THE DELEGATION PROBLEM AND PRACTICAL PKI-BASED SOLUTIONS

Venus L.S. Cheung, Lucas C.K. Hui, S.M. Yiu
Department of Computer Science and Information Systems
The University of Hong Kong, Hong Kong

Keywords: Delegation, Public Key Infrastructure (PKI), Information Systems Security

Abstract: Delegation is a process where a delegator grants or authorizes all or some of his/her power to another a delegate to work on his/her behalf. In an office, it is common for officers to delegate their power to subordinates. In a digital environment (e.g. a secure enterprise information system with confidential electronic documents), how delegation can be handled properly is still an open question. In this paper, we address the delegation problem in the context of a secure information system, lay down a set of requirements from the users' point of view and propose several practical PKI-based schemes to solve the problem. Analysis on the proposed schemes concludes that Proxy Memo can solve the problem quite efficiently while reducing the key management problem.

1 INTRODUCTION

Delegation is a process where a delegator grants or authorizes all or some of his/her power to a delegate, to work on his/her behalf. This is not uncommon for officers to delegate their power to subordinates when they need assistance. The delegation of power is divided into three main types:

- **Acting a role**
  A Manager, Alice, is leaving for a business trip for a certain period of time and she will have no Internet access in the place she is going to. Therefore, she wants somebody to act on her behalf, in order to keep the projects on-going. In this case, she may like to delegate her power to somebody during her trip.
- **Sharing power**
  Another Manager, Bob, has a lot of responsibilities that he wants somebody to share some of his workload. Therefore, he delegates some of his power to, say, his Secretary, Carol, so she will be responsible for screening and sorting documents before passing to Bob, or signing documents (e.g. notices) on behalf of Bob.
- **Collaborative group**
  Similar to the above, Bob may want to delegate his power to more than one people in order to lessen the workload of Carol. He delegates his power of signing documents to Carol and the power of receiving confidential documents to another Manager, David. As a result, Carol and David will be working as a group of delegates of Bob.

In a paper-based environment, delegation can be achieved easily as discussed by Cheung et al. (Cheung et al., 2004). However, more and more organizations are switching to a digital documents. Office procedures are handled electronically. In such a digital office, a secure information system will usually be deployed to control the access to information (especially for confidential information) and to ensure the confidentiality of communication between any two parties. Manual procedures for handling delegation cannot be directly applied in such a secure information system. In fact, how delegation can be handled properly within a secure information system is still an open question. In this paper, we focus on the discussion of this delegation problem within the context of a secure information system.

The security goals of confidentiality, integrity, authentication and non-repudiation in a secure information system can usually be realized by the Public-Key Infrastructure (PKI) (Ford and Baum, 1997). Intuitively, one may wonder if the way we handle delegation in a paper-based environment can be seamlessly simulated in the digital office environment. It is possible but there are serious security concerns arising from the digital environment setting (Cheung et al.,

L. S. Cheung V., C. K. Hui L. and M. Yiu S. (2004).
THE DELEGATION PROBLEM AND PRACTICAL PKI-BASED SOLUTIONS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 73-78
DOI: 10.5220/0002627100730078

2004).

In this paper, we lay down a set of requirements for the delegation problem from a user's point of view and propose simple and practical PKI-based solutions to solve this delegation problem within a secure information system. The rest of the paper is organized as follows. Section 2 discusses some related work for this delegation problem in a secure information system. Section 3 identifies the requirements for solving the delegation problem. Section 4 suggests some simple and practical schemes that can be used to solve the problem. In Section 5, we analyze the schemes based on the identified requirements. Section 6 concludes the paper and discusses some of our future works on the problem.

