UNIT II

2 Key Management and Authentication

Syllabus

Key Management and Distribution: Symmetric Key Distribution, Distribution of Public Keys, X.509 Certificates, Public-Key Infrastructure. User Authentication: Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos Systems, Remote User Authentication Using Asymmetric Encryption.

Contents

2.1 Key Management and Distribution

- The purpose of public key cryptography is,
  1. The distribution of public keys.
  2. The use of public key encryption to distribute secret keys.

2.1.1 Distribution of Public Keys

- Different methods have been proposed for the distribution of public keys. These are
  1. Public announcement.
  2. Publicly available directory.
  3. Public key authority.
  4. Public key certificates.

1. Public announcement

- In public key algorithm, any participant can **send** his or her public key to any other participant or **broadcast** the key to the community at large.
- Fig. 2.1.1 shows the public key distribution.

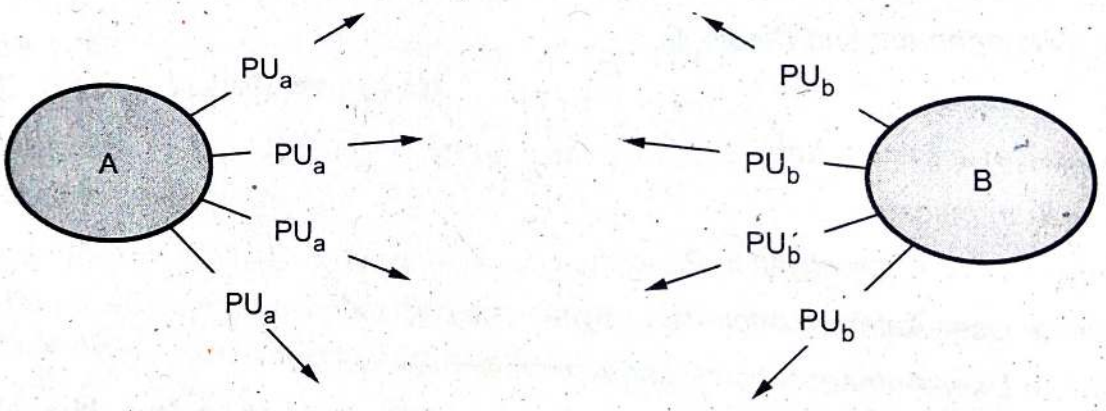

Fig. 2.1.1 Public key distribution

- Because of the growing popularity of PGP, which makes use of RSA, many PGP users have adopted the practice of appending their public key to messages that they send to public forums, such as USENET newgroups and Internet mailing lists.
- The disadvantage is that, anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key.

2. Public available directory

- Greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization.

- Fig. 2.1.2 shows public key publication.

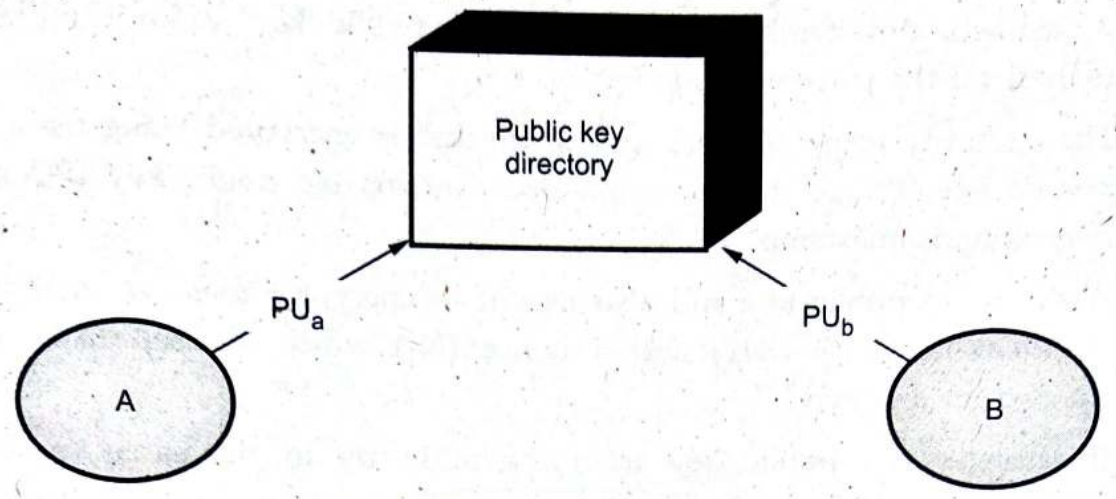

Fig. 2.1.2 Public key publication

- Such a scheme would include the following elements :
  1. The authority maintains a directory with a {name, public key} entry for each participant.
  2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
  3. A participant may replace the existing key with a new one at any time.
  4. Participants could also access the directory electronically.

3. Public key authority

- Fig. 2.1.3 shows public key distribution scenario.

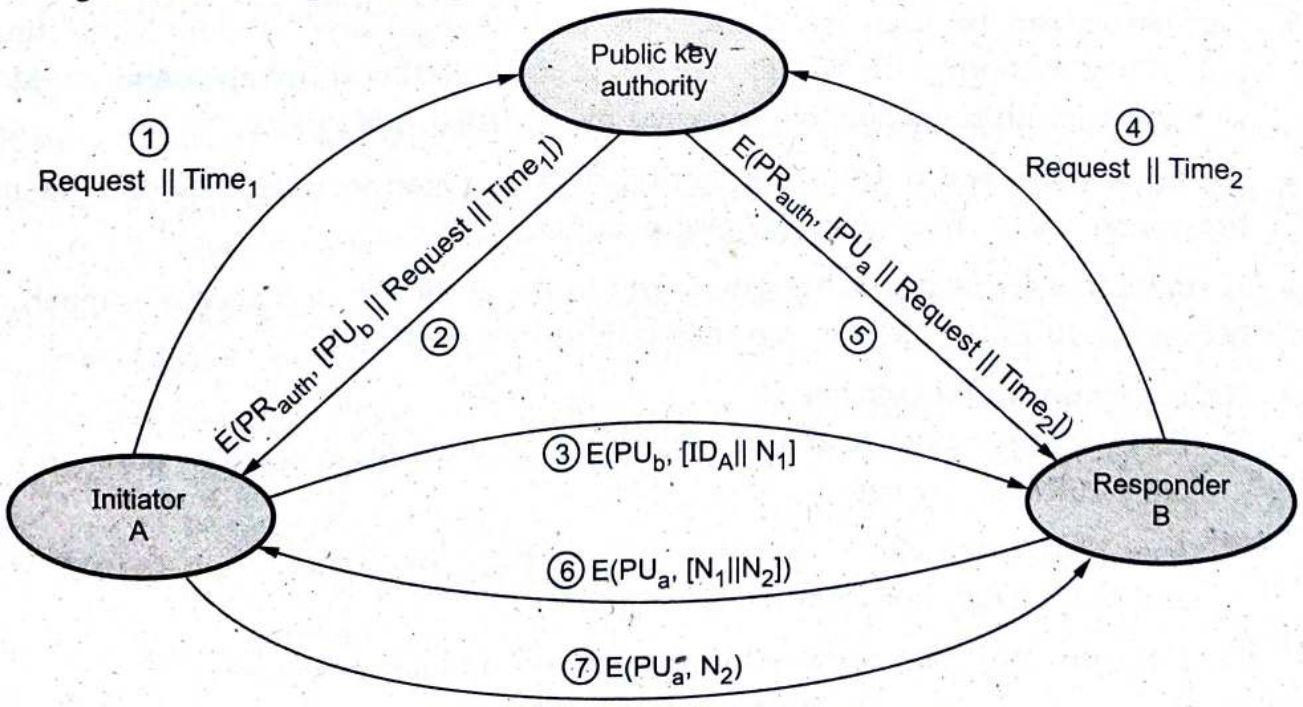

Fig. 2.1.3 Public key distribution scenario

- Following steps occur in public key distribution :
  1. A sends a timestamped message to the public key authority containing a request for the current public key of B.
  2. The authority responds with a message that is encrypted using the authority's private key, PR_{auth}. The message also contains B's public key (PU_b), original request and timestamp.
  3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1) which is used to identify this transaction uniquely.
  4. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.
  5. Public keys have been securely delivered to A and B and they may begin their protected exchange.
  6. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B(N_2).
  7. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

Drawback

Public key authority could be somewhat of a bottleneck in the system. The directory of name and public keys maintained by the authority is vulnerable to tampering.

4. Public key certificates

- Certificates can be used by participants to exchange keys without contacting a public key authority. Certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.
- The third party is a certificate authority, such as government agency or a financial institution, that is trusted by the user community.
- A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate.
- Requirements on this scheme :
  1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
  2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
  3. Only the certificate authority can create and update certificates.
  4. Any participant can verify the currency of the certificate.

- A certificate scheme is illustrated in Fig. 2.1.4. Each participant applies to the certificate authority, supplying a public key and requesting a certificate.

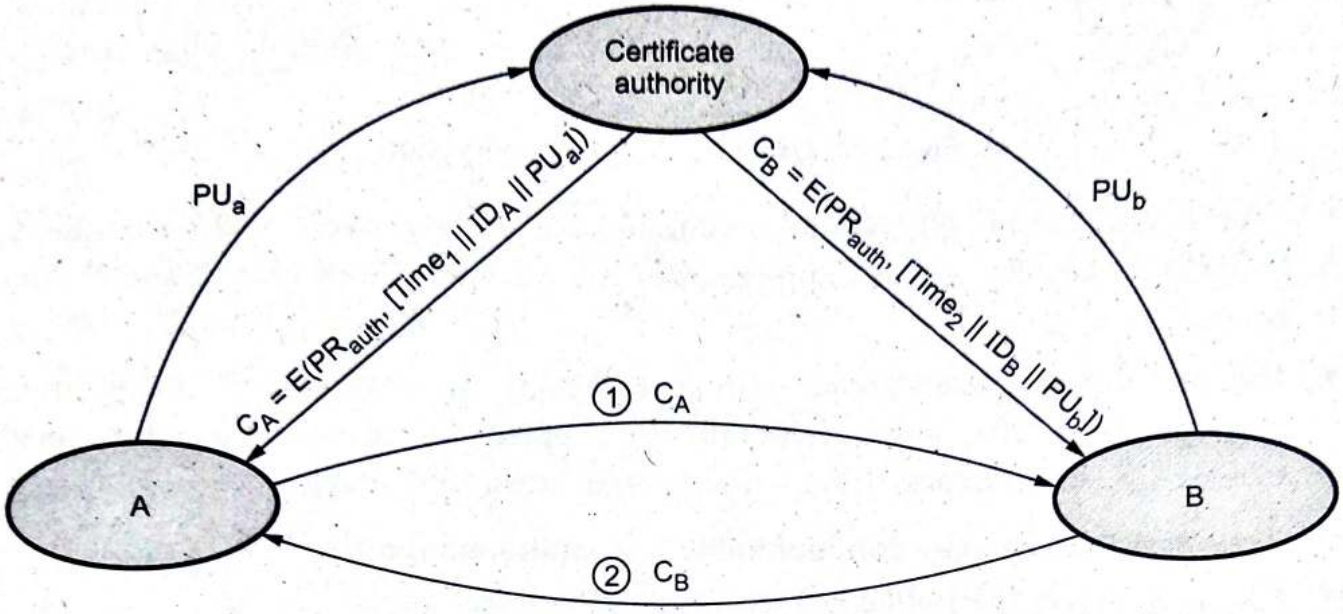

Fig. 2.1.4 Exchange of public key certificates

- For participant A, the authority provides a certificate of the form

$$C_A = E(PR_{auth}, [T \,||\, ID_A \,||\, PU_a])$$

where PR_{auth} is the private key used by the authority and T is a timestamp.

