Volume 3, Issue 8, August 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Key Management in Public Key Cryptosystem

Prof. Vikas Malik
Computer Science and Engineering (Network Security), India

Anju Sharma
Computer Science and Engineering (Network Security), India

Annu Malik (Guide)
Computer Science and Engineering (Network Security), India

Abstract— This paper presents a survey on Key Management in Public Key Cryptosystem. This discussion is centered on overview of distribution of public and secret keys in public key cryptography. Key management is the management of cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use, and replacement of keys. It includes cryptographic protocol design, key servers, user procedures, and other relevant protocols. Key management concerns keys at the user level, either between users or systems. This is in contrast to key scheduling; key scheduling typically refers to the internal handling of key material within the operation of a cipher.

The concept of public key cryptography was most appealing because it greatly simplifies some of the problems involved in distribution of secret keys. When applied to encryption, it allows a person sending a message to send a message that can only be read by the receiver, without having a need for the sender and receiver to agree on any secret key. The reason for this is that in public key cryptography, the key used for encryption is different from the key used for decryption. In practice, the methods that have been developed for realizing public key encryption are comparatively slow, and public key cryptography is generally used for encrypting ``session keys'' that are then used for a faster traditional single-key encryption method such as the Data Encryption Standard (DES). In addition to the convenience of key management for encryption provided by public key cryptography, it also provides a means to implement digital signatures. The separation of public and private keys is exactly what is required to allow users to sign their data (with their secret key), allow others to verify their signatures with the public key, but not have to disclose their secret key in the process.

Keywords— Key Management, public keys distribution, secret keys distribution , certificate authority

I. *INTRODUCTION*

Key Management is basically used to handle the distribution different types of keys.

Cryptographic systems may use different types of keys, with some systems using more than one. These may include secret keys or public keys.

- Secret algorithm the keys involved are identical for both encrypting and decrypting a message. Keys must be chosen carefully, and distributed and stored securely.
- Public keys, in contrast, are two distinct keys that are mathematically linked. They are typically used in conjunction to communicate.

II. *KEY MANAGEMENT*

Public Key Encryption has been used to address the problem of key distribution.There are two distinct aspects to the use of public-key cryptography in this regard:

- *Distribution of public keys*
- *Use of public-key encryption to distribute secret keys*

Distribution of Public Keys

It can be grouped into the following general schemes:

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

Public announcement of Public Keys

The point of public key encryption is that the public key is public. Thus if there is some broadly accepted public-key algorithm, such as RSA , any participant can send his or her public keys to any other participant or broadcast the key to the community at large.

© 2013, IJARCSSE All Rights Reserved

Fig. 1 Uncontrolled Public-Key Distribution

For example, because of the growing popularity of PGP(pretty good privacy), which makes use of RSA, many PGP users have adopted the practice of appending their public keys to message that they send to public forums, such as USENET newsgroups and Internet mailing lists.

Weakness

- major weakness is forgery
  - anyone can create a key claiming to be someone else and broadcast it
  - until forgery is discovered and alerts other participants, the forger is able to read all encrypted messages

Publicly Available Directory

A 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. Such a scheme would include the following elements:

1. The authority maintains a directory with a { name ,public keys } entry for each participant.
2. Each participant registers a public keys with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.

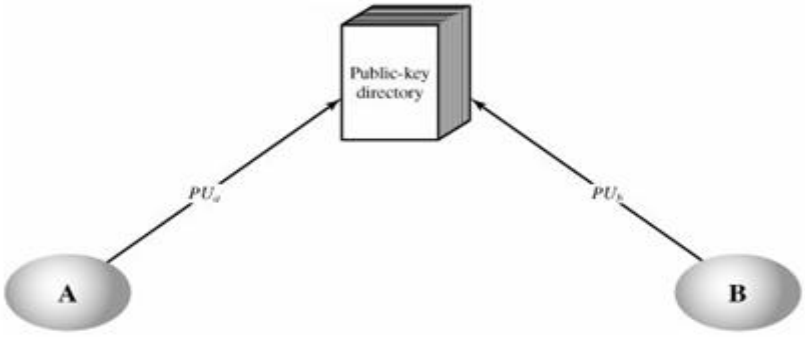

Fig. 2 Public-Key Publication

3. A participant may replace the existing key with a new one at any time.
4. Particiapnt could also access the directory electronically.For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

Weakness

- Still vulnerable to tampering or forgery
  - If an adversary succeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and subsequently impersonate any participant and eavesdrop on messages sent to any participant.

Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. It conssist of central authority maintains a dynamic directory. It requires users to know the public key for the directory, and that they interact with directory in real-time to obtain any desired public key securely. The following steps occur:

© 2013, IJARCSSE All Rights Reserved

Steps :

1. A sends a timestaped 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}. *Thus,* A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:
   - B's public key $PU_{b,}$ which A can use to encrypt message destined for B
   - The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority.
   - The original timestamp, so A can determine that this is not an old message from the authority containing a key other than B's current public key.

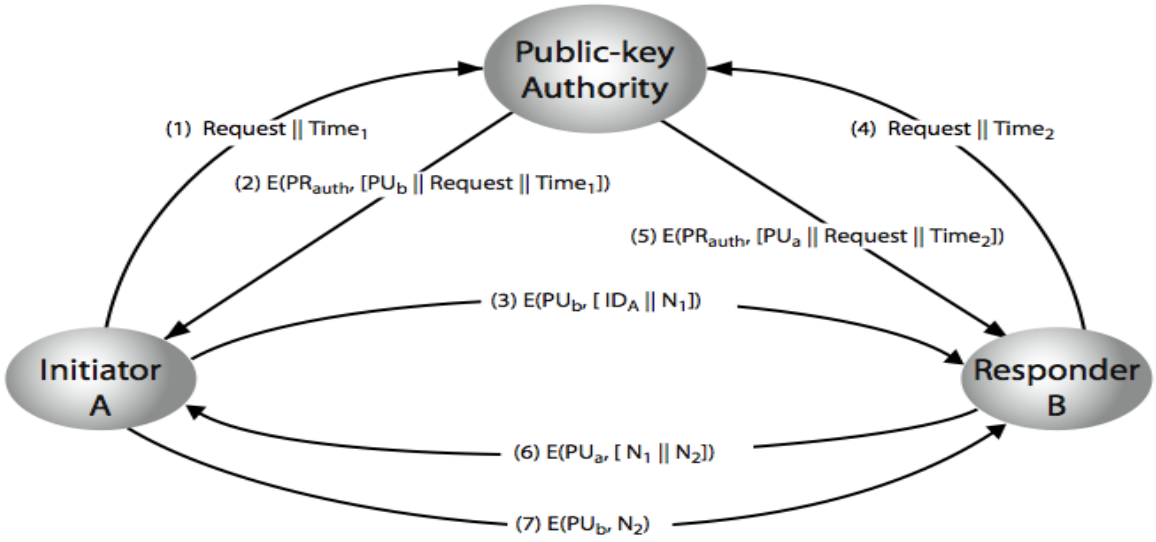

Fig. 3 Public- key Distribution Scenerio

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 retrives A's public key from the authority in the same manner as A retrieved B's public key.
5. At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange. How ever, two additional steps are desirable:
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). Because only B could have decrypted message(3), the presence of N_1 in message (6) assures A that the correspondent is B.
7. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

Thus, a total of seven messages are required. However, the initial four messages need be used only infrequently because both A and B can save the other's public key for future use, a technique known as caching. Periodically, a user should request fresh copies of the public keys of its correspondents to ensure currency.

Public Key Certificates

An further improvement is to use certificates, which can be used to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party. A certificate binds an identity to public key, with all contents signed by a trusted public-key or certificate authority (CA). This can be verified by anyone who knows the public-key authorities public-key.

We can place the following requirements on this scheme:

1. Any participant can read a certificate to determine the name and public keys 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.

© 2013, IJARCSSE All Rights Reserved

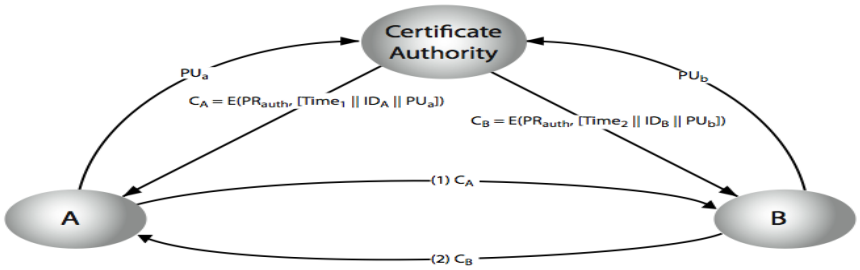

Fig. 4 Exchange of Public-Key Certificates

Each participant applies to the certificate authority, supplying a public key and requesting a certificate. For a participant A, the authority provides a certificate of the form

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

where PR_{auth} is the private key used by the authority and T is a timestamp. A may then pass this certificate on to any other participant, who reads and verifies the certificate as follows:

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

The recipient uses the authority's public key, $PU_{auth,}$ to decrypt the certificate. Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority. The elements ID_A and PU_a provide the recipient with the name and public key of the certificate holder. The timestamp T validates the currency of the certificate.