2 RELATED WORK

The very first discussion about delegation was by Gasser and McDermott (Gasser and McDermott, 1990). They suggested a detailed architecture in a distributed system. Their delegation model was a user authorizing a system or process to act on his/her behalf. Varadharajan et al. (Varadharajan et al., 1991) suggested two signature-based schemes to achieve delegation and compared their results with (Gasser and McDermott, 1990). Varadharajan et al. commented that Gasser et al.'s delegation scheme was less flexible in access control. Neuman (Neuman, 1993) and Ding & Petersen (Ding and Petersen, 1995) gave clear classification of the delegation models that the former suggested a proxy model for use in distributed systems and was capability-based. It was for use in Kerberos Version 5 (not PKI) depending on access control lists (ACLs).

Another direction for solving delegation problem is the proxy signature and proxy decryption. A seminal paper by Mambo et al. (Mambo et al., 1996a; Mambo et al., 1996b) presented a detailed classification and conditions for proxy signatures (the MUO scheme). Mambo et al. proposed several proxy signature schemes and the security of them are checked against the conditions. Based on the signature schemes, Mambo et al. proposed schemes on proxy decryption (Mambo and Okamoto, 1997). Kim et al. (Kim et al., 1997) proposed new types of proxy signatures based on MUO's partial delegation to suit group needs. Lee et al.(Lee et al., 2001) investigated in the weakness of the proxy signature schemes (Mambo et al., 1996b; Kim et al., 1997) and proposed new classifications of proxy signatures. They suggested and concluded that a "strong" type of proxy signatures can achieve the same propose with application examples.

Theoretically, some of the above schemes or models can be deployed to solve the delegation problem in a secure information system, but they are yet too complicated to be used and understood by the majority of users. The security of some of the schemes above rely on the security of their mathematical counterparts, which is quite "abstract" to the users. Also, the security loopholes, if the schemes were implemented, were scarcely discussed. In fact, the implementation of most of these schemes are complicated and there is no existing implementation available to be used with a secure information system. So, a more practical and easy to use solution is desirable.

3 REQUIREMENTS FOR DELEGATION USING PKI

In this section, based on the scenarios we discussed, we have identified a set of requirements for solving the delegation problem. Our investigation commences with some assumptions that facilitate a smooth flow of ideas. It is noted that the assumptions are reasonable while they could be seemed as a tighter control.

Assumptions

For the ease of discussion, in this paper, we only consider a secure intranet and any security operation including delegation should be performed in PCs located in the office. Each user has different sets of keys for signing and for decryption of encrypted documents. This is to separate the signing and decryption ability of private keys. Indeed this makes the delegation of signing and decryption right clearly separated. For simplicity, only two kinds of delegation are considered in our discussion, i.e. "signing" and "decryption". In fact, other kinds of delegation can be handled in a similar manner. In role-based access control, the roles are arranged in a hierarchy and authorization to roles is limited according to their positions in the hierarchy. Here we do not have this restriction but we still conserve the fact that normal delegation should be conducted from an officer in high hierarchy to officer(s) in low hierarchy.

Requirements

R1. Partial delegation should be supported.
The delegator can choose to delegate all or only part of his rights. Here only signing and/or decryption rights are considered for simplicity.

R2a. Re-delegation should be supported.

R2b. Receiver(s) can easily identify the re-delegation relation from what they received.
This is the case when the delegate(s) want(s) to further delegate the delegated power to others (sometimes known as "Transitive delegation"). This will form a "delegation chain". In the first place, re-delegation should be supported and then the

original delegator has control on choosing if this should be activated. Once activated, a receiver can easily identify the re-delegation relation without querying any servers or databases.

R3. Delegation chain should be traceable.
The "delegation chain" should be revealled and the delegator(s) and delegate(s) in the chain should be traceable at first glance (without querying the servers).

R4. Broken chain should remove all the delegated powers of the delegate(s).
This is to avoid improper use of power by delegate(s) at the later part of the chain if some delegated rights in the middle of the chain are removed/expired. For example, if A delegates to B and B delegates to C, if B's service to the company is terminated within the delegation period, all the rights by C (from A) should be automatically removed.

