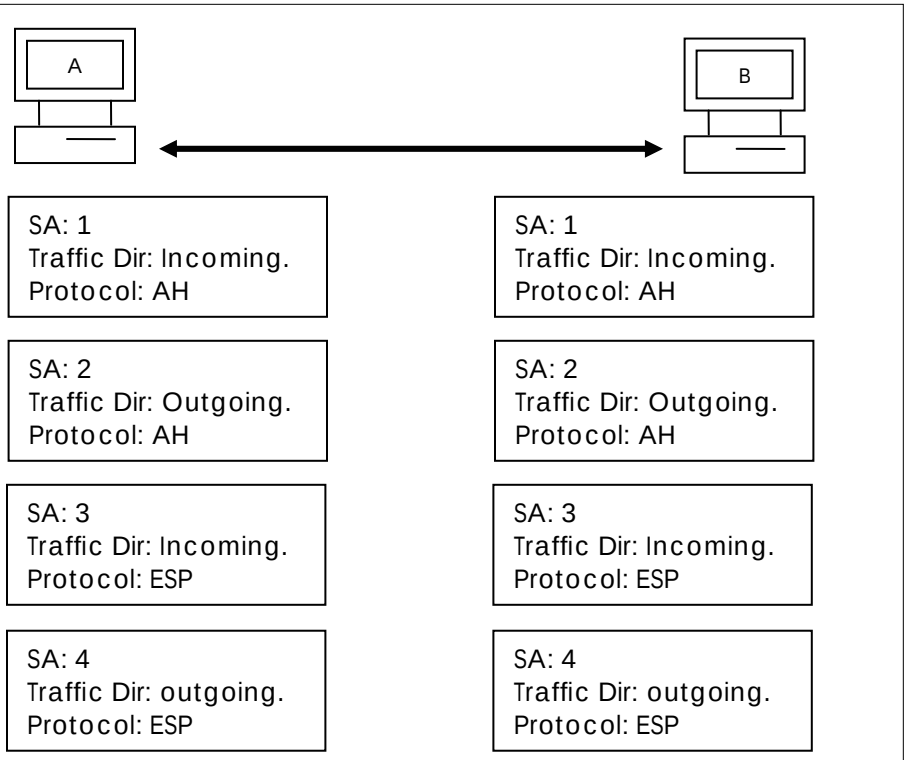


Fig: Security association when AH and ESP used

- Both communicating parties must allocate some storage area for storing the SA information at their end
- For storage purpose a standard storage area called as security association database (SAD) is predefined and used by IPSEC
- So each communicating part requires to maintain its own SAD that contains

1. Sequence number counter
2. Sequence counter overflow
3. Anti replay window
4. AH authentication
5. ESP authentication
6. ESP encryption
7. IPSEC protocol mode

Virtual Private Network (VPN):

There is clear demarcation between private and public network

Public Network: Public telephone system and the Internet

Private Network: Made up of computers owned by a single organization with each other

Consider a corporate office wants to connect two of its branches

Branches are situates at far distances i.e. one in Bangalore and other in Bhilai for achieving this there are two solutions

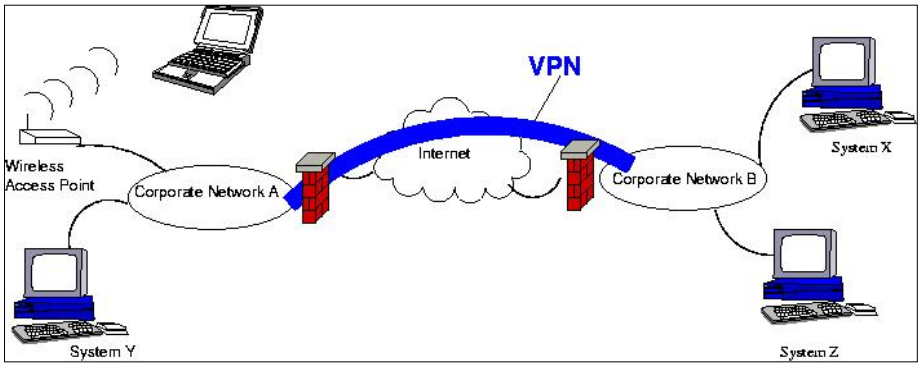
1. Connect two branches using personal network i.e. lay cables or establish radio link between two branches
2. Connect two branches with the help of public network such as N/W of N/W or Internet

Laying cable is costliest solution and not feasible solution so we have to opt second option to use public network for joining two branches

What is VPN?

