Asian Journal of Information Technology 4 (11): 986-988, 2005

Keyword Based Authorisation in PKI

N.Durai Pandian and C.Chellappan
Ramanujan Computing Centre,
Anna University, Chennai-25, Tamil Nadu, India

Abstract: Public key infrastructure is a concept which is discussed often in IT security world. PKI is an infrastructure that uses digital certificates as an authentication mechanism and is built to manage certificates and their associated keys. The mechanism of certifying and revoking of public keys, private key escrow has been explored. But less attention has been given to regulating access to stored keys by others. We know all PKI has support for public key registration, look up, revocation of keys, digital signature etc. In this study we propoed extensions to PKI such as key escrow, protected use of keys. This will allow owners to grant privileges to others to perform various actions on their keys. Authorization is given based on keywords also since all data need not be necessary for the third party.General Category: Network Security.

Key words: Access control, authorization, PKI

INTRODUCTION

Assuring privacy of communication and data storage is an integral structure of our information infrastructure. The main purpose of cryptography is to make things very difficult for a third party to get access to the secured information. At the same time if the third party is an authority who believes that they have the rights to access the information by legal or social reasons then employing cryptography for security purposes poses lot of problem in that scenario. In this case we need some mechanism to permit the authorized persons to view our information at the same time information should be protected from other third party. Generally those systems are called as key escrow and we have several mechanisms to deal those issues. Authorized third party who is allowed to eavesdrop may be a Government Organization who wishes to eavesdrop its own citizen or a Corporation seeking access to its own information protected by its own employee.

Generally Key Escrow is useful for data recovery and key recovery. Many commercial systems are available for key escrow. Some such systems are seen in next sub topic. Government key escrow is useful to the government[1] but not to the user as seen by Clipper system where when key of a person is lost government key escrow is not useful in getting back that key. So to overcome that, many commercial systems are built which allow authorized party to eavesdrop and also help to recover the key when key is lost.

Here we proposed a method which will be very useful in Corporate Environments. In this proposal one can give authorizations to a third party to see his message based on some keywords at the same time protecting his private key. Key owner can give authorization to third party by keywords.

The existing commercial systems are either used for Recovery of data or recovery of keys but not on private key escrow for authorization.

The Clipper/Capstone Chips[2], the Bell Atlantic Yaksha system[3] are used for data recovery purposes. Cylink Key Escrow[2] uses Diffie-Hellman techniques for integrating key escrow services into a public key infrastructure.

Micali and Sidney Resilient Clipper[2] like key escrow proposal allows keys to be split so recovery is possible even if some of the escrow agents fail to produce their key components. Micali Guaranteed Partial key escrow. In this proposal the private keys of users are partially escrowed. The escrow agents verify that the bits in their possession are correct and only a relatively small number of bits are unescrowed.

Threshold Decryption[2] is used to share a secret key by a group of escrow agents in such a way that through collaboration of the agents information can be decrypted without the agents releasing their individual key components.

In TIS Commercial key Escrow, data recovery is enabled through master keys held by a data recovery centre[1].

The user controllable Crypto back up system[4] securely and automatically make copies of cryptographic keys for individuals, corporations and under certain conditions, law enforcement agencies.

Corresponding Author: N.Durai Pandian, Research Scholar, Ramanujan Computing Centre, Anna University, Chennai-25. Tamil Nadu, India Tel: 044 2440 1066

PUBLIC KEY INFRASTRUCTURE

PKI is an infrastructure that uses digital certificates as an authentication mechanism and is built to manage certificates and their associated keys[5]. A PKI can be implemented within an organization for the use of users on its network or it can be a commercial entity that issues certificates to Internet users. In both cases PKI has the following components; 1) A registration Authority which verify the identity of the user / requester, 2) A Certificate Authority to issue certificates, 3) Policies that govern the operation of PKI and, 4) Issuance, management and revocation of certificates.

PKI's are important elements in network and Internet security because many communications such as business and E-commerce transactions are dependent on a reliable method to identify the parties of the transaction.

Operation: Registration authority delivers the certificate application to users connected to it. Users want of certificates, fills up the application, generate the key pair, and send public key along with the certificate requisition format to Registration Authority. (PKI itself can generate the key pair for the users, which has certain advantages and disadvantages). RA verifies the details and if found ok it generated certificate request and send to Certificate Authority. CA generates certificate in X509 format and sends it RA. RA sends it back to user and saves a copy in its database.

In some PKI CA itself will do the operation of both CA and RA which has certain advantages and disadvantages.

KEY ESCROW (THRESHOLD SCHEME)

On receiving the request for key escrowing from A, server divides the private key of A say K in to n pieces $K_1, K_2, \ldots K_n$ such that

- Knowledge of any k or more pieces makes K to be computable
- Knowledge of any k-1 or fewer pieces leaves K completely incomputable[6].