R5. Redemption of the delegated power should be supported.
A delegator should be able to get back or remove the delegated power from the delegate(s) once he resumes duty or revokes their rights.

R6a. Shortening of the delegation period should be supported.

R6b. Extension of the delegation period should be supported.
This happens when a delegator resumes duty before the end of the delegation period or not able to take up his work after the delegation period. This provides flexibility to the delegator when he cannot accurately estimate the end time of the delegation. The delegator reserves the right on activating this function and on the number of days allowed.

R7a. Number of new signing key pairs should be kept minimum.

R7b. Number of new decryption key pairs should be kept minimum
This is to minimize the problem of key management. Therefore, if new keys are required to be generated in the delegation process, its number should be kept minimum. If existing keys are used, it should be ensured that they cannot be forged by anyone inside and outside the delegation relation.

4 PROPOSED SCHEMES

In this section, we propose several simple and practical solutions for solving the delegation problem. Let A be the delegator who grants (some of) his power to a delegate, B, who receives the delegated power. We make use of a server called the *Delegation Server (DelSvr)* which is mainly responsible for handling delegation requests and verifying delegation information. It is connected to the *Delegation Database (DelDB)* for which details of the delegation are stored. The DelSvr is connected to other servers in order to incorporate other functions in a system, such as checking of user credentials when they login. Users have to login to the DelSvr in order to perform delegation request. They login using their unique username and password. Connections between the users, DelSvr, and DelDB are assumed to be secure and leakage of information during transfer is unlikely to occur.

For the delegation of signing right, we propose two approaches. One is the issuance of a new signing key pair (NK) and another is called a "Proxy Memo (PM)". For the delegation of decryption right, we propose three approaches: issuance of a new decryption key pair, Proxy Memo, and decryption-and-re-encryption by DelSvr. The approaches are subdivided into two or three alternatives. In total, there are three variations for solving the delegation of the signing right and five variations for solving the delegation of the decryption right. Any variation for signing can be combined with any variation for decryption to form a valid scheme for solving the delegation problem. The details of these approaches are discussed in the following subsections.

Issuance of new key pair

The issuance of a new key pair is the most straightforward approach for solving the delegation problem and can be divided into two alternatives: a new key pair is generated per delegation request or per delegate.

1. New key pair per delegation request (NK1) A intends to delegate his power to B. A logs in DelSvr and enters delegation information. Information required are almost the same as those required for a Proxy Memo (Table 4) except that the identity of the delegate is identified by his public key. The information is submitted to DelSvr for checking on the eligibility for delegation to take place. (Errors would show if, for example, the delegator delegates the same right to the same delegate a second time during the delegation period.) The information will be stored in DelDB and B will be notified via email. B logs in DelSvr and clicks "accept" to accept A's request. DelSvr sends a "key generation" request as well as the information provided by A to the Certification Authority (CA). A new signing key pair is generated and recorded in the DelSvr and DelDB. The signing (private) key is sent to B and B is asked to download it (in local PC or a disk). B should use the new signing key to perform any signing activities on behalf of A during the delegation period.

The new key pair expires on the end date specified by A. If A resumes her duty before the end of the delegation period, she can send a revoke request to DelSvr and the key pair is revoked. If A knows that

Table 1: Entries in a Proxy Memo

Identity of delegator: (name, staff id, certificate id)
Identity of delegate: (name, staff id, certificate id)
Delegated rights: Signing and/or Decryption
Permission to re-delegate: Yes or No
Delegation period: (from date/time to date/time)
Maximum extension days: 0 or above
(other information)

she is not able to resume duty on time and would like B to act on the position for a few more days, A should have specified the "maximum extension days" to, say, three days, so when the end of period approaches, B can log in the DelSvr and request extension of power. A new key for the extension period will be generated for him. However, B is not able to request extension after the signing power is expired, even though A has specified the "maximum extension days". When A comes back, he simply revokes B's signing key (extension period) if it is not the end of the extension period. If A does not want any extension, he can set the "maximum extension period" to zero.