There are two ways to connect remote sites:
- Use a dedicated line (a private network).
- Use the Internet.
    - Not private, so need to secure the connection.
    - Want to keep internal network hidden from Internet.
    - Want to allow two sites to access LAN at each site as if part of same network.
    - The secure access using the Internet instead of a dedicated line is what makes it a Virtual, Private Network.

Why VPN?



- Connect two sites securely through public network
- Allow remote access by individual users.
- Allows travelling users to remotely access private network
- If we remove VPN link then two sites will be separated with each other
- By employing VPN two remote sites seems to be the one/single virtual site

VPN architecture:

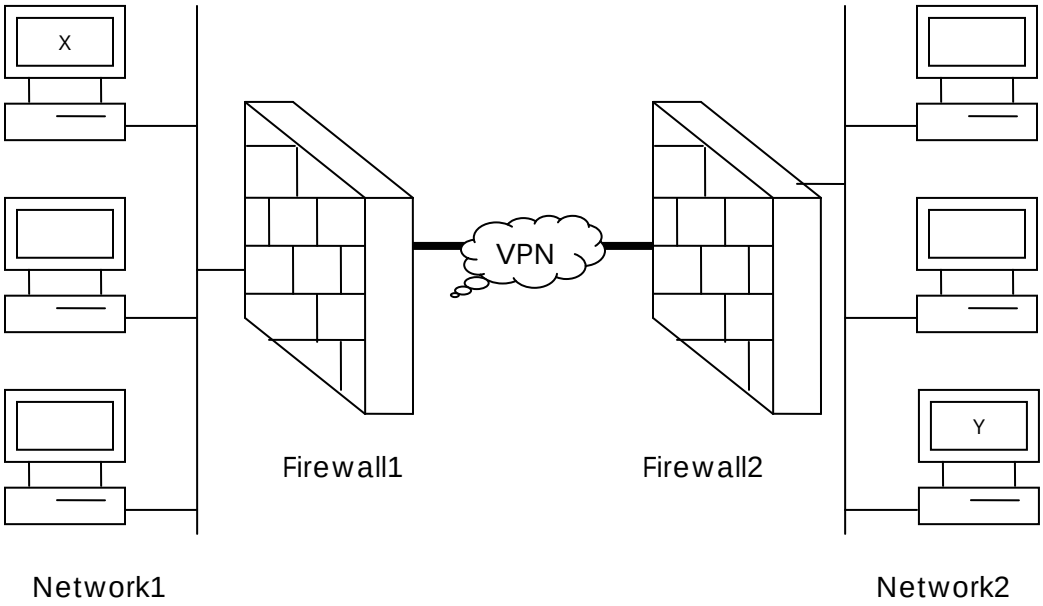


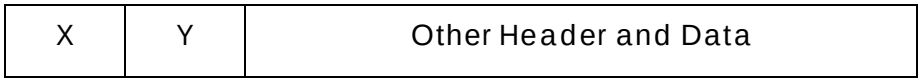
Fig: Architecture of VPN

As shown in above block diagram two networks (two branch offices) are connected with each other through the firewall with best possible configuration for setting firewall was selected by organization