2.1.2 Distribution of Secret Keys using Public Key Cryptography

- Public key encryption provides for the distribution of secret key to be used for conventional encryption.

Simple secret key distribution

If user A wishes to communicate with user B, the following procedure is employed :

1. User A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message to user B consisting of PU_a and an identifier of A, ID_A.
2. User B generates a secret key (K_s) and transmits it to user A, encrypted with A's public key.
3. User A computes $D(PR_a, E(PU_a, K_s))$ to recover the secret key. Because only A can decrypt the message, only user A and user B know the identity of K_s.
4. User A discards PU_a and PR_a and user B discards PU_a.
5. Fig. 2.1.5 shows use of public key encryption.

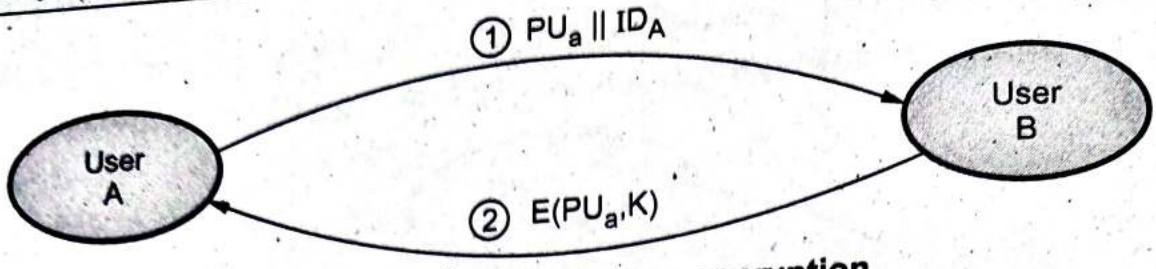

Fig. 2.1.5 Use of public key encryption

- User A and B can now securely communicate using conventional encryption and the session key K_s. At the completion of the exchange, both user A and B discard K_s.
- The protocol discussed above is insecure against an adversary who can intercept messages and then either relay the intercepted message or substitute another message. Such an attack is known as a **man in middle attack.**

Secret key distribution with confidentiality and authentication

- Fig. 2.1.6 shows the public key distribution of secret keys.

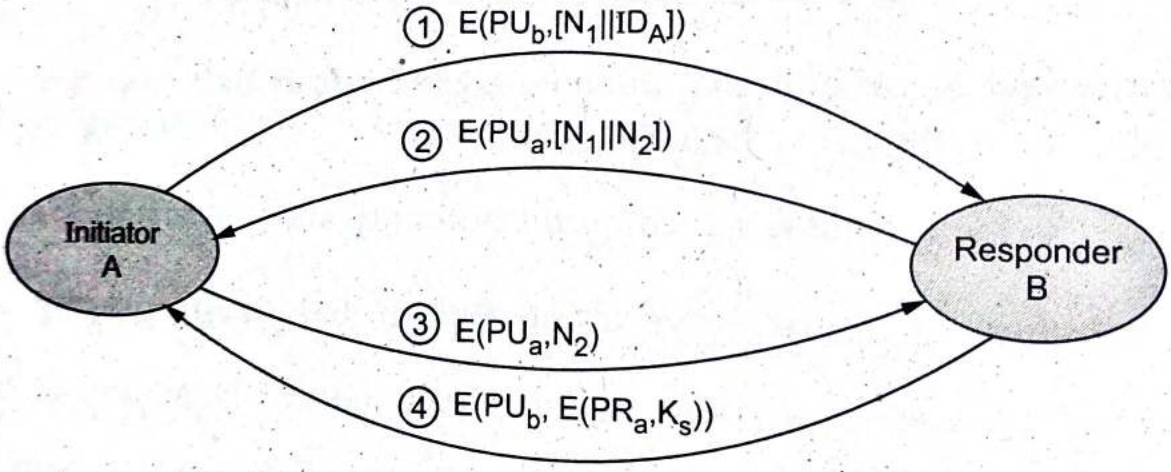

Fig. 2.1.6 Public key distribution of secret keys

- It provides protection against both passive and active attacks.
  1. A uses B's public key to encrypt a message to B containing an identifier of A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.
  2. B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as well as a new nonce generated by B(N_2).
  3. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.
  4. A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B.
  5. B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.

2.1.3 Key Distribution and Certification

- Management and handling of the pieces of secret information is generally referred to as **key management.**
- Activities of key management includes selection, exchange, storage, certification, revocation, changing, expiration and transmission of the key.
- Key management is the set of processes and mechanisms which support key establishment and maintenance of ongoing keying relationship between parties, including replacing older key with new keys.
- Two major issues in key management are :
  1. Key life time 2. Key exposure

Key life time - limit of use which can be measured as a duration of time.

Issue related to key :

1. Users must be able to obtain securely a key pair suited to their efficiency and security needs.
2. Keys need to be valid only until a specified expiration date.
3. The expiration date must be chosen properly and publicized securely.
4. User must be able to store their private keys securely.
5. Certificates must be unforgettable, obtainable in a secure manner.

1. Public Key Infrastructure

- Public Key Infrastructure (PKI) is a well-known technology that can be used to establish identities, encrypt information and digitally sign documents.
- PKI identifies and manages relationships of parties in an electronic exchange, serving a wide array of security needs including access control, confidentiality, integrity, authentication and non-repudiation.
- PKI also uses unique Digital Certificates (DC) to secure eCommerce, email, data exchange and Virtual Private Networks (VPN) and intranets and is also used to verify the identity and privileges of each user.
- The Certificate Authority (CA) provides a full life-cycle management of public keys and certificates, including issuance, authentication, storage, retrieval, backup, recovery, updating and revocation to the PKI.
- All users of PKI must have a registered identity, which is stored in a digital certificate issued by a CA.
- Remote users and sites using public private keys and public key certificates can authenticate each other with a high degree of confidence.

- Authentication is dependent on three conditions :
  1. It must be established that each party have a private key that has not been stolen or copied from the owner.
  2. The certificate must be issued to the owner in accord with the stated policy of the certificate issuer.
  3. The policies of the certificate issuer must be satisfactory to the parties so as to verify identity.

Benefits of PKI

1. **Confidential communication :** Only intended recipients can read files.
2. **Data integrity :** Guarantees files are unaltered during transmission.
3. **Authentication :** Ensures that parties involved are who they claim to be.
4. **Non-repudiation :** Prevents individuals from denying.

Limitation of PKI

The problems encountered deploying a PKI can be categorized as follows :

1. Public key infrastructure is new
2. Lack of standards
3. Shortage of trained personnel
4. Public key infrastructure is mostly about policies.

2. Certificate

- **Certificates** are digital documents that are used for secure authentication of communicating parties.
- A certificate binds identity information about an entity to the entity's public key for a certain validity period.
- A certificate is digitally signed by a Trusted Third Party (TTP) who has verified that the key pair actually belongs to the entity.
- Certificates can be thought of as analogous to passports that guarantee the identity of their bearers.
- **Authorities :** The trusted party who issues certificates to the identified end entities is called a **Certification Authority (CA).**
- Certification authorities can be thought of as being analogous to governments issuing passports for their citizens.
- A certification authority can be managed by an external certification service provider or the CA can belong to the same organization as the end entities.

- CAs can also issue certificates to other (sub) CAs. This leads to a tree-like **certification hierarchy.**
- The highest trusted CA in the tree is called a root CA.
- In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities.
- For example, the number of certificates required may be too large for a single CA to maintain; different organizational units may have different policy requirements; or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.
- The X.509 standard includes a model for setting up a hierarchy of the certification authority.
- Fig. 2.1.7 shows the hierarchy of certificate authorities.

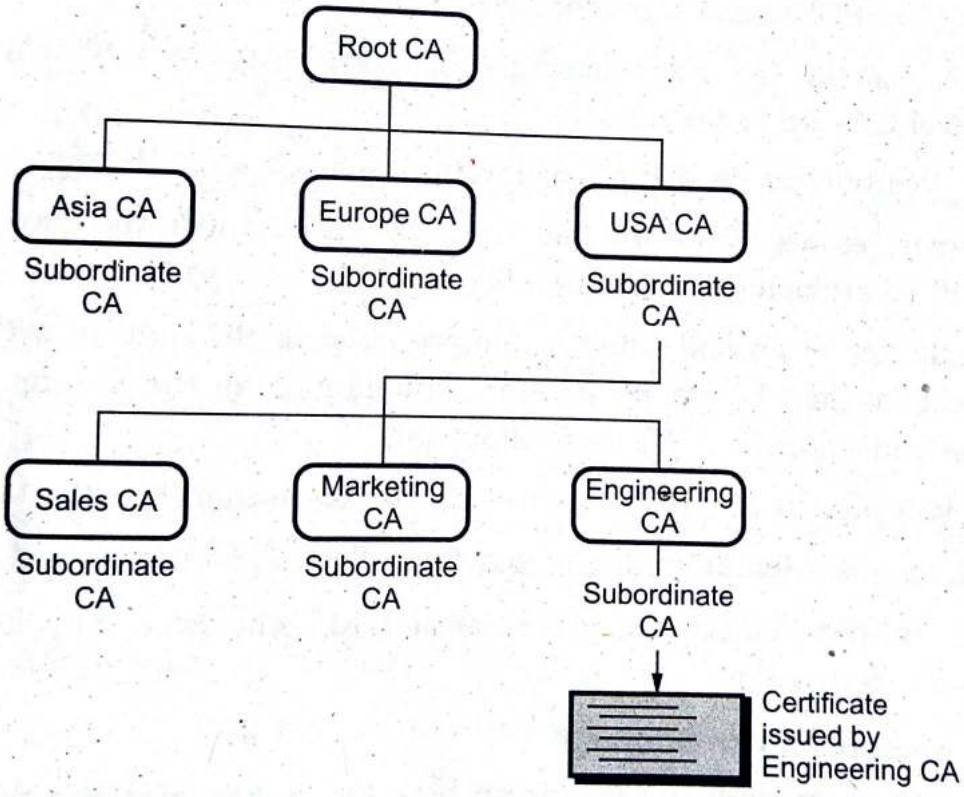

Fig. 2.1.7 Hierarchy of CA

- In the Fig. 2.1.7, the root CA is at the top of the hierarchy. The root CA's certificate is a self-signed certificate : That is, the certificate is digitally signed by the same entity.
- The CAs, that are directly subordinate to the root CA, have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.
- Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies.