The issuance of a new decryption key pair is the similar to above except that A should change or include "decryption" in the delegated rights. All users are required to query DelSvr for the most up-to-date delegation status. Suppose a sender, S, wants to send encrypted documents to A but noted that B is acting on A's behalf, S may have to encrypt the document with B new encryption key and send it to B. This can be made transparent to the user by the application layer of the system. B will then be able to decrypt S's document and respond immediately. After the delegation period, B should share a copy of the new decryption key with A so that A can decrypt the encrypted documents processed during the delegation period.

2. New key pair per person (NK2) This is similar to the above for both signing and decryption except that only one pair of keys will be generated per person and it will be used at any circumstances of delegation (until expiry/revocation). The keys are of no specific delegator and delegation period. The delegation relation is recorded in DelDB when they are in use. In this case, a delegate will not share the private keys with the delegator.

For re-delegation (if the delegator approves this by selecting the "re-delegation" option), B logs in DelSvr and repeat the process. A new key pair will be generated for the new delegate during the delegation period. It should be noted that the delegation period and rights for the new delegate should be within that of B.

Issuance of Proxy Memo

A Proxy Memo (the "Memo") is an "object" which acts as a proof delegation. It is like a digital form that users have to fill information in it. Later it should be signed by the delegator and received and used by one or more delegate(s).

For creation of Proxy Memo, please refer to Cheung et al. (Cheung et al., 2004). The use of Memo for delegation of decryption right has three alternatives:

1. Request to encrypt with DelSvr's encryption key (PMD1) When DelSvr knows a sender, S, wants to encrypt a document to a recipient (A) who has delegated his/her decryption right (to B) during the period, DelSvr will direct S to encrypt the document with DelSvr's encryption key. (S may reject the offer if he did not intend to do so.) S encrypts the document with DelSvr's encryption key and send the document to A. B receives the encrypted document (because of an email forwarding function by DelSvr) and finds that it is encrypted with DelSvr's encryption key. B logs in DelSvr and presents both the Memo and the encrypted document, after verification, DelSvr decrypts the document with its own decryption key and re-encrypts the document with B's encryption key. The re-encrypted document is returned to B.

2. Request to encrypt with delegator's encryption key (PMD2) To facilitate this function, A should have submitted a copy of his decryption key to DelSvr. S does not have to be aware whether A has delegated his right to others. S just encrypts the document with A's public key and send the document to A. B receives the document (because of an email forwarding function by DelSvr) and knows that the document is for A. B logs in DelSvr and presents both the Memo and the encrypted document, similar to the above, DelSvr decrypts and re-encrypts document and returns it to B.

3. Request to encrypt with delegate's encryption key (PMD3) To cut the decryption and re-encryption processes by DelSvr, DelSvr will instruct S to encrypt the document with B's encryption key. When B receives the encrypted document, he can decrypt it and respond immediately.

To re-delegate the decryption right, B logs in DelSvr and makes a re-delegation request similar to the **NK–** schemes.

By using either PMS, NK1, or NK2 for the delegation of signing right and either PMD1, PMD2, PMD3, NK1, NK2 for the delegation of decryption right, we can have 15 schemes. In the next section, we will analyze these 15 schemes according to the set of identified requirements.

5 ANALYSIS OF THE SCHEMES

In this section, we are going to analyze the security and functionality of the constructs of the schemes. A summary of the analysis is given in Table 2. It should

be noted that the number of keys are counted in pairs and the comparison is per delegation process. From the table, it seems that Proxy Memo for the delegation of both signing and decryption is the best choice among the schemes.

On issuance of new pair of keys

Issuing short-term key pairs avoids the problem of giving out one's key copy. The new signing key is held by the delegate during the delegation period and destroyed when it is revoked or expired. The new decryption key will be shared with the delegator after that period so the delegator can view the document after the period.