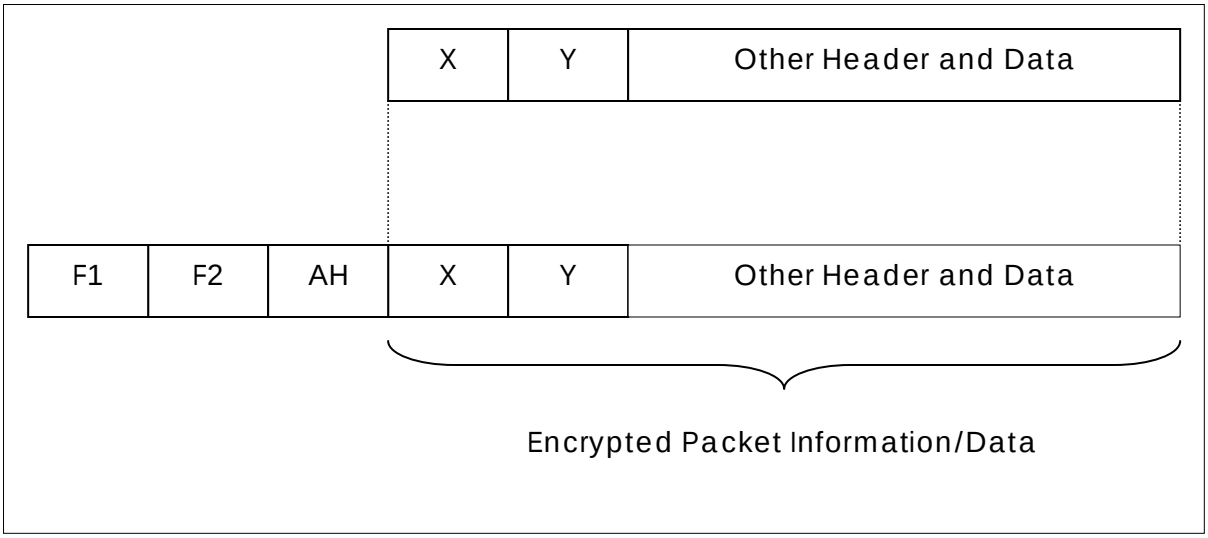
Key point here is two firewalls are connected with each other through Internet as shown through VPN tunnel

Let's consider host X on network-1 wants to transfer data packet to host y on network-2 for this following steps are used

1. Host X creates the packet with header information X <----> Y and gives it to firewall1



2. Firewall1 adds new headers to the packet as well encrypt the original packet data



3. Now Firewall1 diverts packet to Firewall2

4. Firewall2 discards the outer header make check of AH and decrypts the header information and payload this results actual packet created in step-1

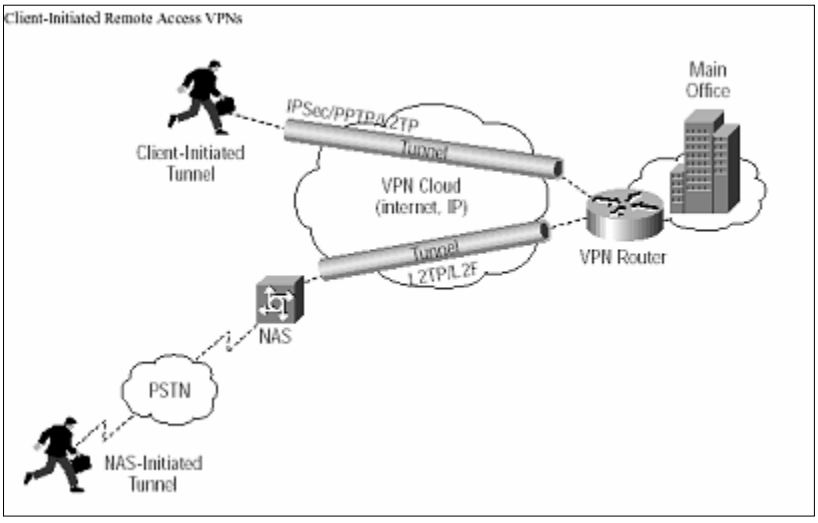
F1	F2	AH	X	Y	Other Header and Data
			X	Y	Other Header and Data

