Session 8&9

Symmetric Key Distribution using Symmetric encryption and Asymmetric encryption.

8&9.1 Aim:

To familiarize students with the basic concept of Session Key distribution using Symmetric Encryption scheme and Asymmetric Encryption scheme.

8&9.2 Instructional Objectives:

- Demonstrate the procedure of key distribution using both Symmetric and Asymmetric Encryption
- Describe Purpose of KDC.

8&9.3 Learning Outcomes:

At the end of this session, you should be able to:

- Define Session key, Master Key.
- Describe KDC usage.
- Summarize the Session key Distribution.

8&9.4 Module Description:

This Module covers various procedures for digital certificate and techniques of user authentication. The explanation of each way of distribution of Session Keys, Public Keys and elaboration of various authentication schemes gives more insight to the trust and confidence.

8&9.5 Session Introduction:

In this session we cover distribution of Secret (Symmetric) key using symmetric and Asymmetric encryption.

8&9.6 Session Description:

Key Management and Distribution:
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.

Key management refers to managing cryptographic keys within a cryptosystem. It deals with generating, exchanging, storing, using and

replacing keys as needed at the user level. A key management system will also include key servers, user procedures and protocols, including cryptographic protocol design.
symmetric schemes require both parties to share a common secret key. Public key schemes require parties to acquire valid public keys.
Cryptographic, protocol, and management issues are all involved in the complicated topics of cryptographic key management and distribution.
Symmetric encryption requires usage of common key by both the parties sharing of common key was also problem here. And that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attacker learns the key. This is one of the most critical areas in security systems - on many occasions systems have been broken, not because of a poor encryption algorithm, but because of poor key selection or management.

Below we look into the methods available for the distribution of symmetric key (session key) by using symmetric encryption: Assumption is A and B are the parties who want to have the communication.

1. A can select key and physically deliver to B. (Awkward)

2. Third party can select & deliver key to A & B. (Awkward)

3. If A & B have communicated previously can use previous key to encrypt a new key. (Risk of old key usage)

4. If A & B have secure communications with a third party C, C can relay key between A & B. (recommended usage of variety of it)

The strength of any cryptographic system thus depends on the key distribution technique. For two parties A and B, key distribution can be achieved in several ways:

Physical delivery (1 & 2) is simplest - but only applicable when there is personal contact between recipient and key issuer. This is fine for link encryption where devices & keys occur in pairs but does not scale as number of parties who wish to communicate grows (see next slide). 3 is mostly based on 1 or 2 occurring first, and also suffers that if an attacker ever succeeds in gaining access to one key, then all subsequent keys will be revealed.

A third party, whom all parties trust, can be used as a trusted intermediary to mediate the establishment of secure communications between them (4). Must trust

intermediary not to abuse the knowledge of all session keys. As the number of parties grow, some variant of 4 is only practical solution to the huge growth in number of keys potentially needed.

The below hierarchy of keys shows the relationship between various keys. The main categories are mentioned below.

Master Key

It is used to encrypt session keys.

It is shared by user & key distribution centre.

Session key

It is a temporary key.

It is used for encryption of data between users.

It is for one logical session then discarded.

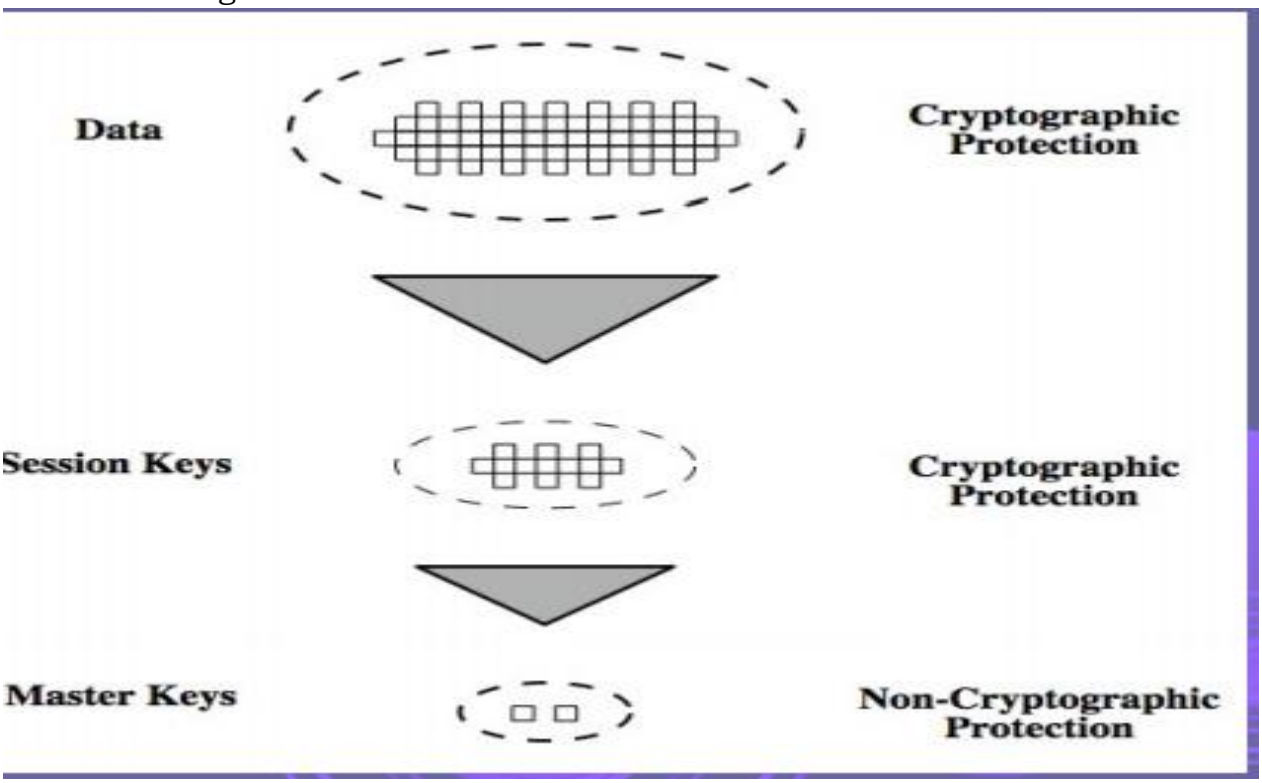

Fig1: Key hierarchy.

Key distribution scenario:

1. A requests from the KDC a session key to protect a logical connection to B. The message includes the identity of A and B and a unique nonce N1.

2. The KDC responds with a message encrypted using Ka that includes a one-time session key Ks to be used for the session, the original request message to enable A to match response with appropriate request, and info for B.

3. A stores the session key for use in the upcoming session and forwards to B the information from the KDC for B, namely, E(Kb, [Ks || IDA]). Because this information is encrypted with Kb, it is protected from eavesdropping.

At this point, a session key has been securely delivered to A and B, and they may begin their protected exchange. Two additional steps are desirable:

4. Using the new session key for encryption B sends a nonce N2 to A.

5. Also using Ks, A responds with f(N2), where f is a function that performs some transformation on N2 (eg. adding one). These steps assure B that the original message it received (step 3) was not a replay. Note that the actual key distribution involves only steps 1 through 3 but that steps 4 and 5, as well as 3, perform an authentication function.