For signing purpose, issuing a new pair of signing keys seems unnecessary. This is because the delegation often appears in the content of the document and the new signature confuses the identity of the signer. However, when the proxy signer ever forgets to address the delegation from time to time for every document, the new signing key serves a "declaration" purpose.

For decryption purpose, issuing a new pair of decryption keys is necessary. It allows the sender to choose how the document should be handled. If immediate response is required, he should use the new encryption key. Or else, he can choose to encrypt it with the delegator's original encryption key. Apart from this, documents are classified by confidentiality and the delegator (as well as the sender) takes control on disclosure of information to delegates.

The total number of decryption key pairs will increase significantly if a delegator delegates frequently. It is necessary for issuing new pairs of keys each time to prohibit cross-reference by different delegates in different periods of time. However, there are also approaches to reduce the number of key pairs for key management issue. One is to reduce the number of key pairs by limiting one key pair (for delegation purpose only) to the one person. By doing so, the encryption key should be "hidden" to all when it is not in-use and "reappear" when delegation (to the same person) takes place.

On issuance of Proxy Memo

For delegation of signing rights, a Proxy Memo is used. The design of the Memo is flexible, it just resembles a text document. Besides it contains essential information in order to prove its authenticity at a later stage of verification by third parties. DelSvr provides a unique identifier and a hash value to each Memo. This is to prohibit the delegator from changing the information on the Memo after its creation but before signing. The Memo should be signed with the delegator's signing key. Table 4 shows the structure of a Proxy Memo.

The delegate(s) should sign according to the signing capabilities listed on the Memo and attach the Memo to the documents as a proof of evidence. A recipient can submit a Memo to DelSvr to check for its validity. A "valid/invalid" or "revoked/expired" value is returned to the recipient.

For delegation of decryption rights, the decryption and re-encryption option puts heavy weights on the DelSvr but it secures and limits the rights of both the delegator and the delegate. By encrypting the document first with DelSvr or the delegator's key ensures the document's originality (not being altered) (because the delegator will have an encrypted copy in his mailbox). The DelSvr will only decrypt the document for the delegate if and only if he can present the correct Memo. The returned document will be re-encrypted using delegate's encryption key so it is secure in transit.

For encryption of documents using the delegate's encryption key, the process is simpler but a dishonest delegate may alter the content before re-encrypt it with the delegator's encryption key (to leave a copy for the delegator). The encryption will only mean securing transaction between the sender and the delegate.

On DelSvr and DelDB

The system relies on the DelSvr and DelDB to record and process the delegation information and requests. DelSvr is connected to other system servers such as email server. DelDB is connected to DelSvr only and it is separated from other servers. The security of DelSvr and DelDB is beyond focus of this paper.

6 CONCLUSION

This paper discusses the problems of delegation in an organization environment within the context of a secure PKI-based information system. The basic design of PKI does not consider the issues of delegation, so a straight-forward deployment of PKI does not solve the delegation problem. Digital document security is a becoming a great concern and delegation is a common practice in an office environment, so it is desirable to find an appropriate solution to solve the problem. We identified a set of requirements for the delegation problem from a user's point of view and proposed several simple and practical schemes for handling the delegation in a PKI-based information system. Based on our analysis, among the proposed solutions, it seems that the use of the Proxy Memo can meet most of the identified criteria while not inducing any problem in key management. However, for applications that handling of PM is not available, issuing new pairs of keys would be a good choice.

Further investigations include the following. A study on the implementation details of this Proxy Memo in order to realize the design idea will be carried out. Existing design of the proposed solution is

Table 2: Summary of the proposed schemes.