Fig. Firewall-II retrieves the original packet content

Types of VPN:

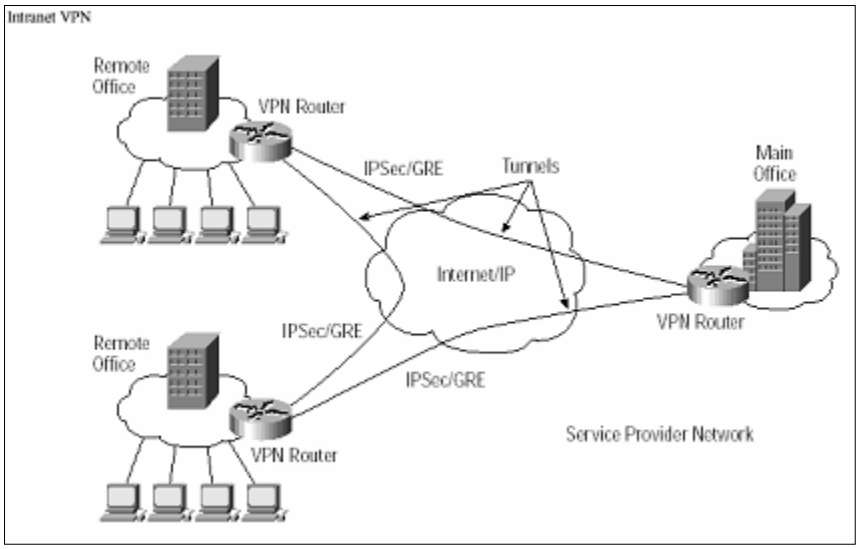
1. Remote access VPN
2. Intranet VPN
3. Extranet VPN

Remote Access VPN:



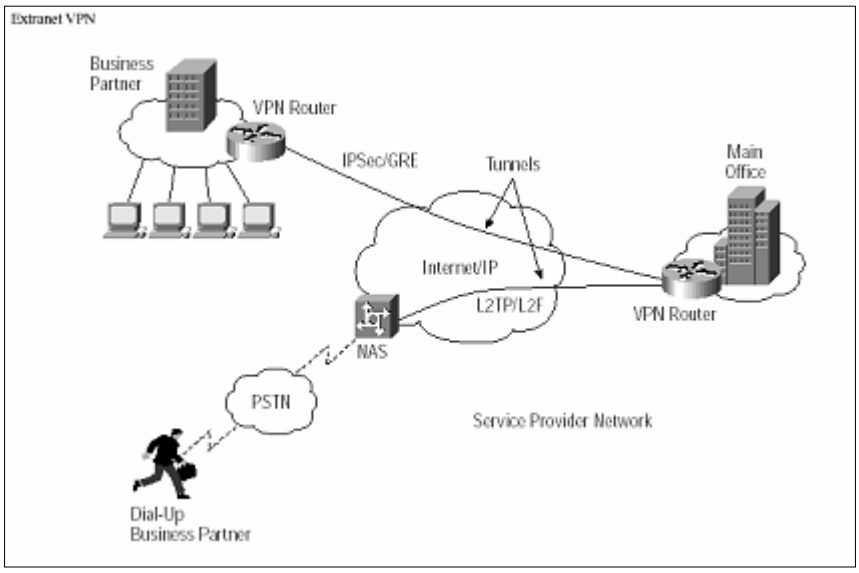
Gives access to remote or roaming users access of Main office / branch office as shown in above block diagram

Intranet VPN



As shown in above block diagram Intranet VPN is used for joining different branches of organization. Important thing here is all the branches are connected through common service provider

Extranet VPN



As shown main branch and branch offices are joined by different service providers through public network