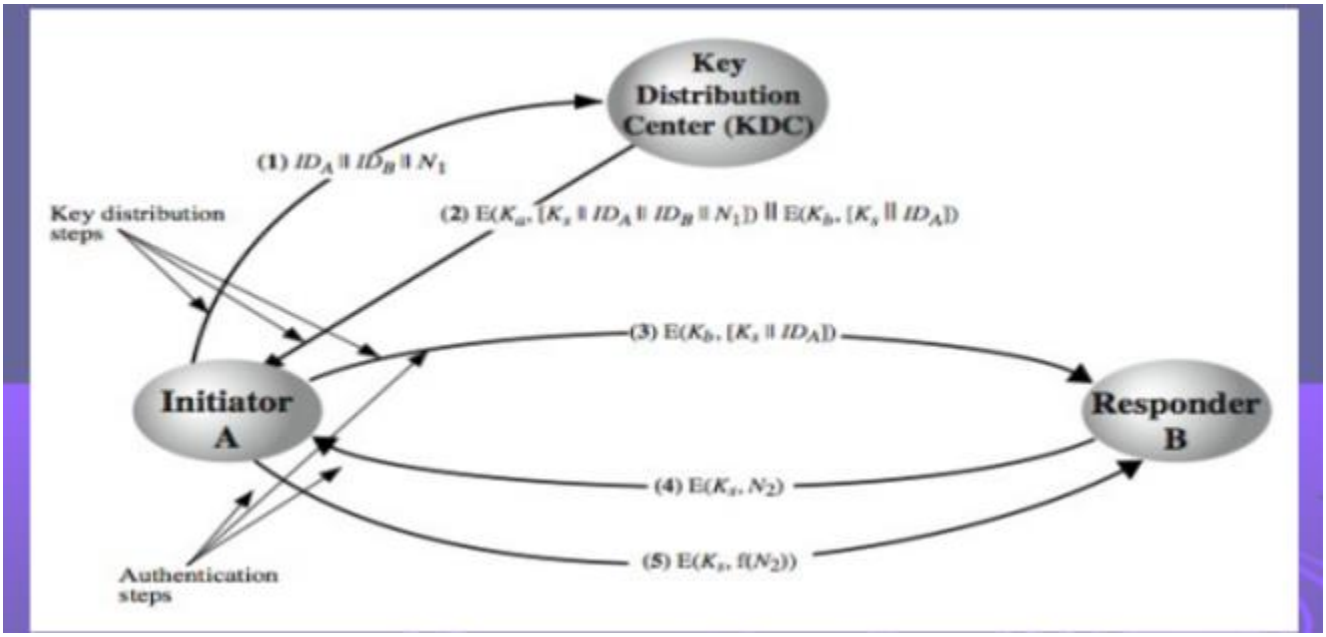

Fig2: Key distribution using KDC.

Key distribution issues:

Some of the major issues associated with the use of Key Distribution Centers (KDC's) are to be noted.

For very large networks, a hierarchy of KDCs can be established. For communication among entities within the same local domain, the local KDC is responsible for key distribution. If two entities in different domains desire a shared key, then the corresponding local KDCs can communicate through a (hierarchy of) global KDC(s)

For each new connection-oriented session, a new session key should be utilized to strike a compromise between security and effort. A new session key for a connectionless protocol is only utilized for a predetermined amount of time or for a predetermined number of transactions.

An automated key distribution approach provides the flexibility and dynamic characteristics needed to allow several terminal users to access several hosts and for the hosts to exchange data with each other, provided they trust the system to act on their behalf.

The use of a key distribution center imposes the requirement that the KDC be trusted and be protected from subversion. This requirement can be avoided if key distribution is fully decentralized.

Symmetric Key Distribution Using Asymmetric Encryption.

Public key and private keys are to be used in Asymmetric encryption. Public key cryptosystems are inefficient so almost never use for direct data encryption, rather use to encrypt secret keys for distribution.

Simple Secret Key Distribution:

Merkle proposed this very simple scheme that allows secure communications and no keys before/after exist.

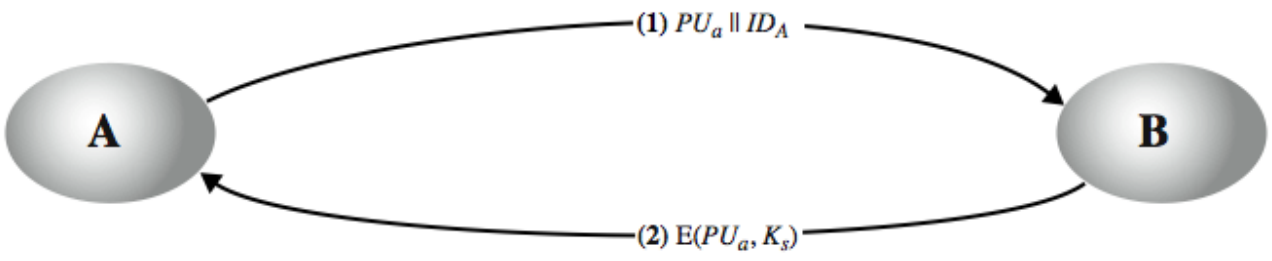

Fig3: Simple Key distribution using Public Keys.

Man-in-the-Middle Attack:

This very simple scheme is vulnerable to an active man-in-the-middle attack.

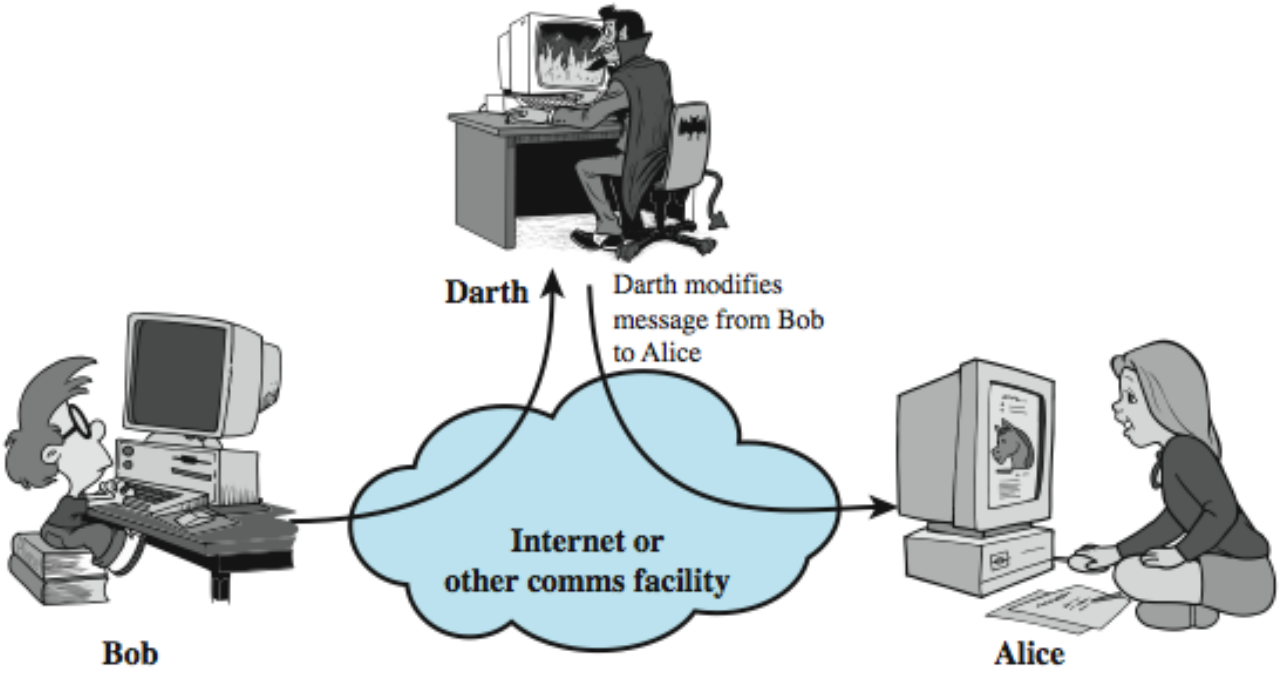

Fig4: Man in the middle attack.

The following steps occur to ensure both confidentiality and authentication:

A uses B's public key to encrypt a message to B containing an identifier of A (IA) and a nonce (N1), which is used to identify this transaction uniquely.

B sends a message to A encrypted with PUa and containing A's nonce (N1) as well as a new nonce generated by B (N2). Because only B could have decrypted message (1), the presence of N1 in message (2) assures A that the correspondent is B.

A returns N2, encrypted using B's public key, to assure B that its correspondent is A.

A selects a secret key Ks and sends M = E(PUb, E(PRa, Ks)) to B. Encryption with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.

B computes D(PUa, D(PRb, M)) to recover the secret key.