Signing	Decryption	R1	R2a	R2b	R3	R4	R5	R6a	R6b	R7a	R7b
PMS	**PMD1**	√	√	√	√	√[a]	√	√	√	0	0
	PMD2	√	√	√	√	√[a]	√	√	√	0	0
	PMD3	√	√	√	√	√[a]	√	√	√	0	0
	NK1	√	√	×	×	√[a]	√	√	×[b]	0	1+[c]
	NK2	√	√	×	×	√[a]	√	√	×[b]	0	1
NK1	**PMD1**	√	√	×	×	√[a]	√	√	√	1+	0
	PMD2	√	√	×	×	√[a]	√	√	√	1+	0
	PMD3	√	√	×	×	√[a]	√	√	√	1+	0
	NK1	√	√	×	×	√	√	√	×[b]	1+	1+
	NK2	√	√	×	×	√	√	√	×[b]	1+	1
NK2	**PMD1**	√	√	×	×	√[a]	√	√	√	1	0
	PMD2	√	√	×	×	√[a]	√	√	√	1	0
	PMD3	√	√	×	×	√[a]	√	√	√	1	0
	NK1	√	√	×	×	√	√	√	×[b]	1	1+
	NK2	√	√	×	×	√	√	√	×[b]	1	1+

[a] For PM–. The power of the PM holder is verified when his PM is submitted to DelSvr.
[b] For NK–. The original keys will be expired on the date specified. New keys will be issued to simulate extension of the delegation period.
[c] +: "or more". More keys will be needed when the delegation period is extended.

based on a simplified model, for example, all documents are of the same security level. So, the next step is to enhance the design of the Proxy Memo so that it will be flexible to handle additional variations arising from actual requirements of the users such as the categorization of the security levels of documents (top secret, confidential, and so forth). Proxy Memo may not be the best practical solution for solving the problem, so continuous research should be carried out to design better schemes for solving this delegation problem. In particular, non-PKI-based solutions may need to be explored.

REFERENCES

Cheung, V. L. S., Hui, L. C. K., Yiu, S. M., Chow, K. P., Pun, K. H., Tsang, W. W., Chan, H. W., and Chong, C. F. (2004). Delegation of Signing and Decryption Rights Using PKI Proxy Memo, to appear in Proceedings of the IASTED International Conference on Software Engineering (SE2004).

Ding, Y. and Petersen, H. (1995). A new approach for delegation using hierarchical delegation tokens. Technical report, University of Technology, Chemnitz-Zwickau.

Ford, W. and Baum, M. (1997). *Secure Electronic Commerce: building the infrastructure for digital signatures and encryption*, chapter 7, pages 263–308. Prentice Hall.

Gasser, M. and McDermott, E. (1990). An Architecture for Practical Delegation in a Distributed System. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 20–30.

Kim, S., Park, S., and Won, D. (1997). Proxy Signatures, Revisited. In *Proceedings of ICICS'97, International Conference on Information and Communications Security*, LNCS 1334, pages 223–232. Springer-Verlag.

Lee, B., Kim, H., and Kim, K. (2001). Strong Proxy Signature and its Applications. In *The 2001 Symposium on Cryptography and Information Security*, Oiso, Japan.

Mambo, M. and Okamoto, E. (1997). Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. In *IEICE Trans. Fundamentals*, volume E80-A, No.1, pages 54–63.

Mambo, M., Usuda, K., and Okamoto, E. (1996a). Proxy Signatures: Delegation of the Power to Sign Messages. In *IEICE Trans. Fundamentals*, volume E79-A, No. 9, pages 1338–1354.

Mambo, M., Usuda, K., and Okamoto, E. (1996b). Proxy Signatures for Delegating Signing Operation. In *Proceedings of Third ACM Conference on Computer and Communications Security*, pages 48–57.

Neuman, B. C. (1993). Proxy-Based Authorization and Accounting for Distributed Systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 283–291, Pittsburgh.

Varadharajan, V., Allen, P., and Black, S. (1991). An Analysis of the Proxy Problem in Distributed Systems. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 255–275.