- **Certificate chains :** Certificate chain is series of certificates issued by successive CAs.
- In some cases, a CA can delegate the actual identification of end entities as well as some other administrative tasks to a separate entity, the **Registration Authority (RA).**

Verifying certificates

- When authentication is required, the entity presents a signatures it has generated from authentication data using its private key, and a certificate corresponding to that key.
- The receiving entity can verify the signature with the public key of the sender contained in the certificate.
- Next the receiving entity must verify the certificate itself by checking the validity time of the certificate and the signature of the CA in the certificate.
- If the CA is a sub CA, the receiving entity must also verify the signatures of all higher-level CAs up to the root CA.
- The list of certificates needed for verification is called a **certification path.**
- If all signatures are valid and the receiving entity trusts the root CA, the first entity will be authenticated successfully.
- If a private key of an end entity is compromised or the right to authenticate with a certificate is lost before its natural expiration date, the CA must revoke the certificate and inform all PKI users about this.
- The CA will periodically publish a **Certificate Revocation List (CRL).**
- The CRL is a list identifying the revoked certificates and it is signed by the CA.
- The end entities should check the latest CRL whenever they are verifying a validity of a certificate.

3. Key length and encryption strength

- The strength of encryption depends on both the cipher used and the length of the key.
- Encryption strength is often described in terms of the size of the keys used to perform the encryption : In general, longer keys provide stronger encryption.
- Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher.
- Roughly speaking, 128-bit RC4 encryption is 3×10^{26} times stronger than 40-bit RC4 encryption.

- Different ciphers may require different key lengths to achieve the same level of encryption strength.
- The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based.
- Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values.
- Thus a 128-bit key for use with a symmetric key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

2.1.4 Key Distribution

- For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. **Key distribution** refers to the means of delivering a key to two parties who wish to exchange data, without allowing others to see the key.
- For two parties A and B, key distribution can be achieved in a number of ways, as follows.
  1. User A can select a key and physically deliver it to user B.
  2. A third party can select the key and physically deliver it to user A and user B.
  3. If user A and user B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
  4. If user A and user B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to user A and user B.
- For manual delivery of key, **options 1 and 2** are used. These options are suitable for **link encryption.**
- Option 3 is suitable for link encryption or end-to-end encryption.
- For end-to-end encryption, some variation on option 4 has been widely adopted.
- The use of a key distribution center is based on the use of a hierarchy of keys. Minimum two levels of keys are used. Fig. 2.1.8 shows the use of a key hierarchy.
- Communication between end systems is encrypted using a temporary key, often referred to as a **session key**. The **session key** is used for the duration of a logical connection, such as a frame relay connection, or transport connection and then discarded.

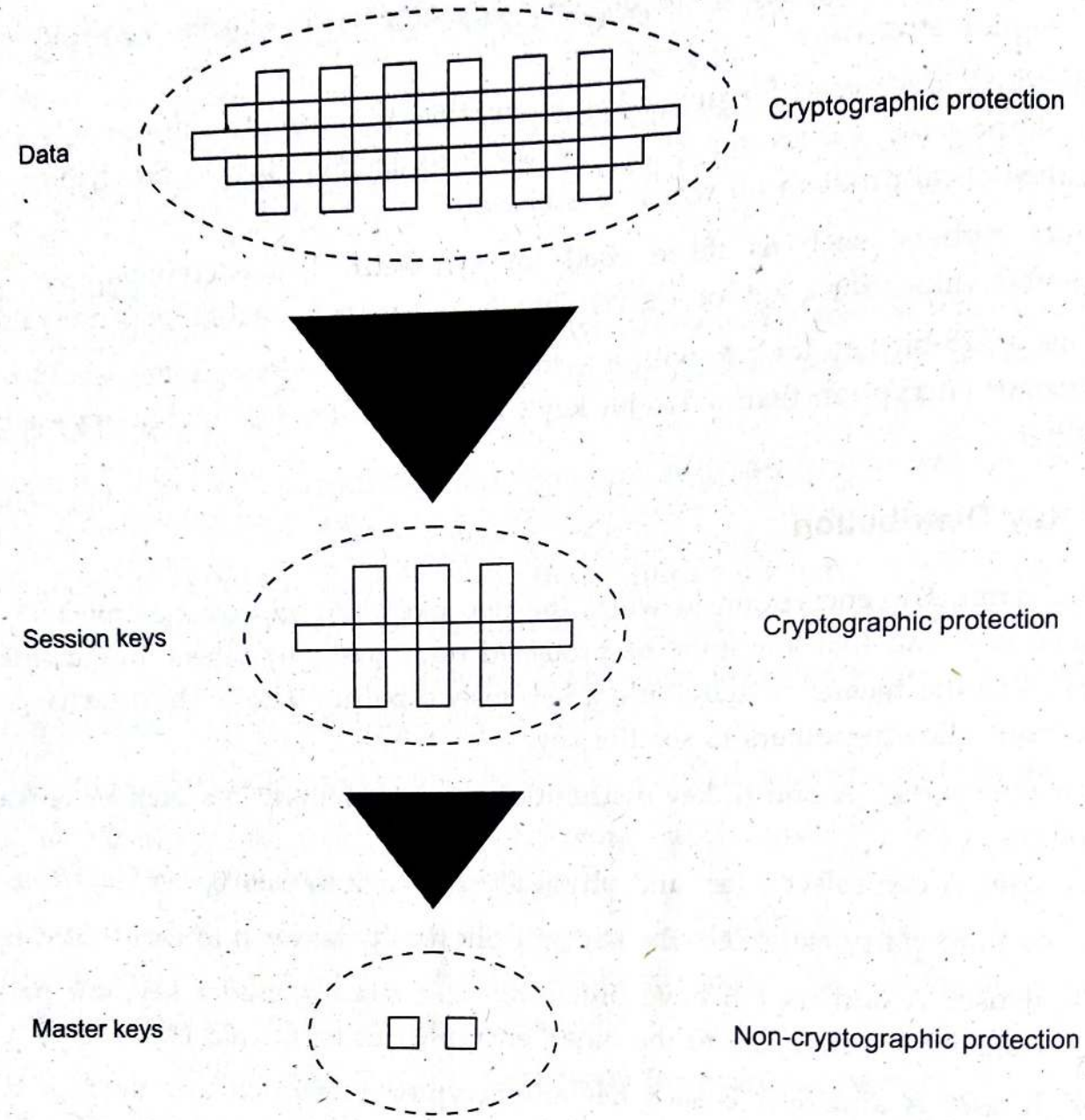

Fig. 2.1.8 Use of a key hierarchy

- Session keys are transmitted in encrypted form, using a **master key** that is shared by the key distribution center and an end system or user. For each end user, there is a unique master key that it shares with the key distribution center.

A key distribution scenario

- User A wishes to establish a logical connection with user B and requires a one time session key to protect the data transmitted over the connection. User A has a master key (K_a), known only to itself and the KDC. User B shares the master key K_b with the KDC. The following steps occur :
  1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier (N_1) for this transaction.
  2. KDC responds with a message encrypted using K_a.

3. A stores the session key for use in the upcoming session and forward to B the information that originated at the KDC for B :
4. User B sends a nonce N_2 to A.

- Fig. 2.1.9 shows the key distribution scenario.

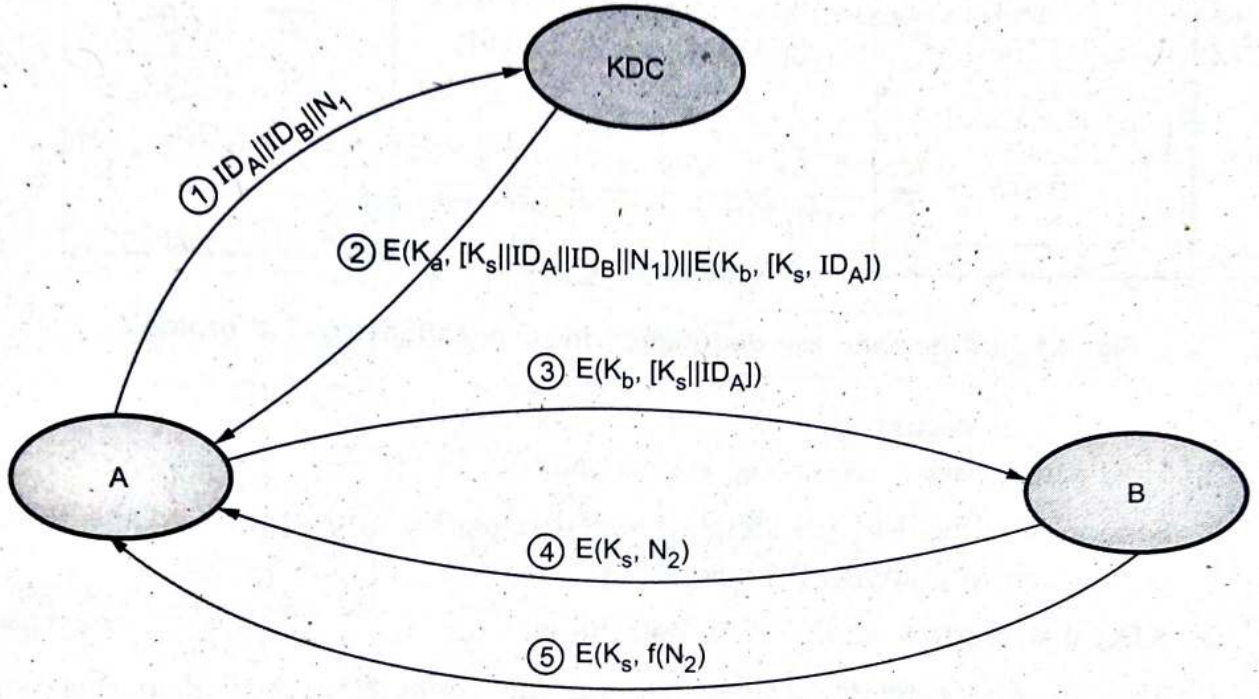

Fig. 2.1.9 Key distribution scenario

- Steps 1, 2 are used for key distribution and steps 3, 4, 5 for authentication.

Session key lifetime

1. **For connection-oriented protocol**
   - Use the same session key for the length of time that the connection is open. Use new session key for each new session.
   - For long lifetime, change the session key periodically.
2. **For connectionless protocol**
   - The most secure approach is to use a new session key for each exchange. For connectionless protocol, such as a transaction-oriented protocol, there is no explicit connection initiation or termination.

Transparent key control scheme

- Fig. 2.1.10 shows automatic key distribution for connection - oriented protocol.
- Assume that communication make use of a connection-oriented end-to-end protocol, such as TCP.