Server sends the pieces to n members. To divide K into n pieces, server picks a random k-1 degree polynomial $q(x)=a_0+a_1x+\ldots a_{k-1}$ x k-1 in which $a_0=D$ and evaluate

$$D_1=q(1),\ldots\ldots\ldots D_i=q(i)\ldots\ldots D_n=q(n).$$

Given K, server picks a prime number p which is bigger than K and n. The Coefficients $a_1, a_2, \ldots\ldots a_k$ are randomly chosen from a uniform distribution over the integers in (0,p). The values $D_1, D_2, \ldots D_n$ are computed modulo p.

On receiving the authorization certificate from C Server sends messages to all n members who have the pieces of K. After getting a minimum k values from members together with their identifying indices server finds the coefficients of q(x) by interpolation and then evaluate K=q(0) which gives K.

SCENARIO

A is a senior manager who deals with some important information. He is going on vacation for a period. During that period there is a possibility he may get some important documents. In his absence others cannot read it as it is encrypted. To avoid this situation A decides to give authorization to C. C can see the messages but he should not know A's private key. At the same time C need not see the private information of A. So A includes some keywords which will be expected to be present in the future messages (like quotation). Based on these keywords messages will be filtered and sent to C.

AUTHORISATION MECHANISM

SERVER is responsible for giving public key certificates to users. When users request, it create public-private key pair, issues public key certificate. It is also responsible for distribution of Authorization Certificate.

Protocol:

- A sends his key pair for escrowing purpose. On receiving the request from A for escrowing, Server perform Threshold scheme which is already discussed.
- A send authorization request to SERVER. A→ SERVER: $E_{kuServer}$ [$N_1||ID_a||ID_c||$ E_{kra} [$ID_a||$ $ID_c||$Auth. Req]] Authorization request → $N_2||$ Keywords

Explanation: Since A wants to get authorization certificate for C, it sends the request to SERVER. This request is encrypted by public key of SERVER. It includes Authorization request which consists of a nonce N_2, keywords. This authorization request is encrypted by private key of A for authentication along with IDs of A and C. The nonce N_2 is included for identification of a particular authorization certificate. Since the identity of the person who is going to get authorization powers should not be disclosed to others it is also included in the message which is getting encrypted so that only server can know.

- SERVER sends authorization certificate to A SERVER→A: E_{kua} [N1|| $E_{kuServer}$[Authorization Certificate]]

Authorization certificate: [ID_a||ID_c||N_2||T|| keywords]

Explanation: The SERVER sends the authorization certificate encrypted by its own public key. It contains ID of A and C, Key words, creation time T and nonce N_2 associated with this certificate. Message also includes N_1 which is sent by A along with its request. This total message is encrypted by public key of A thereby only A can decrypt it.

- A sends authorization certificate to C A→C : E_{kuc}[N_3||Public key certificate of A]¨U$E_{kuServer}$[Auth. Certificate]

Explanation: After getting the certificate for C, A forwards the certificate to C. Along with A sends its public key certificate and N_3 encrypted by public key of C so that only C can decrypt. After getting the authorization certificate, C using public key of A sends N_3 to A for confirmation that it has got the authorization certificate.

- When C wants A's message to be decrypted C→ SERVER: $E_{kuserver}$[N_4]|| $E_{kuServer}$[Auth. Certificate]

Explanation: When C wants A's message to be decrypted, it sends the authorization certificate to the server.It also sends a nonce N_4 encrypted by public key of server.

- Server sends the decrypted message to C encrypted by C's public key. SERVER→C: E_{kuc} [N_4|| $E_{krServer}$ [$MESSAGE_a$]]

Explanation: SERVER verifies the authorization certificate sent by C for its originality and validity by first decrypting certificate with its private key.

Server sends message to the members which contains piece of escrowed private keys of A. After receiving pieces from k members it performs interpolation to get original key of A.

Server then decrypts the files with the escrowed private key of A. It also verifies that message should have been sent in a date after the authorization certificate is created. Then it checks whether the specified keywords are present in the decrypted message of A. If the keywords specified in the authorization certificate are present in the message then server encrypts that message with its private key for authentication along with N_4. The entire message is encrypted by C's public key.Then it enters the name of the files, which is sent to c, into the log of A. Then it destroys the private key of A.

CONCLUSION

In this study we presented a protocol, which extends the capabilities of PKI. The above protocol is implemented in PERL. Further issues that can be done include the extension of this protocol to user keys, interaction with other certification authorities. Also hiding the personal information present in the certificate can be looked in to.

The main contribution of this study is extension of PKI facilities by which one can allow others to use his private key without disclosing.

REFERENCES

1. Walker, T.S., B.L. Steven, C.M. Ellison and D.M. Balenson, 1996. Commercial key recovery. Communications of the ACM, 39: 40-46.
2. Denning, E.D. and Dennis K. Branstad, 1996. Taxonomy for key escrow encryption systems. Communications of the ACM., 39: 33-39.
3. Ganesan, R., 1996. The Yaksha security system. Communications of the ACM., 39: 54-60.
4. Mather, D.P., 1996. Crypto back up and key escrow. Communications of the ACM., 39: 47-53.
5. Santosh Chokhani, 1994. Toward a national public key infrastructure. IEEE , Communications Magazine, pp: 70-74.
6. Shamir, A., 1979. How to share a secret, Communications of the ACM., 22: 612-613.