The result is that this scheme ensures both confidentiality and authentication in the exchange of a secret key.

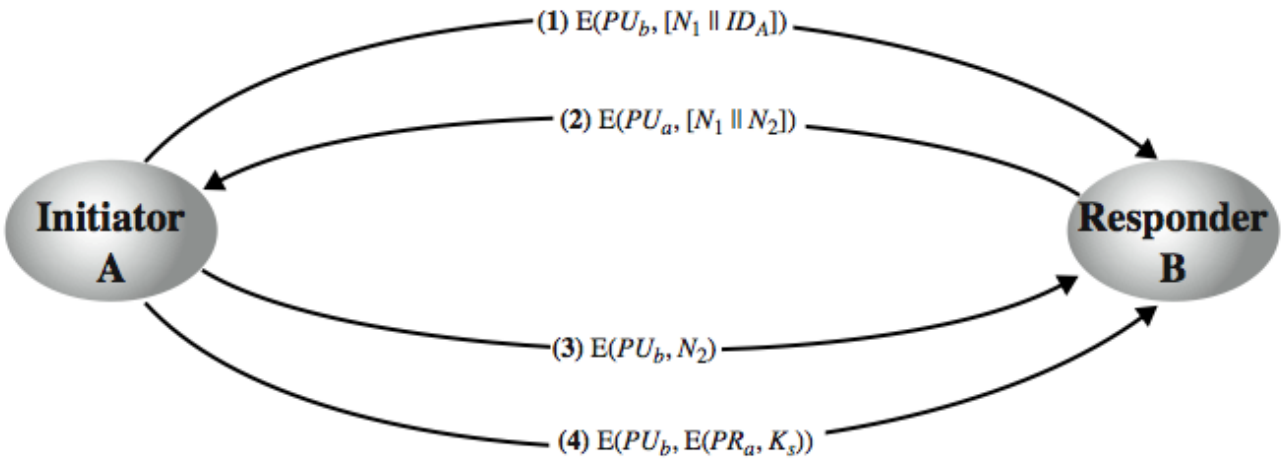

Fig5: Key distribution using Public Encryption.

8&9.7 Important facts related to the session:

Key Distribution in Symmetric Encryption:

Symmetric encryption uses a single shared secret key for both encryption and decryption.
The key must be securely distributed to all parties involved in the communication.
One common method of key distribution is using a trusted courier to physically deliver the key to each party.
Another approach is to establish the key in person before communication begins.
Secure key distribution is crucial because if an unauthorized party obtains the key, they can decrypt the messages.

Key Distribution in Asymmetric Encryption:
Asymmetric encryption involves a pair of mathematically related keys: a public key and a private key.
The public key is freely distributed to anyone who wants to communicate with the owner, while the private key is kept secret.
With asymmetric encryption, the public key is used for encryption, while the private key is used for decryption.
Since the public key can be openly distributed, there is no need for a secure key distribution mechanism.

Asymmetric encryption is often used to securely exchange a symmetric key. Parties can encrypt a symmetric key with the recipient's public key and send it over an insecure channel.
Once the encrypted symmetric key is received, the recipient can use their private key to decrypt it and obtain the symmetric key for further communication.

8&9.8 Examples and contemporary extracts of articles or practices to convey the idea of the session:

NA

8&9.9 Table Numbering:

NA

8&9.10 Figures with Captions:

Fig1: Key hierarchy.
Fig2: Key distribution using KDC.
Fig3: Simple Key distribution using Public Keys.
Fig4: Man in the middle attack.
Fig5: Key distribution using Public Encryption.

8&9.11 Self-Assessment Questions:

1. Which of the following is a method for distributing secret keys using asymmetric encryption?
   (A) Key exchange (B) Key transport (C) Key management (D) Key agreement

2. Mention Basic Symmetric Encryption Algorithm.

3. Which of the following is a method for distributing secret keys using symmetric encryption?
   (A) Key exchange (B) Key transport (C) Key management (D) Key agreement

4. Which of the following is the most common method for distributing secret keys using asymmetric encryption?
   A. Web of trust
   B. Public key directories
   C. Certificate authorities
   D. Direct exchange

8&9.12 Summary:

In this session the clear explanation of distribution of Secret(Session) Key between the parties

8&9.13 Terminal Questions:

1. Describe KDC.
2. List out Keys in keys Hierarchy.
3. Analyze How many keys are required if an organization with n users want to communicate securely using symmetric keys

8&9.14 Case Study: NA

8&9.15 Answer Key:

1) A) Key Exchange.
2) DES.
3) B) Key Transport.
4) D)Direct Exchange.

8&9.16 Glossary:

KDC: Key Distribution Centre.

8&9.17 References:

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 7th edition, 2017. 2

2. Cryptography And Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Tata McGraw Hill Education Private Limited, Fourth edition 2015.

3. William Stallings, “Network Security Essentials”, Pearson Education, 7th Edition, 2017.

8&9.18 Keywords:

Key, Encryption, Public Key, Private Key, KDC.

****

Team- Network and Infrastructure Security.

Session 10

Distribution of Public Keys & PKI

10.1 Aim:

To familiarize students with various ways of the distribution of public keys and Public Key Infrastructure (PKI).

10.2 Instructional Objectives:

This Session is designed to:

- Demonstrate the various public key Distribution schemes.
- Demonstrate the Public Key Infrastructure.

10.3 Learning Outcomes:

At the end of this session, you should be able to:

1. Publicly available Directory.
2. Public key certificates.
3. Public key Authority.
4. Certificate Authority.
5. Public Key Infrastructure.

10.4 Module Description:

This Module covers various procedures for digital certificate and techniques of user authentication. The explanation of each way of distribution of Session Keys, Public Keys and elaboration of various authentication schemes gives more insight to the trust and confidence.

10.5 Session Introduction:

In this we cover the available ways for distribution of the Public Keys and Public Key Infrastructure.

10.6 Session Description:

Several techniques are proposed for the distribution of public keys, those can mostly be grouped into the categories mentioned below.

- Public announcement
- Publicly available directory
- Public-key authority
- Public-key certificates

Public announcement:

Users distribute public keys to recipients or broadcast to community at large.
E.g. Append PGP keys to email messages or post to news groups or email list.

> Major weakness is forgery; anyone can create a key claiming to be someone else and broadcast it until forgery is discovered can masquerade as claimed user

The point of public-key encryption is that the public key is public, hence any participant can send his or her public key to any other participant or broadcast the key to the community at large. Its major weakness is forgery, anyone can create a key claiming to be someone else and broadcast it, and until the forgery is discovered they can masquerade as the claimed user.

Publicly Available Directory:

A higher level of security can be attained by keeping a dynamic directory of public keys that is accessible to the public. A dependable body or organisation would have to be in-charge of maintaining and disseminating the public directory. Although this method is undoubtedly more secure than making individual public statements, it is still susceptible to fraud or sabotage.

- can obtain greater security by registering keys with a public directory.
- directory must be trusted with properties:
  - contains {name, public-key} entries for each participant.
  - participants register securely with directory.
  - participants can replace key at any time.
  - directory is periodically published.
  - directory can be accessed electronically.
- still vulnerable to tampering or forgery.

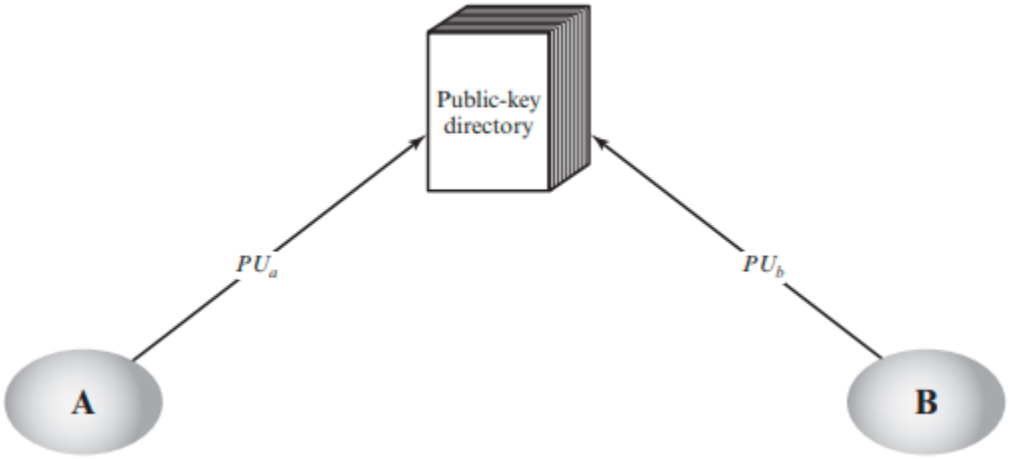

Figure 1: usage of public key directories by parties A and B.