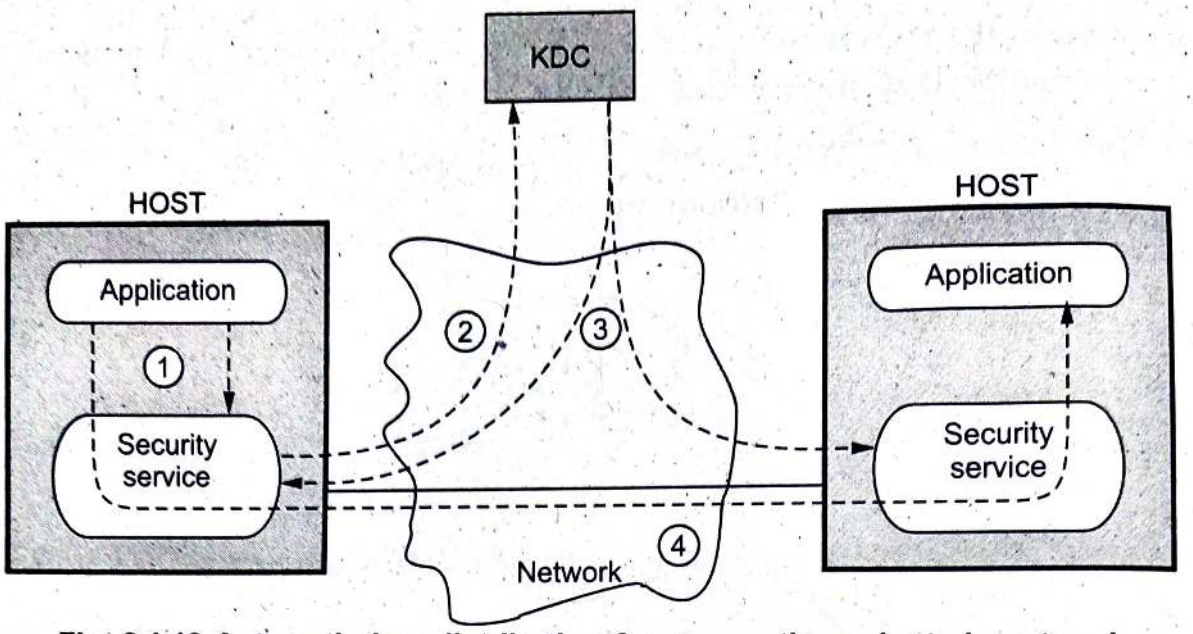

Fig. 2.1.10 Automatic key distribution for connection-oriented protocol

- Following steps occurs :
  1. Host sends packet requesting connection.
  2. Session Security Module (SSM) saves that packet and applies to the KDC for permission to establish the connection.
  3. KDC distributes session key to both hosts.
  4. The requesting SSM can now release the connection request packet, and a connection is set up between the two end systems.

Decentralized key control

- Decentralized approach requires that each end system be able to communicate in a secure manner with all potential parter end systems for purposes of session key distribution.
- A session key may be established with the following sequence of steps.
  1. A issues a request to B for a session key and includes a none, N_1.
  2. B responds with a message that is encryped using the shared master key.
  3. Using the new session key, A returns f (N_2) to B.

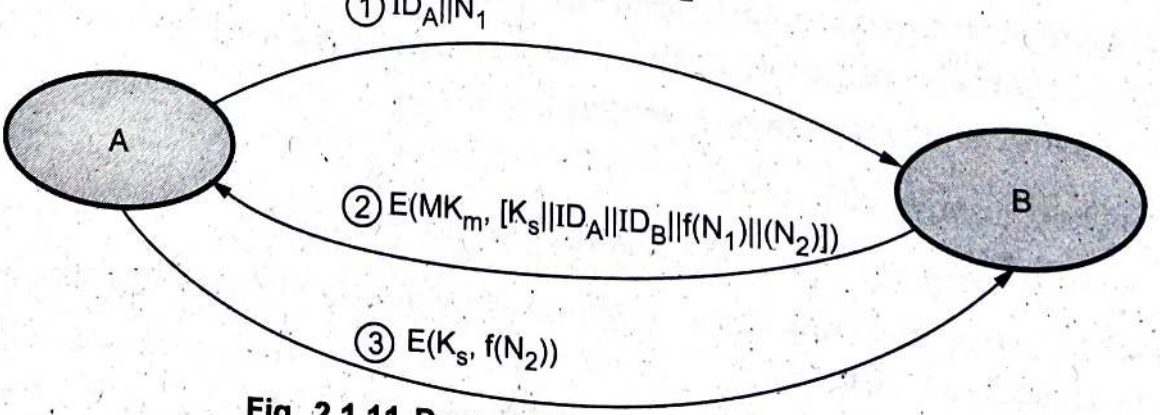

Fig. 2.1.11 Decentralized key distribution

2.2 X.509 Certificates

AU : May-18,19, Dec.-21

- X.509 is part of X.500 recommandations for directory service i.e. set of servers which maintains a database of information about users and other attributes.
- X.509 defines authentication services e.g. certificate structure and authentication protocols. Also X.509 also defines alternative authentication protocols base on use of public-key certificates. The X.509 certificate format is emplied in S/MIME, IP security, SET and SSL/TLS.
- X.509 standard uses RSA algorithm and hash function for digital signature. Fig. 2.2.1 shows generation of public key certificate.

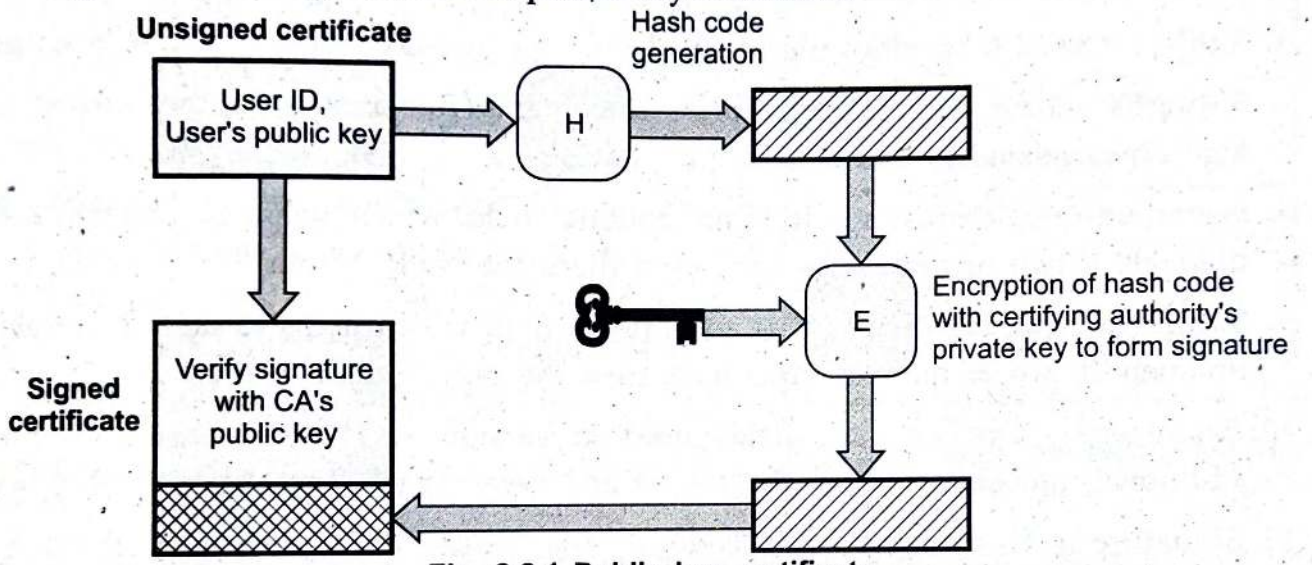

Fig. 2.2.1 Public key certificate

2.2.1 X.509 Format of Certificate

- The current version of the standard is version 3, called as X.509V3. The general format of digital certificate X.509V3 is shown in Fig. 2.2.2.

1	Version
2	Certificate Serial Number
3	Signature Algorithm Identifier
4	Issuer Name
5	Period of Validity
6	Subject Name
7	Subject's Public Key Info.
8	Issuer Unique Identifier
9	Subject Unique Identifier
10	Extensions
11	Signature

Fig. 2.2.2 X.509 Digital certificate format version 3

1. **Version** : Identifies successive versions of certificate format the default is version.
2. **Certificate serial number** : It contains an unique integer number, which is generated by Certification Authority (CA).
3. **Signature algorithm identifier** : Identifies the algorithm used by the CA to sign the certificate.
4. **Issuer name** : Identifies the distinguished name of the CA that created and signed this certificate.
5. **Period of validity** : Consists of two date-time values (not before and not after) within which the certificate is valid.
6. **Subject name** : It specifies the name of the user to whom this certificate is issued.
7. **Subject's public key information** : It contains public key of the subject and algorithms related to that key.
8. **Issuer unique identifier** : It is an optional field which helps to identify a CA uniquely if two or more CAs have used the same Issuer Name.
9. **Subject unique identifier** : It is an optional field which helps to identify a subject uniquely if two or more subjects have used the same Subject Name.
10. **Extensions** : One or more fields used in version 3. These extensions convey additional information about the subject and issuer keys.
11. **Signature** : It contains hash code of the fields, encrypted with the CA's private key. It includes the signature algorithm identifier.

Standard notations for defining a certificate

- $CA\langle\langle A\rangle\rangle = CA\{V, SN, AI, CA, T_A A, A_P\}$

where,

$CA\langle\langle A\rangle\rangle$ indicates the certificate of user A issued by certification authority CA.

$CA\{V \ldots\ldots A_P\}$ indicates signing of $V \ldots\ldots A_P$ by CA.

2.2.2 Obtaining User's Certificate

- The characteristics of user certificate are -
  1. Any user who can access public key of CA can verify user public key.
  2. Only certification Authority (CA) can modify the certificate.
- All user certificates are placed in a directory for access of other users. The public key provided by CA is absolutely secure (w.r.t. integrity and authenticity).
- If user A has obtained a certificate from CA X_1 and user B has obtained a certificate from CA X_2. If A don't know the public key of X_2, then B's certificate

(issued by X_2) is useless to A. The user A can read B's certificate but A can not verify the signature. This problem can be resolved by securely exchanging the public keys by two CAs.

2.2.3 Revocation of Certificates

- The certificate should be revoked before expiry because of following reasons :
  1. User's private key is compromised.
  2. User is not certified by CA.
  3. CA's certificate is compromised.
- Each CA has a list of all revoked but not expired certificates. The Certificate Revocation List (CRL) is posted in directory signed by issuer and includes issuer's name, date of creation, date of next CRL. Fig. 2.2.3. Certificate revocation list. Each certificate has unique serial number of identify the certificate.

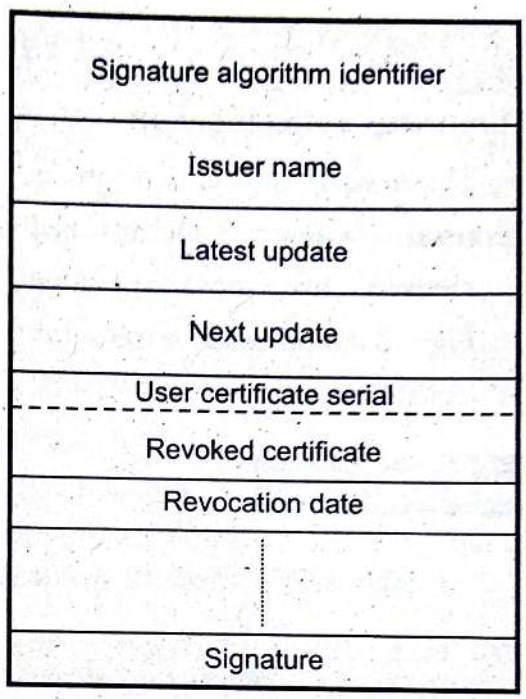

Fig. 2.2.3 Certificate revocation list

2.2.4 Authentication Procedures

- X.509 supports three types of authenticating using public key signatures. The types of authentication are
  1. One-way authentication
  2. Two-way authentication
  3. Three-way authentication

1. One-way authentication

- It involves single transfer of information from one user to other as shown in Fig. 2.2.4.