Advantages of VPN

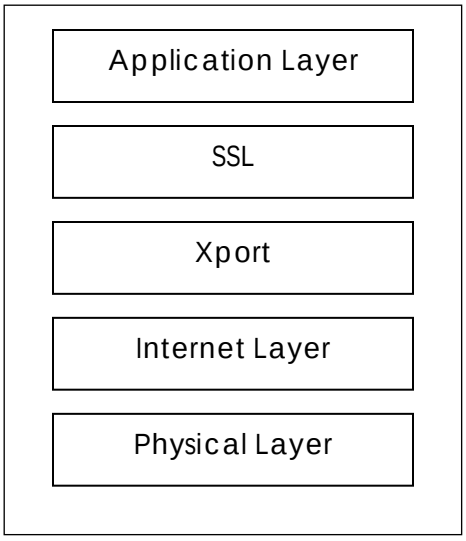
1. Greater scalability
2. Easy to add or remove users
3. Reduce long distance Telecommunication cost
4. Mobility
5. Scalability

Drawbacks:

1. Lack of standards
2. Understanding of security issues
3. Unpredictable Intranet traffic
4. Difficult to accommodate product from different vendors

Secure Socket Layer (SSL):

- SSL is an Internet protocol used for secure exchange of Information between a web browser and web server
- Provides two basic services
  - Authentication
  - Confidentiality
- Logically it provides secure pipe between the web browser and web server
- SSL is developed by Netscape corporation in 1994 since then SSL becomes the world most popular web security mechanism
- SSL is supported by all web browsers available in the market
- SSL comes in three version 2 , 3 and 3.1



As shown in fig SSL can be conceptually considered as an additional layer in TCP/IP protocol suite

SSL layer is located in between transport layer and application layer as shown

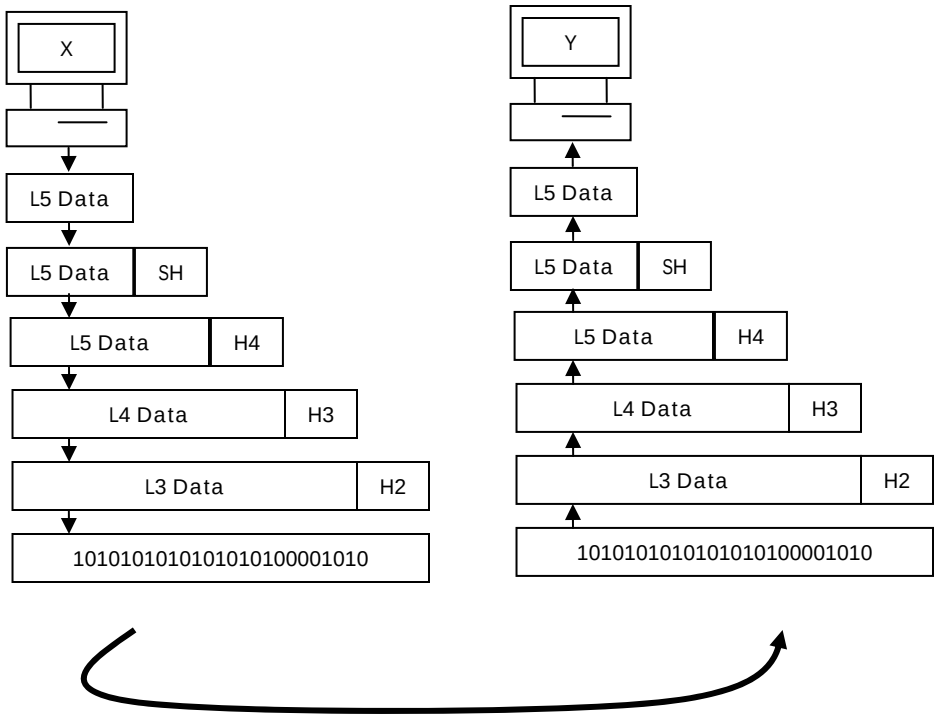


Fig: Conceptual SSL positioning in TCP/IP Protocol

- Application layer of sender computer X-prepares data to be send to receiving computer Y
- As usual what happen in normal case application layers data is passed to transport layer directly but here in this case data is passed to SSL layer
- SSL layer encrypts data received from application layer and adds its own header information
- From SSL layer data is passed to transport layer and it adds its own header H4 and so on rest of the process is similar to normal TCP/IP protocol in which each and every layer is adding its own header to data received from the upper layer(i.e. process of encapsulating data)
- At receiving end exactly reverse process is carried every layer verifies data as per their own functionality if its found correct then it discards corresponding header of that concerned layer and popup the data for the upper layer (i.e. decapsulation process carried)