Public key Authority

"Public-Key Authority" serves as an example of a common protocol exchange. The scenario continues to presume that a central authority has a dynamic directory of all participants' public keys. Additionally, only the authority is aware of the associated private key, whereas each participant reliably knows the authority's public key. The steps of the protocol are detailed in the text. There are a total of seven messages needed. However, because both A and B can cache each other's public keys for later use, the first four messages only need to be sent occasionally.

To maintain currency, a user should periodically request new copies of the public keys of its correspondents.

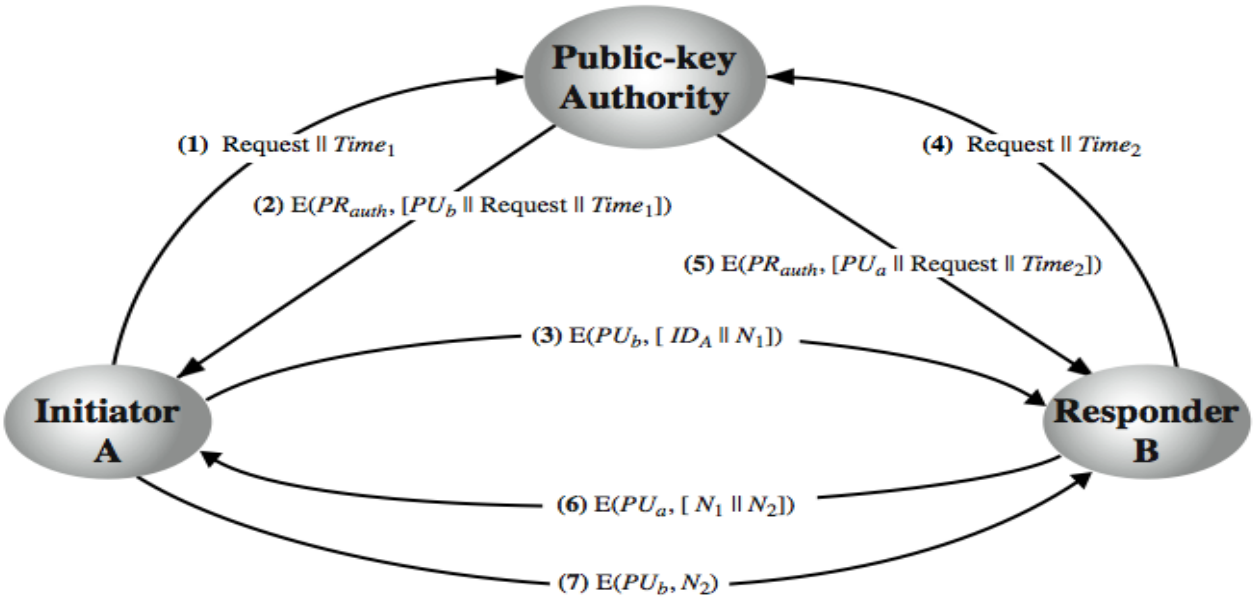

Figure 2: usage of public key authority.

- Stronger security can be achieved by tightening control over distribution of keys from directory.
- has properties of directory and requires users to know public key for the directory
- then users interact with directory to obtain any desired public key securely
    - does require real-time access to directory when keys are needed
    - may be vulnerable to tampering

Public key Authority bottleneck

The public-key authority could be somewhat of 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 tampering.

Public-Key Certificates

Each participant submits an application to the certificate authority with a public key and a certificate request. The application process must be done in person or over a secure, secured communication method. The authority gives participant A a certificate CA. Any other participant may then receive this certificate from A, who may then read it and verify its authenticity by confirming the signature of the certificate authority. This confirms that the certificate originated from the certificate authority because it can only be read with the authority's public key. The following

scenario is counted by the timestamp. An opponent discovers A's secret key. A creates a fresh private/public key pair and requests a new certificate from the certificate authority. The adversary is in the process of playing B the old certificate. The attacker can access messages that B encrypts using the old public key that has been compromised. In this situation, losing a credit card is analogous to losing a private key. The owner cancels the credit card number, but until all potential communicators are informed that the previous credit card is no longer valid, they are at risk. As a result, the timestamp functions as a kind of expiration time. A certificate is considered to have expired if it is old enough.

- certificates allow key exchange without real-time access to public-key authority.
- a certificate binds **identity** to **public key**
  >usually with other info such as period of validity, rights of use etc
- with all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key

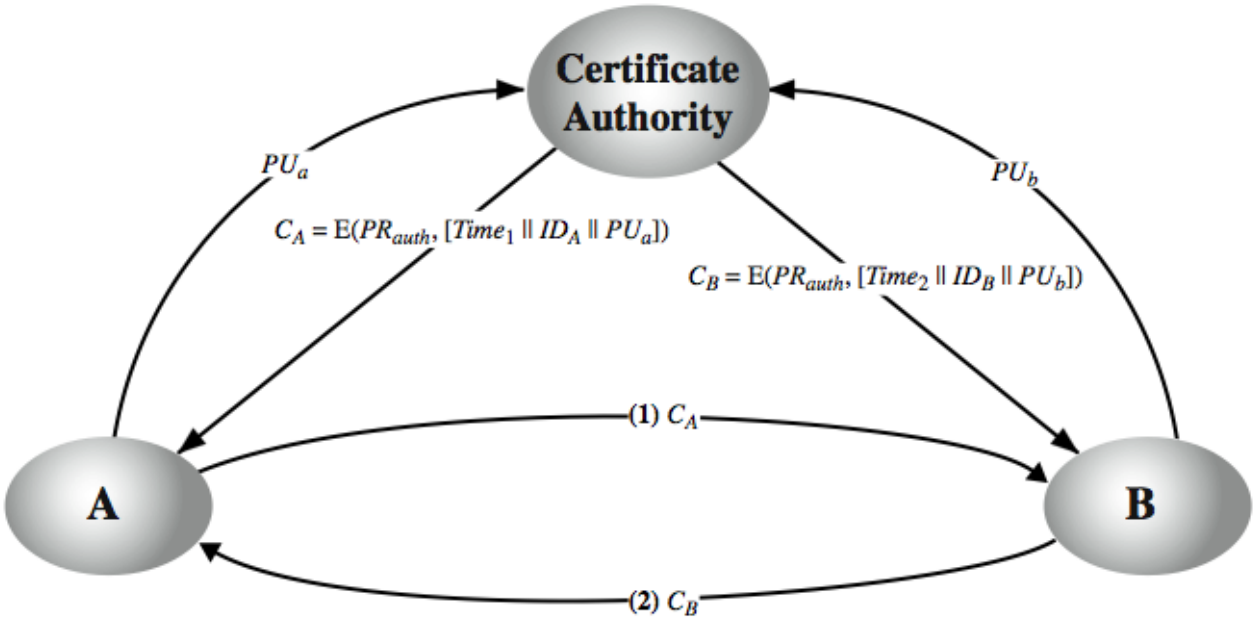

Figure 3: usage of certificate authority.

Public Key Infrastructure

RFC 4949 (Internet Security Glossary) defines public-key infrastructure (PKI) as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography.

The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) working group has been the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a certificate-based architecture on the Internet.