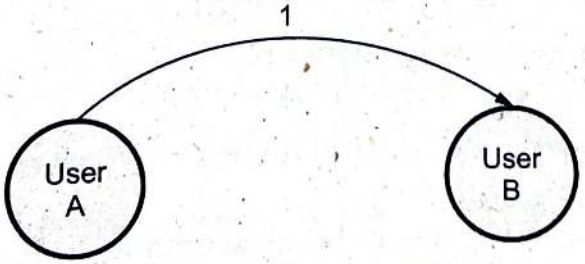

Fig. 2.2.4 One way authentication

2. Two-way authentication

- Two-way authentication allows both parties to communicate and verify the identity of the user.

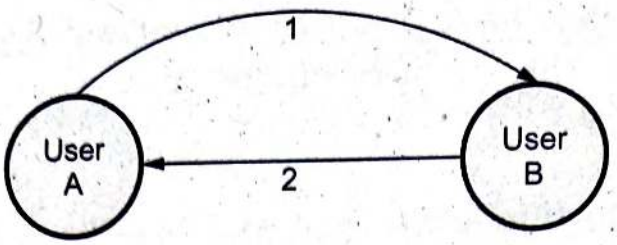

Fig. 2.2.5 Two-way authentication

3. Three-way authentication

- Three-way authentication is used where synchronized clockes are not available Fig. 2.2.6 shows three-way authentication.

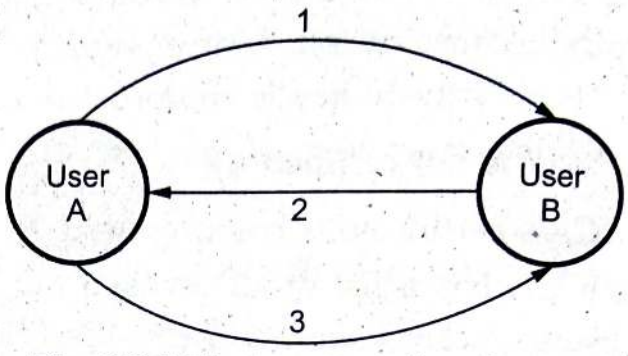

Fig. 2.2.6 Three-way authentication

Review Questions

1. *Explain briefly about the architecture and certification mechanisms in Kerberos and X.509.* **AU : May-18, Marks 16**
2. *Explain the format of the X.509 certificate.* **AU : May-19, Marks 6**
3. *Shortly describe about the elements of X509 certificate.* **AU : Dec.-21, Marks 9**

2.3 Public-Key Infrastructure

- Management and handling of the pieces of secret information is generally referred to as **key management.**
- Activities of key management includes selection, exchange, storage, certification, revocation, changing, expiration and transmission of the key.
- Key management is the set of processes and mechanisms which support key establishment and maintenance of ongoing keying relationship between parties, including replacing older key with new keys.
- Two major issues in key management are :
  1. Key life time
  2. Key exposure

Key life time - limit of use which can be measured as a duration of time.

Issue related to key :

1. Users must be able to obtain securely a key pair suited to their efficiency and security needs.
2. Keys need to be valid only until a specified expiration date.

3. The expiration date must be chosen properly and publicized securely.
4. User must be able to store their private keys securely.
5. Certificates must be unforgettable, obtainable in a secure manner.

1. Public key infrastructure

- Public Key Infrastructure (PKI) is a well-known technology that can be used to establish identities, encrypt information and digitally sign documents.
- PKI identifies and manages relationships of parties in an electronic exchange, serving a wide array of security needs including access control, confidentiality, integrity, authentication and non-repudiation.
- PKI also uses unique Digital Certificates (DC) to secure eCommerce, email, data exchange and Virtual Private Networks (VPN) and intranets and is also used to verify the identity and privileges of each user.
- The Certificate Authority (CA) provides a full life-cycle management of public keys and certificates, including issuance, authentication, storage, retrieval, backup, recovery, updating and revocation to the PKI.
- All users of PKI must have a registered identity, which is stored in a digital certificate issued by a CA.
- Remote users and sites using public private keys and public key certificates can authenticate each other with a high degree of confidence.
- Authentication is dependent on three conditions :
  1. It must be established that each party have a private key that has not been stolen or copied from the owner.
  2. The certificate must be issued to the owner in accord with the stated policy of the certificate issuer.
  3. The policies of the certificate issuer must be satisfactory to the parties so as to verify identity.

Benefits of PKI

1. Confidential communication : Only intended recipients can read files.
2. Data integrity : Guarantees files are unaltered during transmission.
3. Authentication : Ensures that parties involved are who they claim to be.
4. Non-repudiation : Prevents individuals from denying.

Limitation of PKI

The problems encountered deploying a PKI can be categorized as follows :

1. Public key infrastructure is new

2. Lack of standards
3. Shortage of trained personnel
4. Public key infrastructure is mostly about policies.

2. Certificate

- **Certificates** are digital documents that are used for secure authentication of communicating parties.
- A certificate binds identity information about an entity to the entity's public key for a certain validity period.
- A certificate is digitally signed by a Trusted Third Party (TTP) who has verified that the key pair actually belongs to the entity.
- Certificates can be thought of as analogous to passports that guarantee the identity of their bearers.
- **Authorities :** The trusted party who issues certificates to the identified end entities is called a **Certification Authority (CA).**
- Certification authorities can be thought of as being analogous to governments issuing passports for their citizens.
- A certification authority can be managed by an external certification service provider or the CA can belong to the same organization as the end entities.
- CAs can also issue certificates to other (sub) CAs. This leads to a tree-like **certification hierarchy.**
- The highest trusted CA in the tree is called a root CA.
- In large organizations, it may be appropriate to delegate the responsibility for issuing certificates to several different certificate authorities.
- For example, the number of certificates required may be too large for a single CA to maintain; different organizational units may have different policy requirements; or it may be important for a CA to be physically located in the same geographic area as the people to whom it is issuing certificates.
- The X.509 standard includes a model for setting up a hierarchy of the Certification Authority.
- Fig. 2.3.1 shows the hierarchy of certificate authorities.
- In the Fig. 2.3.1, the root CA is at the top of the hierarchy. The root CA's certificate is a self-signed certificate : that is, the certificate is digitally signed by the same entity -- the root CA.

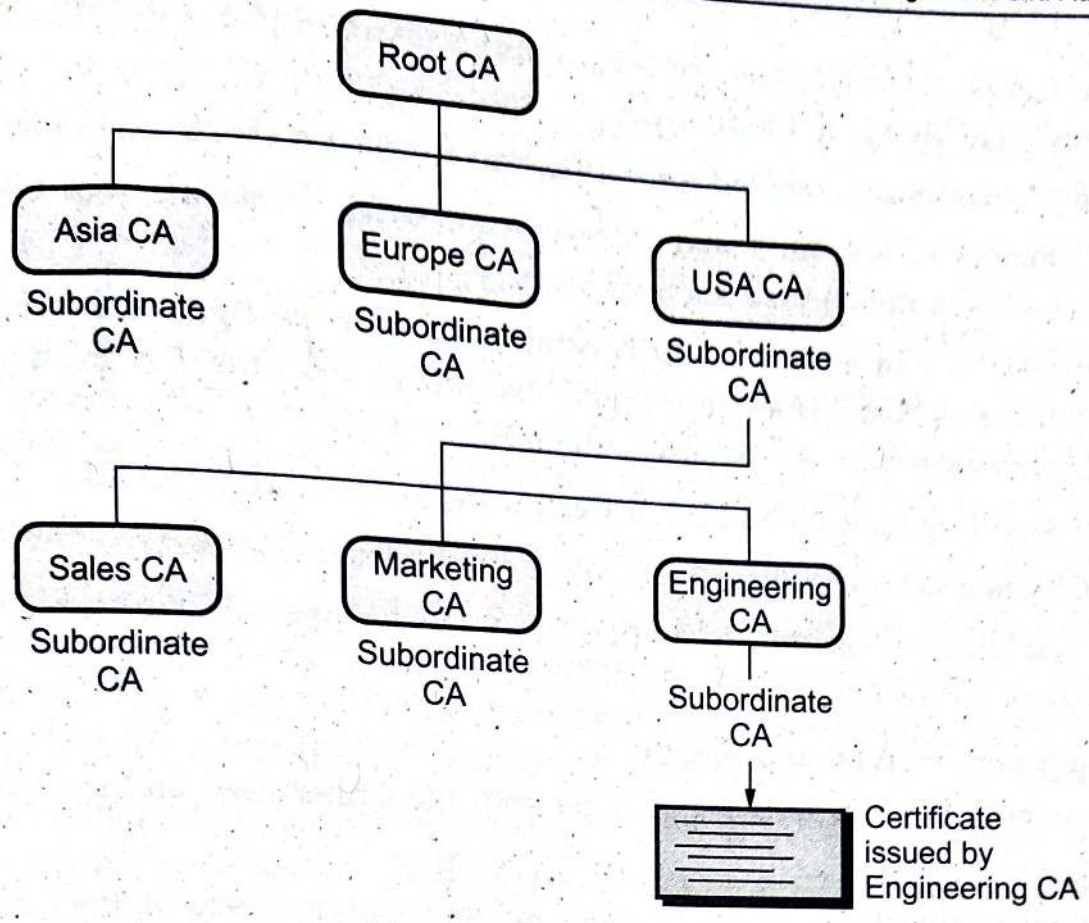

Fig. 2.3.1 Hierarchy of CA

- The CAs, that are directly subordinate to the root CA, have CA certificates signed by the root CA. CAs under the subordinate CAs in the hierarchy have their CA certificates signed by the higher-level subordinate CAs.
- Organizations have a great deal of flexibility in terms of the way they set up their CA hierarchies.
- **Certificate chains :** Certificate chain is series of certificates issued by successive CAs.
- In some cases, a CA can delegate the actual identification of end entities as well as some other administrative tasks to a separate entity, the **Registration Authority (RA).**

Verifying certificates

- When authentication is required, the entity presents a signature it has generated from authentication data using its private key, and a certificate corresponding to that key.
- The receiving entity can verify the signature with the public key of the sender contained in the certificate.
- Next the receiving entity must verify the certificate itself by checking the validity time of the certificate and the signature of the CA in the certificate.

- If the CA is a sub CA, the receiving entity must also verify the signatures of all higher-level CAs up to the root CA.
- The list of certificates needed for verification is called a **certification path.**
- If all signatures are valid and the receiving entity trusts the root CA, the first entity will be authenticated successfully.
- If a private key of an end entity is compromised or the right to authenticate with a certificate is lost before its natural expiration date, the CA must revoke the certificate and inform all PKI users about this.
- The CA will periodically publish a **Certificate Revocation List (CRL).**
- The CRL is a list identifying the revoked certificates and it is signed by the CA.
- The end entities should check the latest CRL whenever they are verifying a validity of a certificate.

3. Key length and encryption strength

- The strength of encryption depends on both the cipher used and the length of the key.
- Encryption strength is often described in terms of the size of the keys used to perform the encryption : in general, longer keys provide stronger encryption.
- Key length is measured in bits. For example, 128-bit keys for use with the RC4 symmetric-key cipher supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher.
- Roughly speaking, 128-bit RC4 encryption is 3×10^{26} times stronger than 40-bit RC4 encryption.
- Different ciphers may require different key lengths to achieve the same level of encryption strength.
- The RSA cipher used for public-key encryption, for example, can use only a subset of all possible values for a key of a given length, due to the nature of the mathematical problem on which it is based.
- Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values.
- Thus a 128-bit key for use with a symmetric key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

2.4 User Authentication

- User authentication is a security process that prevents unauthorized users from accessing device or network. It's a login procedure where an application requests personalized passwords to give us authorized access to it. If a user lacks the proper login rights to the network, their authentication fails.
- User authentication operates on advanced technology. A hacker trying to force their way into the secured network would have to go high and above to bypass it. If there are other cybersecurity measures such as intrusion detection systems on the network, the attacker will be detected before they gain access.