SSL Working:

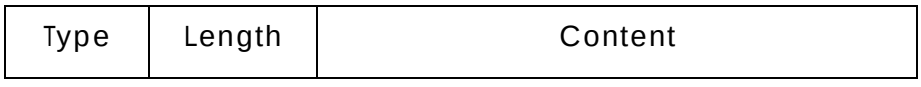
SSL has three sub protocols namely

1. Handshake protocol
2. Record protocol
3. Alert protocol

Above three sub protocols constitute the overall working of SSL

1. Handshake protocol

- First sub protocol of SSL used by client and server to communicate using and SSL enabled connections
- Handshake protocol has series of messages between client and server and format of message is



1-byte 3-byte 1 or more byte

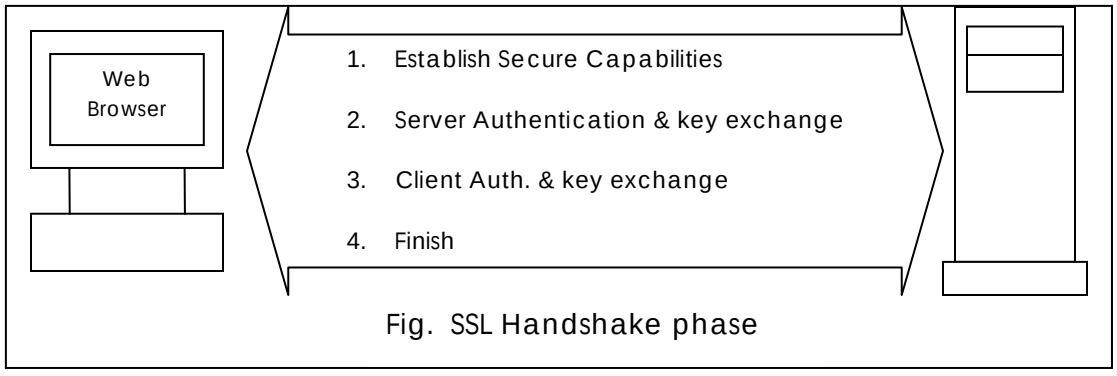
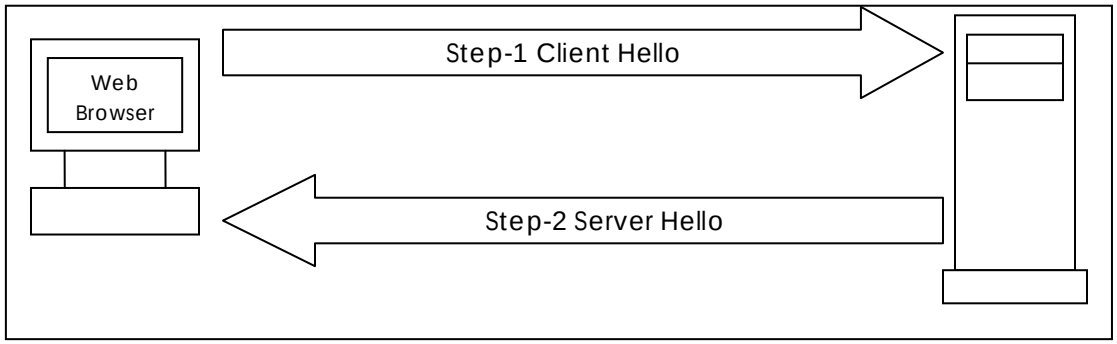


Fig. SSL Handshake phase

Phase-I: Establish Security capabilities: This phase of SSL handshake is used to initiate a logical connection as shown below