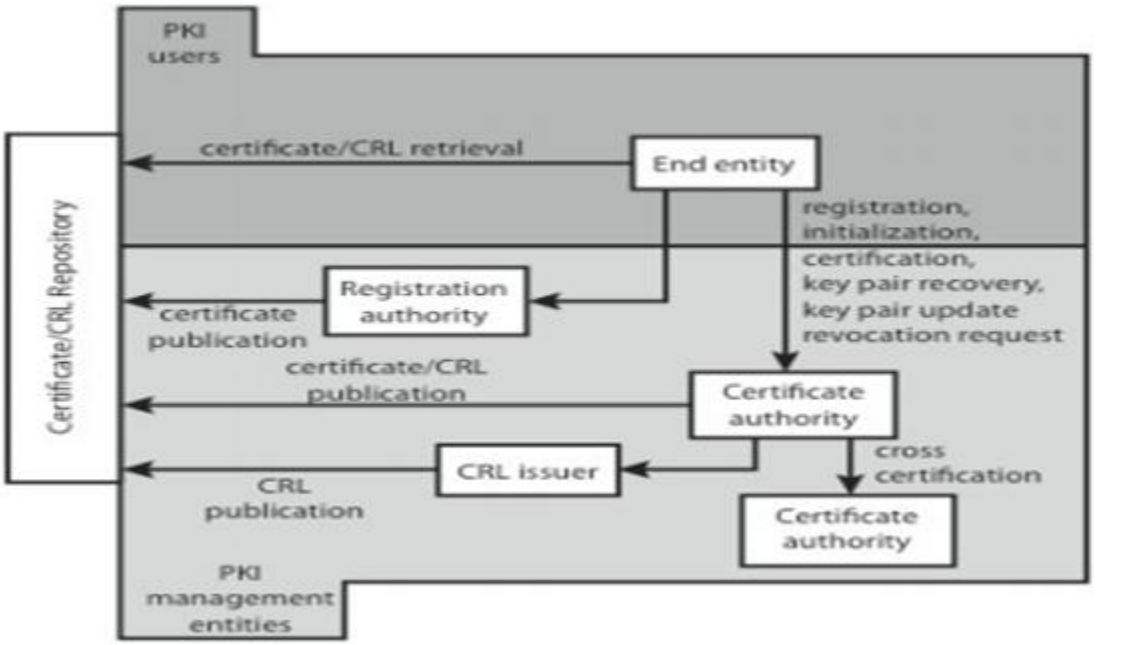

Figure 4: Public Key Infrastructure.

The main elements are:

1. End entity
2. Certification authority (CA)
3. Registration authority (RA)
4. CRL issuer
5. Repository

PKIX identifies several management functions that potentially need to be supported by management protocols, as shown in Figure 14.16:

• **Registration:** whereby a user first makes itself known to a CA, prior to issue of a certificate(s) for that user. It usually involves some off-line or online procedure for mutual authentication.

• **Initialization:** to install key materials that have the appropriate relationship with keys stored elsewhere in the infrastructure.

• **Certification:** process where a CA issues a certificate for a user's public key and returns it to the user's client system and/or posts it in a repository.

• **Key pair recovery:** a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible.

• **Key pair update:** key pairs need to be updated and new certificates issued.
• **Revocation request:** when authorized person advises need for certificate revocation, e.g. private key compromise, affiliation change, name change.
• **Cross certification:** when two CAs exchange information used in establishing a cross-certificate, issued by one CA to another CA that contains a CA signature key used for issuing certificates.

10.7 Important facts related to the session:

Public key distribution schemes leverage trusted third parties or a web of trust to securely distribute public keys. PKI is a comprehensive framework that supports the management, distribution, and revocation of digital certificates, enabling secure communication and authentication in various applications.

10.8 Examples and contemporary extracts of articles or practices to convey the idea of the session:

NA.

10.9 Table Numbering: NA

10.10 Figures with Captions:

Figure 1: usage of public key directories by parties A and B.
Figure 2: usage of public key authority.
Figure 3: usage of certificate authority.
Figure 4: Public Key Infrastructure.

10.11 Self-Assessment Questions:

1. What is the purpose of distributing public keys?
2. Which method of distributing public keys is the most secure?
3. What are some of the risks associated with distributing public keys?

10.12 Summary:

The session has explained all the ways available for the distribution of public keys. And PKI also has been explained clearly.

10.13 Terminal Questions:

1. Describe Public announcement.
2. Mention about the content of Public available Directory.
3. Mention the purpose of Certificate authority.
4. Define process of caching.

10.14 Case Study:

NA

10.15 Answer Key:

1) The purpose(s) of distributing the public keys:

- To allow users to verify the authenticity of messages and signatures.
- To allow users to encrypt messages.
- To allow users to authenticate each other.

2) Among all Direct exchange is the most secure method of distributing public keys because it does not rely on a third party.

3) The main risks associated with distribution of public keys are:

- Public keys can be intercepted and used to impersonate users.
- Public keys can be compromised and used to decrypt messages or forge signatures.
- Public keys can be used to track users' activity.

10.16 Glossary:

PKI - Public Key Infrastructure, PKA – Public Key Authority, CRL – Certificate Revocation List, CA- Certification Authority & RA – Registration Authority

10.17 References:

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 7th edition, 2017. 2

2. Cryptography And Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Tata McGraw Hill Education Private Limited, Fourth edition 2015.

3. William Stallings, “Network Security Essentials”, Pearson Education, 7th Edition, 2017.

10.18 Keywords:

Public Key, Public Key Infrastructure, Public Key Authority, Certificate Revocation List, Certification Authority & Registration Authority.

****

Team- Network and Infrastructure Security.

Session 11

X.509 Certificates

11.1 Aim:

To familiarize students with various ways of the X.509 Certificate, its purpose, obtaining of it.

11.2 Instructional Objectives:

This Session is designed to:

- Demonstrate the Purpose of X.509 Certificate.
- Demonstrate the Usage of X.509 Certificate.

11.3 Learning Outcomes:

At the end of this session, you should be able to:

- X.509 Certificate and it's use.
- Elements of X.509 Certificate.
- Format of X.509 Certificate.
- Obtaining of X.509 Certificate.
- Certificate Revocation.

11.4 Module Description:

This Module covers various procedures for digital certificate and techniques of user authentication. The explanation of each way of distribution of Session Keys, Public Keys and elaboration of various authentication schemes gives more insight to the trust and confidence.

11.5 Session Introduction:

In this we cover the format of X.509 Certificates of different versions and purpose and use of X.509 Certificate.

11.6 Session Description:

Introduction:

- Recommendation of ITU-D and series of X.500.
- X.509 defines a framework for the provision of authentication services by the X.500 directory to its users.
- Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

- X.509 defines alternative authentication protocols based on the use of public-key certificates.
- X.509 certificate format is used in S/MIME.
- X.509 is based on the use of public-key cryptography and digital signatures.

X.509 Public-Key Certificate Use:

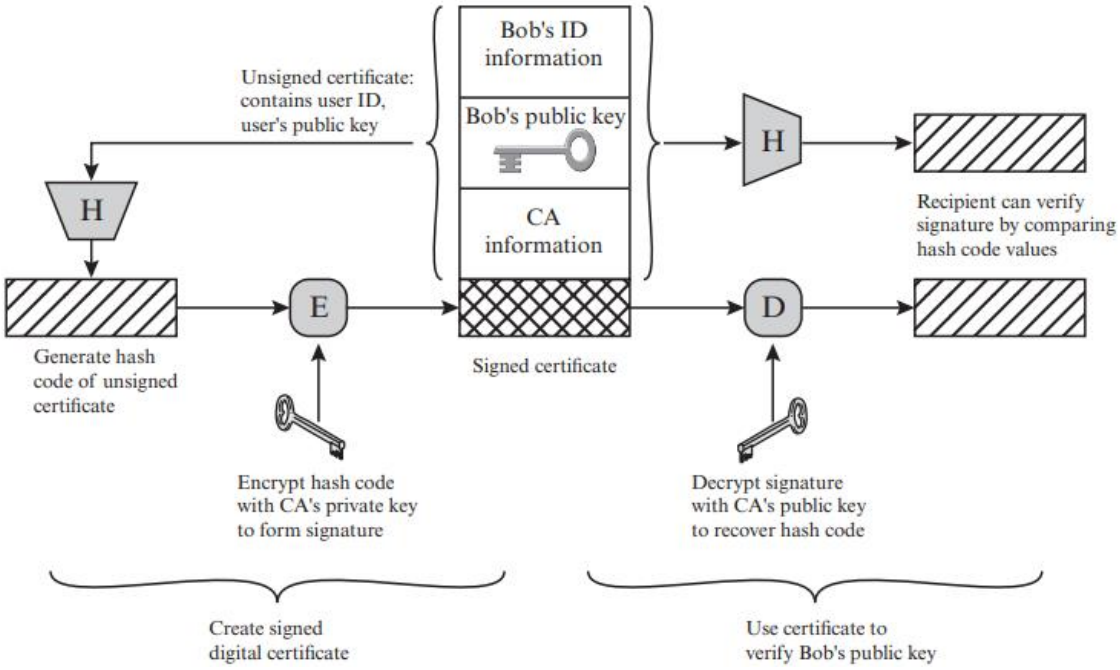

Figure1: Usage of Public key certificate.