➢ Why is user authentication important ?

- Targeting unsuspecting victims is the day job of cybercriminals. As an active online user, protecting your devices against unauthorized access is necessary to stay safe. From shopping online to e-learning and connecting with peers, we leave digital footprints that hackers can trace and manipulate to compromise our device.
- User authentication is effective in reducing cyber threats to the barest minimum. The antics of attackers only hold water if they get into network. The authentication is like a barricade that locks them out. As long as it's strong, they can't pull it down.
- User authentication enforces confidentiality, establishes trust and guarantees privacy. Visitors to network will be willing to spend a minute or two on the authentication process along as it secures them from attacks.

2.5 Remote User Authentication Principles

- Authentication Protocols are used to convince parties of each other's identity and to exchange session keys. They may be one-way or mutual.
- Authentication techniques are used to verify identity. The authentication of authorized users prevents unauthorized users from gaining access to corporate information systems. Authentication method is of validating the identity of user, service or application. The use of authentication mechanisms can also prevent authorized users from accessing information that they are not authorized to view.
- User is authenticated by four ways :
  1. Something the individual knows : Examples include a password, a personal identification number (PIN).
  2. Something the individual possesses : Examples include cryptographic keys, electronic keycards, smart cards and physical keys.
  3. Something the individual is : Examples include recognition by fingerprint, retina and face.

4. Something the individual does : Examples include recognition by voice pattern, handwriting characteristics etc.

2.5.1 Mutual Authentication

- Mutual authentication allows for both ends to know that they truly know whom they are communicating with.
- Central to the problem of authenticated key exchange are two issues : Confidentiality and timeliness.
- To prevent masquerade and to prevent compromise of session keys, essential identification and session key information must be communicated in encrypted form.
- This requires the prior existence of secret or public keys that can be used for this purpose. The second issue, timeliness, is important because of the threat of message replays. Timeline prevent the replay attacks.
- This requires the prior existence of secret or public keys that can be used for this purpose. The second issue, timeliness, is important because of the threat of message replays. Timeline prevent the replay attacks.

Examples of replay attacks

1. Simply replay : The opponent simply copies a message and replays it later.
2. Repetition that can be logged : Replay time stamped message within valid time.
3. Repetition that cannot be detected : Original message suppressed and only reply message arrives.
4. Backward replay without modification.

Replay attack countermeasures

- Replay Attacks are where a valid signed message is copied and later resent. Such replays, at worst, could allow an opponent to compromise a session key or successfully impersonate another party.
- At minimum, a successful replay can disrupt operations by presenting parties with messages that appear genuine but are not.
- Possible countermeasures include the use of :
  1. Sequence numbers : Generally impractical since must remember last number used with every communicating party.
  2. Timestamps : Needs synchronized clocks amongst all parties involved, which can be problematic.
  3. Challenge/response : Using unique, random, unpredictable nonce, but not suitable for connectionless applications because of handshake overhead.

2.5.2 One Way Authentication

- It involves single transfer of information from one user to other.
- Client authenticates itself to the server. The server may or may not be authenticated to the client. This is referred to as one way authentication.

2.5.2.1 Password based Authentication

- Password is a front line protection against the unauthorized access (intruder) to the system. A password authenticate the identifier (ID) and provides security to the system. Therefore almost all systems are password protected.

1] Password vulnerability

Passwords are extremely common. Passwords can often be guessed. Use of mechanisms to keep passwords secret does not guarantee that the system security can not be broken. It only says that it is difficult to obtain passwords. The intruder can always use a trial and error method. A test of only a limited set of potential strings tends to reveal most passwords because there is a strong tendency for people to choose relatively short and simple passwords that they can remember. Some techniques that may be used to make the task of guessing a password difficult are as follows

1. Longer passwords.
2. Salting the password table.
3. System assistance in password selection.

The length of a password determines the ease with which a password can be found by exhaustion. For example, 3-digit password provides 1000 variations whereas a four digit passwords provides 10,000 variations. Second method is the system assistance. A password can be either system generated or user selected. User selected passwords are often easy to guess. A system can be designed to assist users in using passwords that are difficult to guess.

2] Encrypted passwords

Instead of storing the names and passwords in plain text form, they are encrypted and stored in cipher text form in the table. In this case, instead of directly using a user specified name and password for table lookup, they are first encrypted and then the results are used for table lookup. If the stored encoded password is seen, it can not be loaded, so the password cannot be determined. The password file does not need to be kept secret.

3] One time passwords

Set of paired passwords solve the problem of password sniffing. When a session begins, the system randomly selects and presents one part of a password pair; user must supply the other part. In this, user is challenged and must respond with the correct

answer to that challenge. In this method, the password is different in each instance. One time passwords are among the only ways to prevent improper authentication due to password exposure. Commercial implementations of one time password system such as secur ID, use hardware calculators.

Password selection strategies

- Too short password is too easy to guess. If the passward is 8 random character, it is impossible to crack the password. In order to eliminate gaussable passwords four basic techniques are suggested.
  1. User education
  2. Computer generated password
  3. Reactive password checking
  4. Proactive password checking

2.6 Remote User-Authentication using Symmetric Encryption

1. Mutual authentication :

- The Needham Schroeder protocol refers to two methods of communication protocols through an insecure network.
  1. Needham Schroeder symmetric key protocol, which is based on symmetric encryption algorithm to establish a session key between two parties in a network.
  2. Needham Schroeder public-key protocol, based on the public key cryptography to provide mutual authentication between two communication parties over a network.

Needham schroeder public key authentication protocol

- The Needham Schroeder public key authentication protocol aims to provide a mutual authentication between two parties Alice (A) and Bob (B).
- Both parties want to insure each other identity before starting to communicate.
- The protocol is as follows :
  a. K_A and K_B are Alice's public key and Bob's public key respectively,
  b. N_A and N_B are nonces generated by A and B respectively.

1. $A \rightarrow B : \{N_A, A\}_{K_B}$ (Init)

Alice generates a nonce N_A and sends it to Bob with her identity. Everything is encrypted using Bob's public key.

2. $B \rightarrow A : \{N_A, N_B\}_{K_A}$ (Challenge)

Bob generates a nonce N_B, and sends it to Alice with N_A he has just received. It is a way to prove that he is really the owner of the private key corresponding to K_B. In

other word, this mechanism is implemented in order to authenticate Bob. Sending back to Alice N_A is also a way to avoid a replay of this message.

3. $A \rightarrow A : \{N_B\}_{K_B}$ (Response)

Alice decrypts the message and check if it contains the right value of N_A. Then, she sends back N_B to Bob to prove her ability to decrypt with her private key and so to authenticate herself.

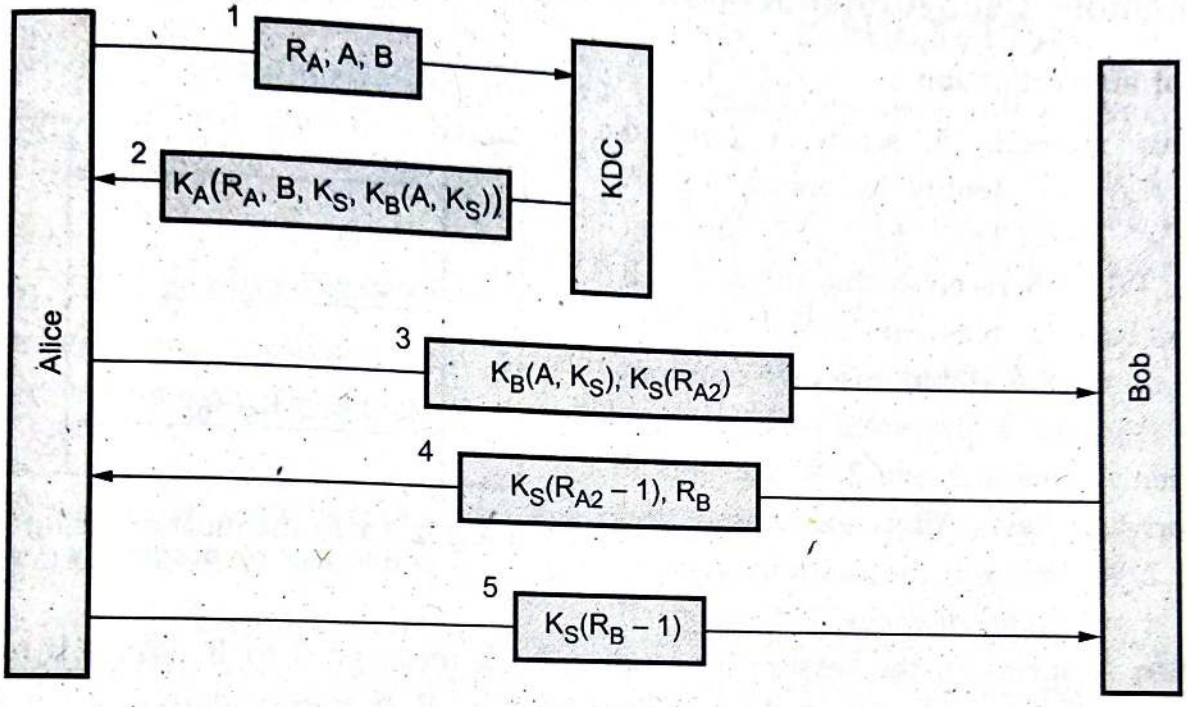

Fig. 2.6.1 Needham Schroeder authentication protocol

2. Based on shared secret key :

In this protocol, a secret key is shared with both party. i.e. source and destination. One party sends random number to the other, other side transforms it in a special way and then returns a result. This type of protocols are called challenge-response protocols. The working of this protocol is as follows.

First the party 1 sends a message 1 to party 2 i.e. identification of party 1. The party 2 needs to find out the message which it received is from party 1 or any other third party. Party 2 sends a large random number to party 1 in plaintext. The party 1 then encrypts the message with the key which shares with party 2 and sends the ciphertext back in message 3. When party 2 receives this message, they know that message is from party 1 because of the shared secret key. Uptill now

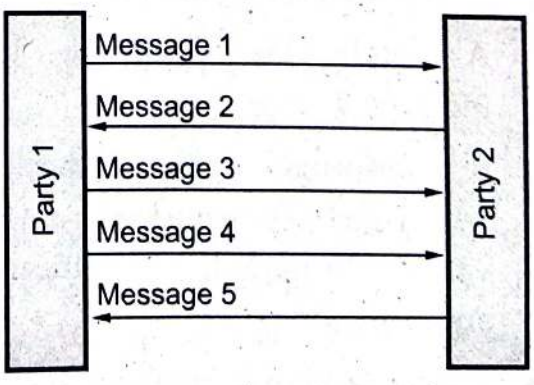

Fig. 2.6.2 Two way authentication using a challenge-response protocol

party 2 is sure only about communication, but party 1 is not sure about the communication between him and party 2. The party 1 sends a random number to party 2 as plaintext in message 4. When party 2 responds with secret key, party 1 knows they are communicating with party 2. This protocol has some disadvantages. It is slower and contains extra messages. These can be eliminated by combining information.