As shown in above block diagram web browser and web server establishes secure capabilities by exchanging version of SSL, Random session ID , cipher suite and compression method

Phase-II: Server authentication and key exchange:

- Server initiates the second phase of SSL handshake and is the sole sender of all the messages in this phase
- The client is sole recipient of all these messages this phase contain four steps as shown below

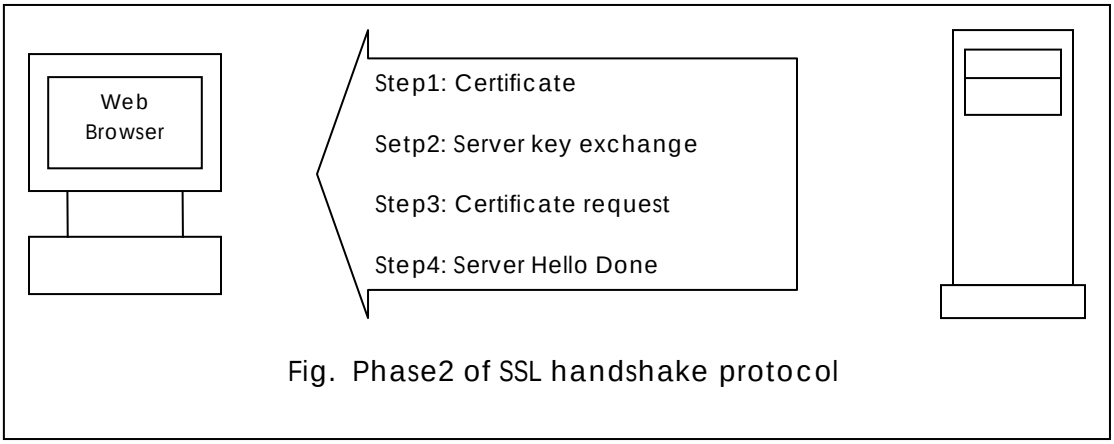


Fig. Phase2 of SSL handshake protocol

Phase-III: Client authentication and key exchange:

- Client initiate 3rd phase of SSL and is whole and sole sender of all messages in this phase
- Server is sole recipient of all the messages this phase contains 3 – steps as shown below

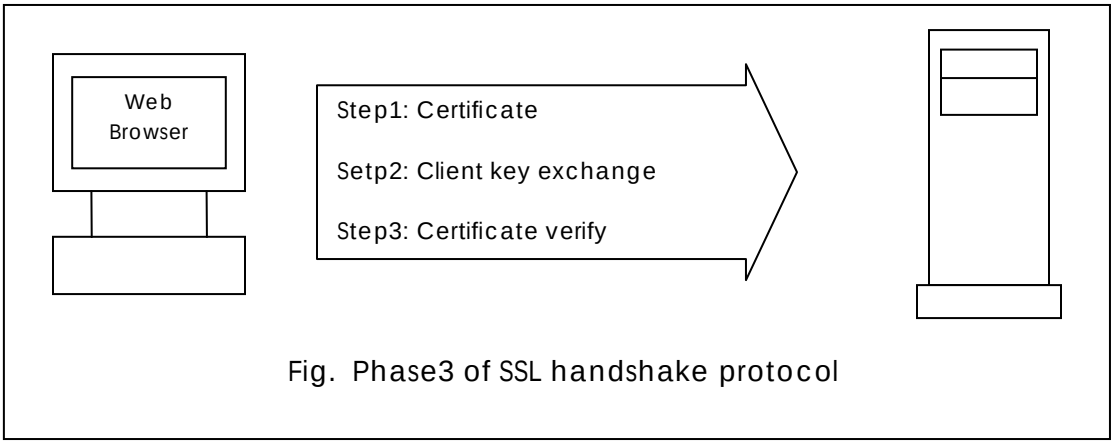
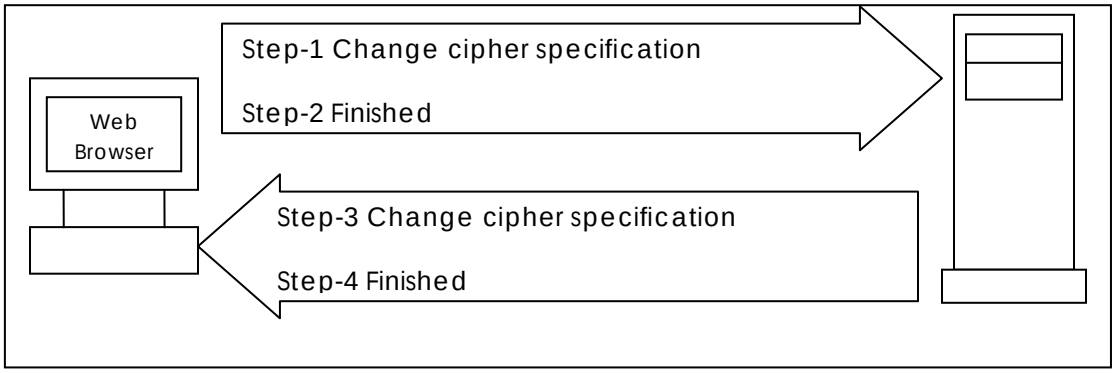