The certificate for Bob's public key includes unique identifying information for Bob, Bob's public key, and identifying information about the CA, plus other information as explained subsequently. This information is then signed by computing a hash value of the information and generating a digital signature using the hash value and the CA's private key.

The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user.

X.509 Certificate Format:

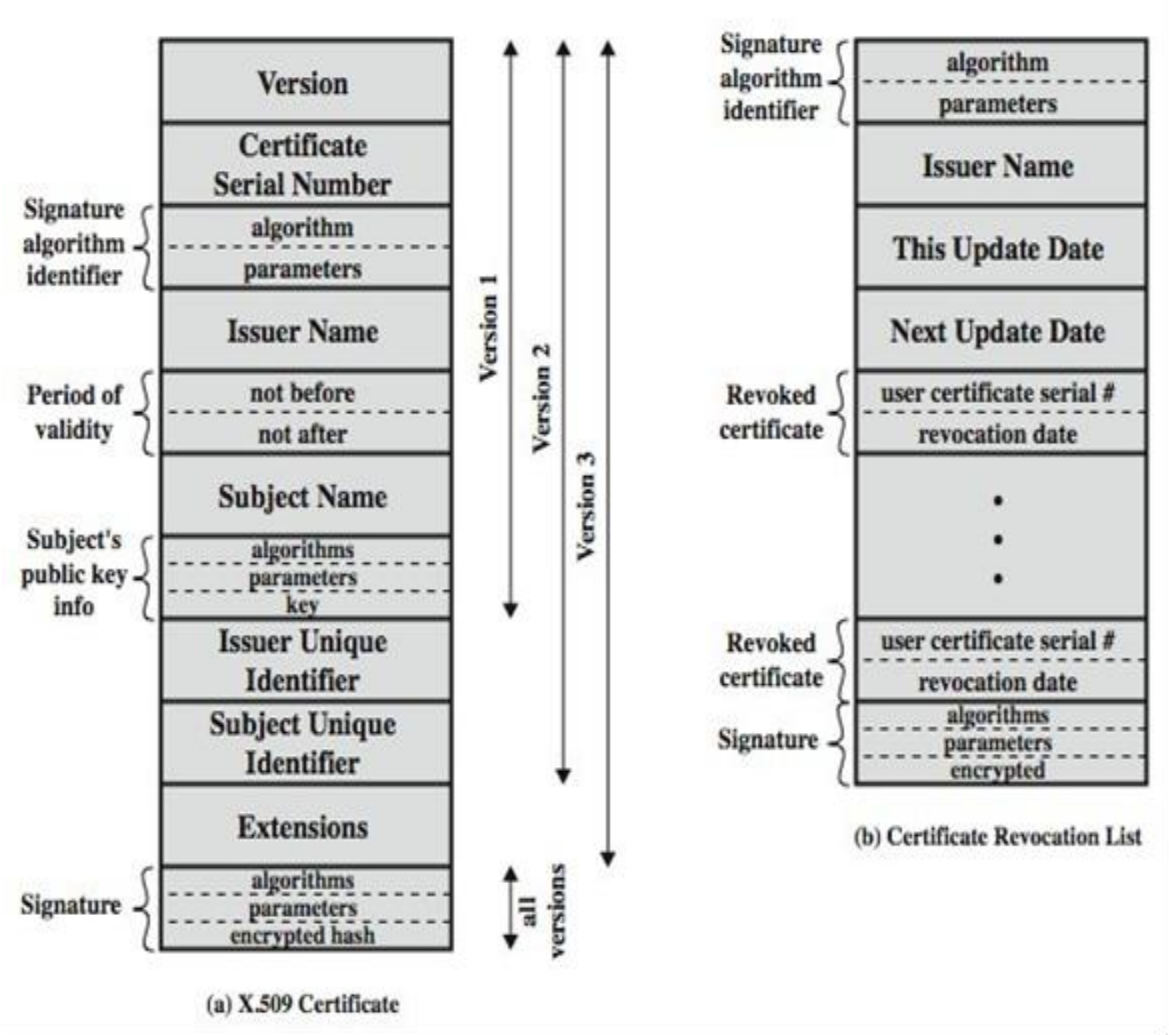

Figure2: X.509 Certificate format with different versions

Obtaining a certificate:

User certificates generated by a CA have the characteristics that any user with access to the public key of the CA can verify the user public key that was certified, and no party other than the certification authority can modify the certificate without this being detected. Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them.

CA Hierarchy

If both parties use the same CA, they know its public key and can verify other's certificates. If there is a large community of users, it may not be practical for all users to subscribe to the same CA. With many users, it may be more practical for there to be a number of CAs, each of which securely provides its public key to some fraction of the users. Hence there must be some means to form a chain of certifications between the CA's used by the two parties, by the use of client and parent certificates. All these certificates of CAs by CAs need to appear in the directory, and the user needs to know how they are linked to follow a path to another user's public-key certificate. X.509 suggests that CAs be arranged in a hierarchy so that navigation is straightforward. It is assumed that each client trusts its parent's certificates.

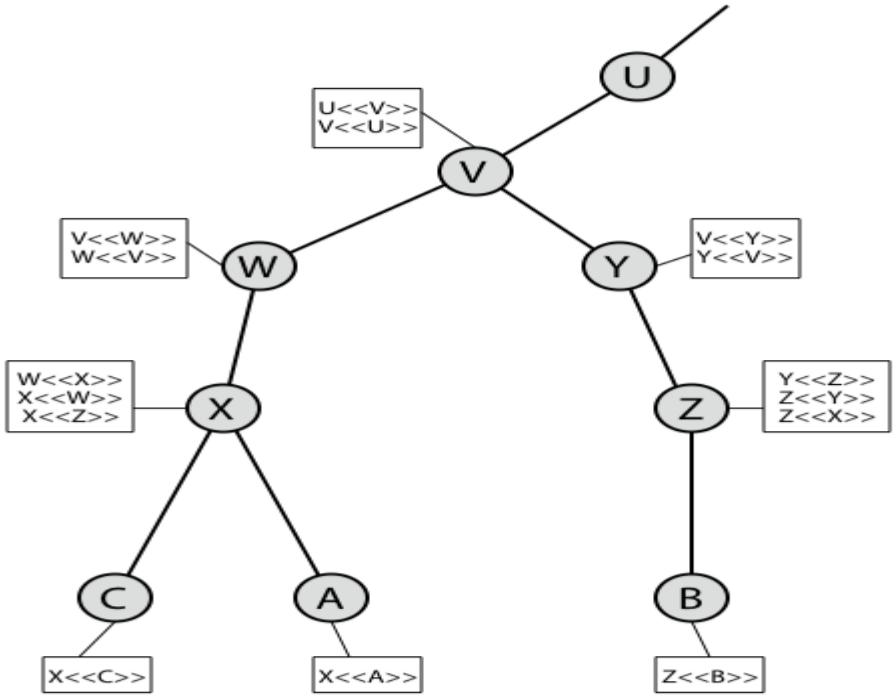

Figure3: CA Hierarchy

A certificate includes a period of validity. Typically, a new certificate is issued just before the expiration of the old one.