2.7 Remote User-Authentication Using Asymmetric Encryption

1. Mutual authentication :

In this method, A sends a random number R_A and identity by encrypting. A uses B's public-key E_B for sending message. When B receives this messages, B sends A back a message containing A's random number R_A and his own random number R_B and a proposed session key, K_s. When A gets message 2, A decrypts it using private key. After examining the message 2, A finds out the random number R_A. A knows that message 2 is from B only. Then A agrees to the session by sending back message 3 to B. When B reads R_B encrypted with the session key which is generated by B, B knows that A got message 2 and verified R_A.

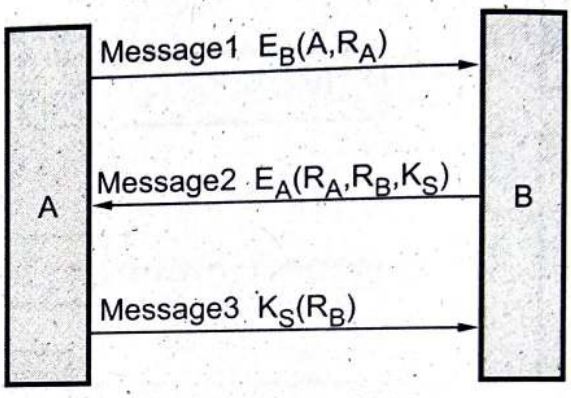

Fig. 2.7.1 Authentication using public key cryptography

This protocol does has some disadvantage. It assumes that both user (A and B) already know each others public keys.

2. Certificate based authentication :

- Client have a public key certificate. Fig. 2.7.2 shows the certificate based authentication.
- A sends his certificate in message 1.
- B performs certain checks which includes principal name, validity period, certificate authority etc.
- B then sends his challenge i.e a nonce R.

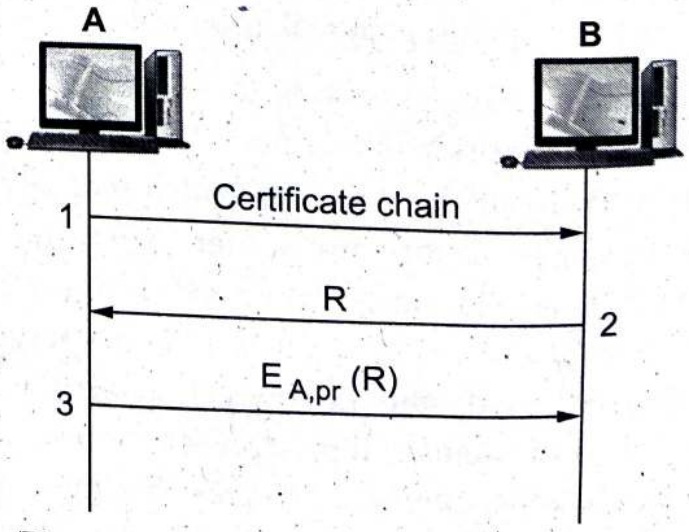

Fig. 2.7.2 Certificate based authentication

- A responds by encrypting the challenge with his private key.
- When B receives $E_{A,pr}(R)$, he decrypts it with A's public key and compares it with the nonce he transmitted in message 2.
- It they match, he concludes that A has used the private key corresponding to the public key in his certificate.

2.8 Kerberos Systems

AU : May-14,15,18,19, Dec.-21

- Kerberos is an **authentication protocol**. It provides a way to authenticate clients to services to each other through a trusted third party.
- Kerberos makes the assumption that the connection between a client and service is insecure. Passwords are encrypted to prevent others from reading them. Clients only have to authenticate once during a pre-defined lifetime.
- Kerberos was designed and developed at MIT by Project Athena. Currently, Kerberos is upto Version 5. Version 4 being the first version to be released outside of MIT.
- Kerberos has been adopted by several private companies as well as added to several operating systems.
- Its creation was inspired by client-server model replacing time-sharing model. Kerberos is a network authentication protocol designed to allow users, clients and servers, authenticate themselves to each other.
- This **mutual authentication** is done using **secret-key cryptography** with parties proving to each other their identity across an insecure network connection.
- Communication between the client and the server can be secure after the client and server have used Kerberos to prove their identity.
- From this point on, subsequent communication between the two can be encrypted to assure privacy and data integrity.

Requirement of Kerberos

- Kerberos client/server authentication requirements are :
  1. **Security :** That Kerberos is strong enough to stop potential eavesdroppers from finding it to be a weak link.
  2. **Reliability :** That Kerberos is highly reliable employing a distributed server architecture where one server is able to back up another. This means that Kerberos systems are fail safe, meaning graceful degradation, if it happens.
  3. **Transparency :** That user is not aware that authentication is taking place beyond providing passwords.
  4. **Scalability :** Kerberos systems accept and support new clients and servers.

- To meet these requirements, Kerberos designers proposed a third-party trusted authentication service to arbitrate between the client and server in their mutual authentication.

2.8.1 Kerberos Terminology

- Kerberos has its own terminology to define various aspects of the service.
  1. **Authentication Server (AS) :** A server that issues tickets for a desired service which are in turn given to users for access to the service.
  2. **Client :** An entity on the network that can receive a ticket from Kerberos.
  3. **Credentials :** A temporary set of electronic credentials that verify the identity of a client for a particular service. It also called a ticket.
  4. **Credential cache or ticket file :** A file which contains the keys for encrypting communications between a user and various network services.
  5. **Crypt hash :** A one-way hash used to authenticate users.
  6. **Key :** Data used when encrypting or decrypting other data.
  7. **Key Distribution Center (KDC) :** A service that issue Kerberos tickets and which usually run on the same host as the Ticket-Granting Server (TGS).
  8. **Realm :** A network that uses Kerberos composed of one or more servers called KDCs and a potentially large number of clients.
  9. **Ticket-Granting Server (TGS) :** A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.
  10. **Ticket-Granting Ticket (TGT) :** A special ticket that allows the client to obtain additional tickets without applying for them from the KDC.

2.8.2 Kerberos Version 4

- Kerberos version 4 uses DES for providing authentication service. Some aspect of version 4 are

  A) Simple Authentication Dialogue. B) More Secure Authentication Dialogue.

2.8.2.1 Simple Authentication Dialogue

- For a secure transaction, server should confirm the client and its request. In unprotected network it creates burden on server, therefore an authentication server (AS) is used. The authentication server (AS) maintains password of all users in centralized database. Also the authentication server shares a unique secret key with each server.

- Let

  Client is represented as C

  Authentication server is represented as AS

  Server is represented as V

  Identifier of user on C is represented as ID_C

  Identifier of V is represented as ID_V

  Password of user on C is P_C

  Network address of C is represented as AD_C

  Secret encryption key shared by AS and V is K_V

  Then consider a hypothetical dialogue.

	Sender and receiver		Contents of message
1.	$C \rightarrow AS$	:	$ID_C \| P_C \| ID_V$
2.	$AS \rightarrow C$	:	Ticket
3.	$C \rightarrow V$	:	$ID_C \|$ Ticket
4.	Ticket	=	$E [K_V , (ID_C \| AD_C \| ID_V)]$

Explanation

1. **Client Clogs on to workstation requesting to access to server V :** The workstation requests user's passward and sends message to AS including user ID + server ID + user passward. The AS checks this message with database and verifies it.
2. **AS issues ticket :** On verifying the tests. AS issues ticket containing user ID + server ID + network address.
3. **Client C applies server V :** With this ticket, client C asks server V for access. Server V decrypts the ticket and verify the aunthenticity of data then grants the requested service. In above hypothetical dialogue, symbol || represents concatenation.

2.8.2.2 Secure Authentication Dialogue

- Kerberos version 4 protocol ensures secure authentication dialogue involving three sessions.

  i] Authentication Service - Exchange to obtain ticket-granting ticket.

  ii] Ticket-granting Service - Exchange to obtain service granting ticket.

  iii] Client/server authentication - Exchange to obtain service.

- Each of the above session has two steps, as shown in table below

Session	Step	Sender-Receiver
[i]	1.	C → AS
	2.	AS → C
[ii]	3.	C → TGS (Ticket-granting server)
	4.	TGS → C
[iii]	5.	C → V
		V → C

- Fig. 2.8.1 shows how the steps are executed in Kerberos version 4.

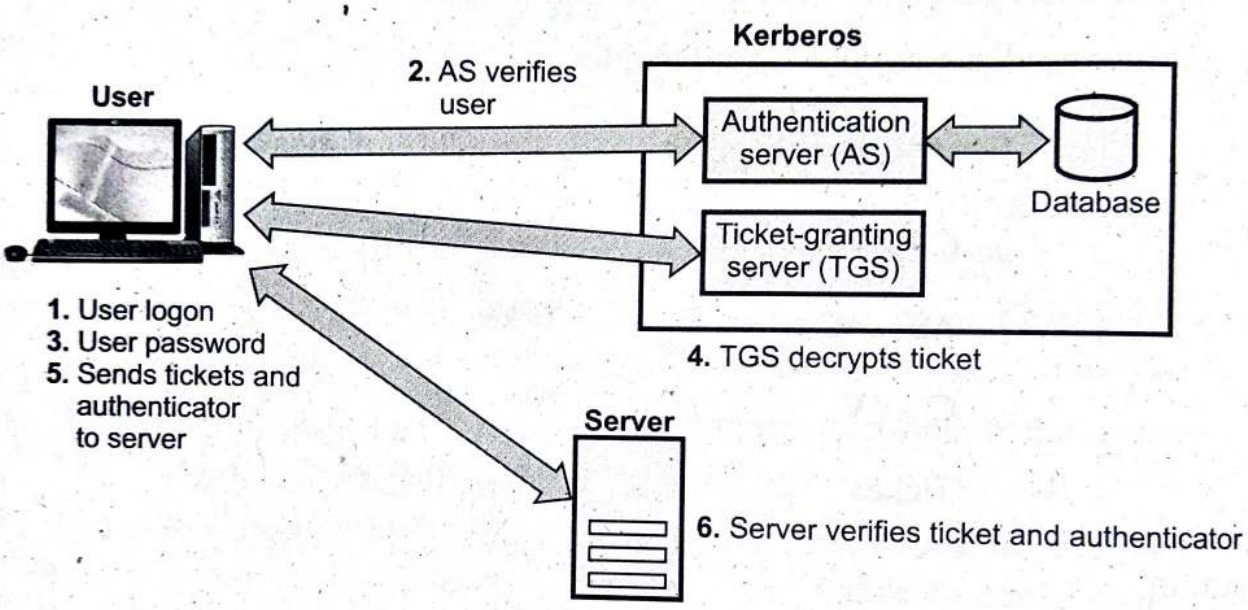

Fig. 2.8.1 Overview of kerberos

2.8.2.3 Kerberos Realms

- The constituents of a full-service Kerberos environment are,
  a) A Kerberos server
  b) Clients
  c) Number of application server.
- Requirements of Kerberos sever :
  a) Kerberos server should have user ID.
  b) Hashed password for all users.
  c) All users should be registered with Kerberos server.
  d) Kerberos server should have secret key with each server.
  e) All servers should be registered with Kerberos server.