Weakness

- If A's private key is learned by an adversary . A generates a new private/public key pair and applies to the certificate authority for a new certificate. Meanwhile, the adversary replays the old certificate to B . If B then encrypts messages using the compromised old public key, the adversary can read those messages.

Distribution of Secret Keys Using Public-Key Cryptography

Once public keys have been distributed or have become accessible, secure communication that thwarts eavesdropping, tampering, or both, is possible. However, few users will wish to make exclusive use of public-key encryption for communication Because of the relatively slow data rates that can be achieved. Accordingly, public-key encryption provides for the distribution of secret keys to be used for conventional encryption.

Simple Secret Key Distribution

An extremely simple scheme was put forward by Merkle in1979, as illustated in Fig. 5. If A wishes to communicate with B, the following procedure is followed:

Steps:

1. A generates a public/private key pair { PU_a, PR_a } and transmitts a message to B consisting of PU_a and an identifier of A, ID_A.
2. B generates a secret key, K_s, and transmitts it to A, encrypted with A's public key.

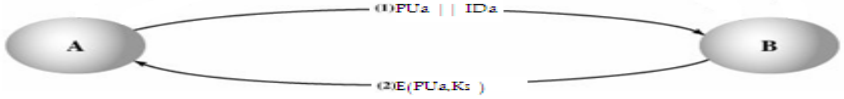

Fig. 5 Simple Use of Public-Key Encryption to Establish a Session Key

3. A computes D(PR_a, K_s)) to recover the secret key. Because only A can decrypt the message only A and B will know the identity of K_s.
4. A discards PU_a and PR_a and discards PU_a.

© 2013, IJARCSSE All Rights Reserved

A and B can now securely communicate using conventional encryption and session keys K_s. At the completion of the exchange , both A and B discards K_s.

Weaknesss

- It 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-the-middle attack.

Secret Key Distribution with Confidentiality and Authentication

It can provides protection against both active and passive attacks. We begin at a point when it is assumed that A and B have exchanged public keys by one of the scheme described earlier.Then the following steps occur.

Steps:

1. A uses B's public key to encrypt a message to B containing an identifier of A encrypted (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). Because only B could have decrypted message (1), the presence of N_1 in message (2) assures A that the correspondent is B.

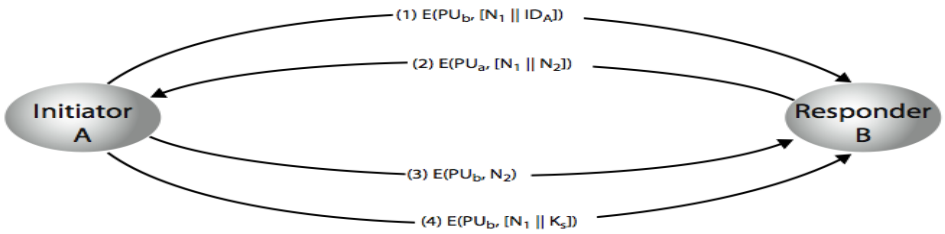

Fig. 6 Public-Key Distribution of Secret Keys

3. A returns N_2, encrypted using B's public-key, to assures 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. Encryption of this message 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.
5. B computes D(PU_a, D(PR_b,M)) to recover the secret key.

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

III. APPLICATIONS

Key management is an important issue in wireless sensor network (WSN) design. There are many key distribution schemes in the literature that are designed to maintain an easy and at the same time secure communication among sensor nodes. The most accepted method of key distribution in WSNs is key predistribution, where secret keys are placed in sensor nodes before deployment. When the nodes are deployed over the target area, the secret keys are used to create the network. For more info see: key distribution in wireless sensor networks.

IV. CONCLUSION

Key Management means managing the keys in a communication. Most of the communications use multicast communication because if the message is sent once by the sender, it will be received by all the users. Main problem in multicast communication is its security. In order to improve the security, various keys are given to the users. Using the keys the users can encrypt their messages and send secretly. The major issue is length of key use, and therefore frequency of replacement. Because it increases any attackers required effort, keys should be frequently changed. This also limits loss of information, as the number of stored encrypted messages which will become readable when a key is found will decrease as the frequency of key change increases. Historically, symmetric keys have been used for long periods in situations in which key exchange was very difficult or only possible intermittently. Ideally, the symmetric key should change with each message or interaction, so that only that message will become readable if the key is learned (*e.g.*, stolen, cryptanalyzed, or social engineered).

REFERENCES

1) http://en.wikipedia.org/wiki/Key_management
2) http://www.google.co.in/
3) http://www.cryptographyworld.com/key.htm
4) Cryptography and Network Security by William Stallings

© 2013, IJARCSSE All Rights Reserved