Sometimes it may be desirable on occasion to revoke a certificate before it expires, for one of a range of reasons, such as those shown above.

To support this, each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA, known as the certificate revocation list (CRL). Each certificate revocation list (CRL) posted to the directory is signed by the issuer and includes the issuer's name, the date the list was created, the date the next CRL is scheduled to be issued, and an entry for each revoked certificate. Each entry consists of the serial number of a certificate and revocation date for that certificate. Because serial numbers are unique within a CA, the serial number is sufficient to identify the certificate.

When a user receives a certificate in a message, the user must determine whether the certificate has been revoked, by checking the directory CRL each time a certificate is received, this often does not happen in practice.

Key and Policy Information:

A certificate policy is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.

It includes:

1. Authority key identifier.
2. Subject key identifier.
3. Key usage.
4. Private-key usage period.
5. Certificate policies.
6. Policy mappings.

11.7 Important facts:

X.509 certificates provide a standardized format for digital certificates used in PKI systems. They contain identity information, a public key, and other relevant details, allowing for secure authentication, encryption, and verification of digital communications.

11.8 Example:

NA.

11.9 Table Numbering:

NA

11.10 Figures with Captions:

Figure1: Usage of Public key certificate.
Figure 2: X.509 Certificate format with different versions
Figure 3: CA Hierarchy.

11.11 Self-Assessment Questions:

1. What is an X.509 certificate?
2. How are X.509 certificates used?
3. How key compromise risk associated with X.509 certificates show the effect?
4. What is certificate revocation?
5. What is certificate pinning?

11.12 Summary:

The session has explained each field in the format of X.509 Certificate and included the details of different versions. In detail explanation of CA Hierarchy is also given.

11.13 Terminal Questions:

1. Describe Certificate.
2. What is X.509.
3. Define Certification revocation.

11.14 Case Study:

NA

11.15 Answer Key:

1) An X.509 certificate is an electronic document that binds a public key to an identity, such as a person, organization, or server.

2) X.509 certificates are used for a variety of purposes, including:

- Secure email: X.509 certificates can be used to encrypt emails and verify the identity of the sender.
- Secure websites: X.509 certificates can be used to secure websites by verifying the identity of the website owner.
- Digital signatures: X.509 certificates can be used to create digital signatures, which can be used to verify the authenticity of a document or message.

3) If the private key associated with an X.509 certificate is compromised, then an attacker could use the key to impersonate the certificate holder.

4) Certificate revocation is the process of invalidating a previously issued X.509 certificate before its expiration date. This is done when a certificate has been compromised, the private key is lost or suspected to be compromised, or the certificate's information needs to be updated.

5) It involves associating a specific X.509 certificate or its public key with a particular server or service. By "pinning" the certificate or public key, clients can detect and prevent attacks that use rogue or compromised certificates, providing an extra layer of trust and security.

11.16 Glossary:

PKI - Public Key Infrastructure, PKA – Public Key Authority, CRL – Certificate Revocation List, CA- Certification Authority & RA – Registration Authority

11.17 References:

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 7th edition, 2017. 2

2. Cryptography And Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Tata McGraw Hill Education Private Limited, Fourth edition 2015.

3. William Stallings, "Network Security Essentials", Pearson Education, 7th Edition, 2017.

11.18 Keywords:

Pubic Key, Public Key Infrastructure, Public Key Authority, Certificate Revocation List, Certification Authority & Registration Authority.

****

Team- Network and Infrastructure Security.

Session 12

User Authentication using Symmetric Encryption

12.1 Aim:

To familiarize students with the way of authenticating user using symmetric encryption.

12.2 Instructional Objectives:

This Session is designed to demonstrate the Authentication of user using symmetric encryption using popular protocols.

12.3 Learning Outcomes:

At the end of this session, you should be able to understand:

- Identification.
- Authentication.
- Mutual authentication.
- One way authentication.
- Difference between Needham-Schroeder Protocol and Denning protocol.

12.4 Module Description:

This Module covers various procedures for digital certificate and techniques of user authentication. The explanation of each way of distribution of Session Keys, Public Keys and elaboration of various authentication schemes gives more insight to the trust and confidence.

12.5 Session Introduction:

User authentication is the fundamental building block and the basis for many types of access control. RFC 4949 (Internet Security Glossary) defines user authentication as the process of verifying an identity claimed by or for a system entity.

The user authentication process is consisting of two steps:

Identification step: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)

Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier. For example, user Alice Toklas could have the user identifier ABTOKLAS. This information needs to be stored on any server or computer system that Alice wishes to use and could be known to system administrators and other users.

A typical item of authentication information associated with this user ID is a password, which is kept secret (known only to Alice and to the system). If no one can obtain or guess Alice's password, then the combination of Alice's user ID and password enables administrators to set up Alice's access permissions and audit her activity. Because Alice's ID is not secret, system users can send her email, but because her password is secret, no one can pretend to be Alice.

General Means of Authenticating User Identity.

Four means of authenticating user's identity based one something the individual

- Something the individual knows - e.g. password, PIN
- Something the individual possesses - e.g. key, token, smartcard
- Something the individual is (static biometrics) - e.g. fingerprint, retina
- Something the individual does (dynamic biometrics) - e.g. voice, sign

12.6 Session Description:

Symmetric encryption is a type of encryption where the same key is used to encrypt and decrypt data. This makes it relatively easy to implement, but it also means that the key must be kept secret.

User authentication is the process of verifying the identity of a user. This is typically done by requiring the user to provide something they know (such as a password), something they have (such as a security token), or something they are (such as a fingerprint).

To use symmetric encryption for user authentication, the client and server must share a secret key. The client encrypts a challenge message using the secret key and sends it to the server. The server decrypts the challenge message and uses it to verify the client's identity.

Here is an example of how symmetric encryption can be used for user authentication:

The client sends a request to the server for a service.

The server generates a random challenge message and encrypts it with the shared secret key.

The server sends the encrypted challenge message back to the client.

The client decrypts the challenge message using the shared secret key.

The client sends the decrypted challenge message back to the server.

The server compares the decrypted challenge message to the original challenge message. If they match, then the client is authenticated.

Symmetric encryption can be a secure way to authenticate users, but it is important to keep the shared secret key secret. If the key is compromised, then an attacker could use it to impersonate users and gain unauthorized access to services.

Here are some additional security considerations for using symmetric encryption for user authentication:

The shared secret key must be strong and unique. It should be generated using a secure random number generator and should not be reused for other purposes.

The shared secret key must be kept secret. It should not be stored in cleartext on any system.

The shared secret key should be rotated periodically. This will help to mitigate the risk of compromise.

By following these security considerations, you can help to ensure that symmetric encryption is used securely for user authentication.

This strategy involves the use of a trusted key distribution center (KDC).

Each party in the network shares a secret key, known as a master key, with the KDC.

The KDC is responsible for generating keys to be used for a short time over a connection between two parties, known as session keys, and for distributing those keys using the master keys to protect the distribution.

One-Way Authentication

- The email message is forwarded to the receiver's electronic mailbox, where it is buffered until the receiver is available to read it.
- A second requirement is that of authentication. Typically, the recipient wants some assurance that the message is from the alleged sender.