Fig. Phase3 of SSL handshake protocol

- First step is optional and performed only if there is request from server
- Second step related with client key exchange and key is for symmetric algorithm
- Here client creates 48 bit premaster secret and encrypt it with server key and send this encrypted premaster to server
- Third step(certificate verify) is only necessary if server has demanded clients authentication as we know client already send his certificate now its time for client to prove the server that he is correct and authorized holder of the private key corresponding to certificate

Phase-IV: Finish



Client initiate fourth phase of SSL first two messages are there from client i.e. change cipher specification and finished similarly server responds with two identical messages change cipher specification and finish

2. Record protocol

Record protocol in SSL comes into picture after completion of successful handshake between client and server

This protocol provides two services to an SSL connection as follows

1. Confidentiality: Achieved by using the secret key that is defined by handshake protocol

2. Integrity: Handshake protocol also defines a shared secret key (MAC) that is used for assuring the message integrity

3. Alert protocol:

- When either client or server detects an error the detecting party sends an alert message to the other part
- If error is fatal both parties immediately close connection and destroy session identifier and secret key associated with this connection
- Non secure errors do not result in the termination of connection , instead the parties handle the error and continue session

Transport layer security (TLS):

- SSL is also called as TLS after version 3.0
- Transport layer security service
- Originally developed by Netscape
- Version-3 developed with public I/P
- Subsequently become Internet standard known as TLS (Transport layer security)
- Uses TCP to provide a reliable end – to – end service
- SSL has two layers of protocol

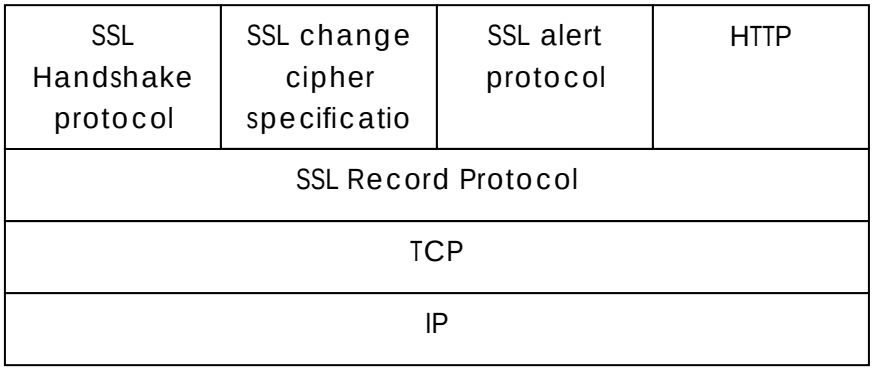


Fig: Architecture of SSL