- A **Kerberos realm** is referred as is the environment where
  - All nodes share same secured database.
  - Changing and accessing the Kerberos database requires Kerberos master password.
  - A read only copy of Kerberos database resides in computer system.
- Networks have different realms under different administrative organizations. The users of one realm may access the servers in other realm provided the users are authenticated. The interoperating Kerberos shares a secret key with the server in other realm.

2.8.3 Kerberos Version 5

- Version 4 of Kerberos have some environmental shortcoming and technical deficiencies.

Environmental shortcomings of version 4

1. Encryption system dependence
2. Internet protocol dependence
3. Message byte ordering
4. Ticket lifetime
5. Authentication forwarding
6. Inter realm authentication.

Technical deficiencies of version 4

1. Double encryption
2. PCBC (Propagating Cipher Block Chaining) encryption
3. Session keys
4. Password attacks

2.8.3.1 Version 5 Authentication Dialogue

- The Kerberos version 5 message exchange involves three session, these are
  1. Authentication Service Exchange
  2. Ticket - Granting Exchnage
  3. Client/Server Authentication Exchange.
- Each session has two steps. Table 2.8.1 summarizes session, steps and their functions.

	Session	Step	Function
[i]	Application Service Exchange	C → AS AS → C	To obtain ticket-granting ticket.
[ii]	Ticket-Granting Service Exchange	C → TGS TGS → C	To obtain service-granting ticket.
[iii]	Client/Server Authentication Exchange	C → V V → C	To obtain service.

Table 2.8.1

- The flags field is expanded in ticket in version 5 of Kerberos. Various flags that may be included in a ticket, are

i) INITIAL ii) PRE - AUTHENT iii) HW-AUTHENT

iv) RENEWABLE v) MAY-POSTDATE vi) POSTDATED

vii) INVALID viii) PROXIABLE ix) PROXY

x) FORWARDABLE xi) FORWARDED

2.8.4 Comparison between Kerberos Versions 4 and 5

Parameters	Kerberos Versions 4	Kerberos Versions 5
Encryption algorithms used	DES only	DES and other encryptions
Ticket lifetime	5 min units, Maximum = 1280 minutes	Start and end time is arbitrary
Message byte ordering	Tagged message with ordering	Abstract syntax notation on basis encoding rules.
Password attack	Initial request in clear and use it for offline attack.	Need to send pre-authentication data
Two times encryption	Supported	Not supported
Session Keys	Replay risk using repeated ticket	Sub session key once only
Hierarchy of Realms	Limits to pairs	Transition allowed

2.8.5 Strengths of Kerberos

1. Passwords are never sent across the network unencrypted. This prevents those unscrupulous people from being able to read the most important data sent over the network.
2. Clients and applications services mutually authenticate. Mutual authentication allows for both ends to know that they truly know whom they are communicating with.
3. Tickets have a limited lifetime, so if they are stolen, unauthorized use is limited to the time frame that the ticket is valid.
4. Authentication through the AS only has to happen once. This makes the security of Kerberos more convenient.
5. Shared secret keys between clients and services are more efficient than public-keys.
6. Many implementations of Kerberos have a large support base and have been put through serious testing.
7. Authenticators, created by clients, can only be used once. This feature prevents the use of stolen authenticators.

2.8.6 Weakness of Kerberos

1. Kerberos only provides authentication for clients and services.
2. Kerberos 4 uses DES, which has been shown to be vulnerable to brute-force-attacks with little computing power.
3. The principal-key database on the KDC has to be hardened or else bad things can happen.
4. Like any security tool, it is also vulnerable to users making poor password choices.
5. Kerberos doesn't work well in a time-sharing environment.
6. Kerberos requires a continuously available Kerberos Server. If the Kerberos Server goes down, the Kerberos network is unusable.
7. Kerberos does not protect against modifications to system software like Trojan horses.

2.8.7 Difference between Kerberos and SSL

No.	Kerberos	SSL
1	Uses private key encryption.	Uses public key encryption.
2	Based on the trusted third party.	Based on certificate.
3	Ideal for network environment.	Ideal for the WWW.
4	Key revocation can be accomplished by disabling a user at the authentication server	Key revocation requires revocation server to keep track of bad certificate.
5	Password resides in user's minds where they are usually not subject to secret attack.	Certificates sit on a user hard drive where they are subject to being cracked.
6	Kerberos open source and free available.	Uses patented material, so the service is not free.

Review Questions

1. *Elaborately explain Kerberos authentication mechanism with suitable diagrams.*
2. *Explain Kerberos Version 4 in detail.* **AU : May-14, Marks 16** **AU : May-15, Marks 16**
3. *Explain briefly about the architecture and certification mechanisms in Kerberos and X.509.*
4. *What is Kerberos ? Explain how it provides authenticated service.* **AU : May-18, Marks 16** **AU : May-19, Marks 7**
5. *Discuss the four requirements of Kerberos.* **AU : Dec.-21, Marks 4**

2.9 Two Marks Questions with Answers

Q.1 Name the four requirements defined by Kerberos. **AU : Dec.-22**

Ans. : Kerberos requirements are secure, reliable, transparents and scalable.

Q.2 What types of attacks are addressed by message authentication ?

Ans. :
- **Content modification :** Changes to the contents of the message.
- **Sequence modification :** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
- **Timing modification :** Delay or replay of messages.

Q.3 What is public-key certificate ?

Ans. : The public-key authority could be a bottleneck in the system, for a User must appeal to the authority for a public key for every other user that it wishes to contact.

As before the directory of names and public keys maintained by the authority is vulnerable to tempering.

Q.4 What are the requirements for the use of a public-key certificate scheme ?

Ans. :
- Any participant can read a certificate to determine the name and public key of the certificate's owner.
- Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
- Only the certificate authority can create and update certificates.
- Any participant can verify the currency of the certificate.

Q.5 What is the life cycle of a key ?

Ans. : Keys have limited lifetimes for a number of reasons. The most important reason is protection against cryptanalysis. Each time the key is used, it generates a number of ciphertexts. Ford describes the life cycle of a key as follows :
- key generation and possibly registration for a public key.
- key distribution
- key activation/deactivation
- key replacement or key update
- key revocation
- key termination, involving destruction and possibly archival.

Q.6 Define password protection.

Ans. : Password protection is the front line protection against intruder to the system. A password authenticate the ID and provides security to the system.

Q.7 Name the authentication protocols. **AU : Dec.-15**

Ans. : Kerberos is an authentication protocol. It provides a way to authenticate clients to services to each other through a trusted third party.

Q.8 List four requirements that were defined for kerberos. **AU : Dec.-15**

Ans. : Requirement of Kerberos: Security, Reliability, Transparency and Scalability.

Q.9 List any four password selection strategies. **AU : Dec.-15**

Ans. :
- In order to eliminate gaussable passwords four basic techniques are suggested.

1. User education
2. Computer generated password
3. Reactive password checking
4. Proactive password checking

Q.10 Specify the various types of authentication protocol. **AU : May-17**

Ans. : Authentication protocols are Mutual vs one-way authentications, symmetric vs public-key approaches, Needham Schroeder protocol.

Q.11 How digital signatures differ from authentication protocols ? AU : May-18

Ans. : Digital signatures provide the ability to verify author, date and time of signature, authenticate message contents and verified by third parties to resolve disputes. Authentication Protocols used to convince parties of each others and identity and to establish session keys.

Q.12 What entities constitute a full-service Kerberos environment ?

Ans. : A full service environment consists of a Kerberos server, a number of clients and a number of application servers.

Q.13 What are the principle differences between Kerberos version 4 and version 5 ? AU : May-11, IT

Ans. :

i) Kerberos V.4 requires DES and V.5 allows many encryption techniques.

ii) V.4 requires use of IP and V.5 allows other network protocols.

iii) Version 5 has a longer ticket lifetime.

iv) Version 5 allows tickets to be renewed.

v) Version 5 can accept any symmetric-key algorithm.

vi) Version 5 uses a different protocol for describing data types.

vii) Version 5 has more overhead than version 4.

Q.14 When are the certificates revoked in X.509 ? AU : May-15

Ans. : The certificate should be revoked before expiry because of following reasons :

1. User's private key is compromised.

2. User is not certified by CA.

3. CA's certificate is compromised.

Q.15 Show how SHA is more secure than MD5 ? AU : May-19

Ans. : SHA is more secure than MD5 due to a variety of reasons. First, it produces a larger digest, 160-bit compared to 128-bit, so a brute force attack would be much more difficult to carry out. Also, no known collisions have been found for SHA.

Q.16 What is realm in Kerberos ? AU : Dec.-19

Ans. : A Kerberos realm is the domain over which a Kerberos authentication server has the authority to authenticate a user, host or service. A realm name is often, but not always the upper case version of the name of the DNS domain over which it presides. The Kerberos server shares a secret key with other Kerberos servers. Therefore, A Kerberos realm is a set of these managed "nodes" that share the same Kerberos database.

Q.17 What entities constitute a full service in Kerberos environment ? AU : Dec.-19

Ans. : A full-service environment consists of a Kerberos server, a number of clients, and a number of application servers.

Q.18 What is key distribution center ?

AU : Dec.-15

Ans. : A key distribution center is responsible for distributing keys to pairs of users such as hosts, processes, applications. Each user must share a unique key with the key distribution center for purposes of key distribution.

Q.19 What are the advantages of key distribution ?

Ans. :

- It is easy to add and remove entities from the network.
- Each entity needs to store only one long-term secret key.
- The public file could reside with each entity.
- Prevent an active adversary from impersonation.

Q.20 What is key management?

Ans. : Key management is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.

Q.21 What is master key?

Ans. : Session keys are transmitted in encrypted form, using a master key that is shared by the key distribution center and an end system or user.

Q.22 Define session key.

Ans. : Communication between end systems is encrypted using a temporary key, often referred to as a session key.

Q.23 What is PKI?

Ans. : A Public-Key Infrastructure (PKI) is defined as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

Q.24 What is key distribution?

Ans. : Key distribution is the function that delivers a key to two parties who wish to exchange secure encrypted data. Some sort of mechanism or protocol is needed to provide for the secure distribution of keys.

Q.25 What is digital certificate?

Ans. : Certificates are digital documents that are used for secure authentication of communicating parties.

Q.26 What is Certification Authority?

Ans. : The trusted party who issues certificates to the identified end entities is called a Certification Authority.

Q.27 What is a nonce?

Ans. : A random value to be repeated in message to assure that the response is fresh and has not been replayed by an opponent.

Q.28 What is ticket-granting server ?

Ans. : A server that issues tickets for a desired service which are in turn given to users for access to the service. The TGS usually runs on the same host as the KDC.

Q.29 Define Kerberos.

Ans. : Kerberos is an authentication protocol. It provides a way to authenticate clients to services to each other through a trusted third party.

Q.30 Define Realm

Ans. : A network that uses Kerberos composed of one or more servers called KDCs and a potentially large number of clients.

Q.31 What is challenge/response ?

Ans. : Party A, expecting a fresh message from B, first sends B a nonce(challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

Q.32 Define Kerberos realm.

Ans. : Kerberos realm is a set of managed nodes that share the same Kerberos database.

❑❑❑