Needham-Schroeder Protocol:

original third-party key distribution protocol
for session between A B mediated by KDC
protocol overview is:

1. A->KDC: IDA || IDB || N1

2. KDC -> A: E(Ka,[Ks||IDB||N1|| E(Kb,[Ks||IDA])])

3. A -> B: E(Kb, [Ks||IDA])

4. B -> A: E(Ks, [N2])

5. A -> B: E(Ks, [f(N2)])

- Used to securely distribute a new session key for communications between A & B
- But is vulnerable to a replay attack if an old session key has been compromised.
- then message 3 can be resent convincing B that is communicating with A
- Modifications to address this disadvantage was proposed by Denning by considering the following:
  - timestamps in steps 2 & 3 (Denning 81)
  - using an extra nonce (Neuman 93)

Denning Approach

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A: \quad E(K_s, N_1)$
5. $A \rightarrow B: \quad E(K_s, f(N_1))$

- T is a timestamp that assures A and B that the session key has only just been generated.
- Thus, both A and B know that the key distribution is a fresh exchange.
- A and B can verify timeliness by checking that:

use refinement of KDC to secure email

since B no online, drop steps 4 & 5

protocol becomes:

1. A->KDC: IDA || IDB || N1

2. KDC -> A: E(Ka, [Ks||IDB||N1 || E(Kb,[Ks||IDA])])

3. A -> B: E(Kb, [Ks||IDA]) || E(Ks, M)

provides encryption & some authentication.

does not protect from replay attack.

12.7 Important facts:

It's important to note that while symmetric encryption can be used for user authentication, it may have limitations compared to asymmetric encryption methods such as public-key cryptography, which provide additional security features like key distribution without the need for a shared secret key.

12.8 Example:

N/A

12.9 Table Numbering:

NA

12.10 Figures with Captions:

NA

12.11 Self-Assessment Questions:

1. What is the most common way to authenticate users using symmetric encryption?
2. Mention one of potential security risk of using symmetric encryption for user authentication?
3. What is the main purpose of using symmetric encryption for user authentication?

12.12 Summary:

The session has given clear picture of user authentication using symmetric encryption.

12.13 Terminal Questions:

1. What is Authentication?
2. Difference between one way authentication and mutual authentication
3. What is replay attack?

12.14 Case Study:

NA

12.15 Answer Key:

1) Password-based authentication is the most common way to authenticate users using symmetric encryption. The user enters their password, which is then encrypted using a symmetric key. The server then decrypts the password using the same symmetric key to verify that it is correct.

2) The main potential risks are:

A. The symmetric key could be intercepted by an attacker.

B. The symmetric key could be guessed by an attacker.

C. The symmetric key could be reused for multiple users.

3) To prevent unauthorized access is the main purpose.

12.16 Glossary:

- KDC – Key Distribution Center,
- ID – Identifier.
- N – Nonce.
- K – Key.

12.17 References:

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 7th edition, 2017. 2

2. Cryptography And Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Tata McGraw Hill Education Private Limited, Fourth edition 2015.

3. William Stallings, “Network Security Essentials”, Pearson Education, 7th Edition, 2017.

12.18 Keywords:

Authentication, KDC, User, Symmetric encryption.

****

Team- Network and Infrastructure Security.

Session 13

User authentication using Asymmetric encryption.

13.1 Aim:

To familiarize students with the way of authenticating user using Asymmetric encryption

13.2 Instructional Objectives:

This Session is designed to demonstrate the Authentication of user using Asymmetric encryption using popular protocols.

13.3 Learning Outcomes:

At the end of this session, you should be able to understand:

1. Identification.
2. Authentication.
3. Mutual authentication.
4. One way authentication.
5. Difference between different protocols used for this Asymmetric encryption.

13.4 Module Description:

This Module covers various procedures for digital certificate and techniques of user authentication. The explanation of each way of distribution of Session Keys, Public Keys and elaboration of various authentication schemes gives more insight to the trust and confidence.

13.5 Session Introduction:

User authentication is the fundamental building block and the basis for many types of access control. RFC 4949 (Internet Security Glossary) defines user authentication as the process of verifying an identity claimed by or for a system entity.

The user authentication process is consisting of two steps:

Identification step: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)

Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier. For example, user Alice Toklas could have the user identifier ABTOKLAS. This information needs to be stored on any server or computer system that Alice wishes to use and could be known to system administrators and other users.

A typical item of authentication information associated with this user ID is a password, which is kept secret (known only to Alice and to the system). If no one can obtain or guess Alice's password, then the combination of Alice's user ID and password enables administrators to set up Alice's access permissions and audit her activity. Because Alice's ID is not secret, system users can send her email, but because her password is secret, no one can pretend to be Alice.

General Means of Authenticating User Identity.

Four means of authenticating user's identity based one something the individual

- Something the individual knows - e.g. password, PIN
- Something the individual possesses - e.g. key, token, smartcard
- Something the individual is (static biometrics) - e.g. fingerprint, retina
- Something the individual does (dynamic biometrics) - e.g. voice, sign

13.6 Session Description:

User authentication is the process of verifying the identity of a user. Asymmetric encryption can be used for user authentication by having the user's public key stored on a server. When the user wants to authenticate themselves, they send a message encrypted with their private key to the server. The server can then decrypt the message using the user's public key, which verifies the user's identity.

Two keys are used in asymmetric encryption: a public key and a private key. The private key is kept a secret, whereas the public key can be shared with everyone.

There are two main methods of user authentication using asymmetric encryption:

- Mutual authentication: Both the user and the server authenticate each other.
- One-way authentication: Only the user authenticates themselves to the server.

Mutual authentication using denning protocol:

1. $A \rightarrow AS: \quad ID_A \| ID_B$
2. $AS \rightarrow A: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$
3. $A \rightarrow B: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

The central system is referred to as an authentication server (AS), because it is not actually responsible for secret-key distribution. Rather, the AS provides public-key certificates. The session key is chosen and encrypted by A; hence, there is no risk of exposure by the AS. The timestamps protect against replays of compromised keys.

Another protocol proposed by woo and lam :

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_B)]) \| N_b])$
7. $A \rightarrow B: \quad E(K_s, N_b)$

Revised version of above protocol is:

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [N_b \| E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B])])$
7. $A \rightarrow B: \quad E(K_s, N_b)$

Disadvantages of the Existing Protocols:

Approaches require that either the sender know the recipient's public key 502 CHAPTER 15 / USER AUTHENTICATION (confidentiality), the recipient

know the sender's public key (authentication), or both (confidentiality plus authentication).

In addition, the public-key algorithm must be applied once or twice to what may be a long message.

13.7 Activities/ Case studies/ Important facts related to the session:

Here are some of the benefits of using asymmetric encryption for user authentication:

- It is very secure.
- It is relatively easy to implement.
- It can be used to authenticate users over insecure networks.

Here are some of the challenges of using asymmetric encryption for user authentication:

- It can be slower than other methods of authentication.
- It requires users to keep their private keys safe.
- It can be difficult to scale to large numbers of users.

13.8 Examples:

NA

13.9 Table Numbering:

NA

13.10 Figures with Captions:

NA

13.11 Self-Assessment Questions:

1. How is asymmetric encryption used for user authentication?

2. What are the benefits of using asymmetric encryption for user authentication?

3. What are the challenges of using asymmetric encryption for user authentication?

13.12 Summary:

In this session clear explanation is given how user authentication is done using asymmetric encryption. Popular protocols are explored.

13.13 Terminal Questions:

1. What is Authentication?

2. Difference between one way authentication and mutual authentication.

13.14 Case Study:

NA

13.15 Answer Key:

1) Asymmetric encryption can be used for user authentication by encrypting a message with the user's public key. The message can then only be decrypted with the user's private key. This proves that the user has the private key, and therefore is the legitimate owner of the public key.

2) The benefits of using asymmetric encryption for user authentication include:

- It is very secure.
- It is relatively easy to implement.
- It can be used to authenticate users over insecure networks.

3) The challenges of using asymmetric encryption for user authentication include:

- It can be slower than other methods of authentication.
- It requires users to keep their private keys safe.
- It can be difficult to scale to large numbers of users.

13.16 Glossary:

- KDC – Key Distribution Center,
- ID – Identifier.
- N – Nonce.
- K – Key.

12.17 References:

1. Cryptography and Network Security Principles and Practice, by William Stallings, Pearson, 7th edition, 2017. 2

2. Cryptography And Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Tata McGraw Hill Education Private Limited, Fourth edition 2015.

3. William Stallings, “Network Security Essentials”, Pearson Education, 7th Edition, 2017.

12.18 Keywords:

Authentication, KDC, User, Asymmetric encryption.